SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 1-14, January 1994

SCALING MATRICES TO PRESCRIBED ROW AND COLUMN MAXIMA*

URIEL G. ROTHBLUM†, HANS SCHNEIDER‡, AND MICHAEL H. SCHNEIDER§

Abstract. A nonnegative symmetric matrix B has row maxima prescribed by a given vector r, if for each index i, the maximum entry in the ith row of B equals r_i. This paper presents necessary and sufficient conditions so that for a given nonnegative symmetric matrix A and positive vector r there exists a positive diagonal matrix D such that $B = DAD$ has row maxima prescribed by r. Further, an algorithm is described that either finds such a matrix D or shows that no such matrix exists. The algorithm requires $O(n \lg n + p)$ comparisons, $O(p)$ multiplications and divisions, and $O(q)$ square root calculations where n is the order of the matrix, p is the number of its nonzero elements, and q is the number of its nonzero diagonal elements. The solvability conditions are compared and contrasted with known solvability conditions for the analogous problem with respect to row sums. The results are applied to solve the problem of determining for a given nonnegative rectangular matrix A positive, diagonal matrices D and E such that DAE has prescribed row and column maxima. The paper presents an equivalent graph formulation of the problem. The results are compared to analogous results for scaling a nonnegative matrix to have prescribed row and column sums and are extended to the problem of determining a matrix whose rows have prescribed l_p norms.

Key words. symmetric matrices, potential, directed graph, weighted graph, scalings, row-maxima

AMS subject classifications. 05L20, 15A99, 90B10

1. Introduction. A matrix B is called a *symmetric scaling* of a nonnegative square matrix A if $B = DAD$ for some positive diagonal matrix D. A matrix B is called an *equivalence scaling* of a nonnegative rectangular matrix A if $B = DAE$ for some positive diagonal matrices D and E. In this paper we give necessary and sufficient conditions that a given symmetric nonnegative matrix A has a symmetric scaling with prescribed row maxima. In particular, we show that for a given pattern (i.e., locations of strictly positive entries) if the class of symmetric nonnegative matrices with that pattern and having the prescribed row maxima is nonempty, then every nonnegative matrix with that pattern can be symmetrically scaled into the class. Thus, our conditions relate the prescribed row maxima to the pattern of the matrix A.

Further, we describe an algorithm that for a given matrix A either determines a symmetric scaling B with prescribed row maxima or shows that no such scaling exists. Using our results for symmetric scalings, we also establish corresponding results for the problem of determining an equivalence scaling of a rectangular nonnegative matrix with prescribed row and column maxima. Our results have natural interpretations in terms of weighted undirected graphs.

We call the problem of finding for a given square nonnegative matrix a symmetric scaling with prescribed row maxima *max symmetric scaling* and the problem of finding for a given rectangular nonnegative matrix an equivalence scaling with prescribed row

* Received by the editor November 19, 1991; accepted for publication (in revised form) May 11, 1992.

† Rutgers Center for Operations Research, Rutgers University, New Brunswick, New Jersey 08903 and Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Technion City, Haifa 32000, Israel (IERURI0@TECHNION). Research supported in part by Israel-United States Binational Science Foundation grant 87-00194.

‡ Department of Mathematics, University of Wisconsin, Madison, Wisconsin 53706 (hans@!math.wisc.edu). Research supported in part by National Science Foundation grants DMS-8901445 and ECS 87-18971, Binational Science Foundation grant 87-00194, and the International Scientific Research Program, the Ministry of Education, Science and Culture, Japan.

§ AT&T Bell Laboratories, HO 3K-316, Holmdel, New Jersey 07733 (mschneider@attmail.com) and Department of Mathematical Sciences, Johns Hopkins University, Baltimore, Maryland 21218. Research supported in part by National Science Foundation grant ECS 87-18971.

and column maxima *max equivalence scaling*. These problems are analogues of corresponding well-studied scaling problems in which row and column sums are prescribed. We refer to the sum versions of these problems as *sum symmetric scaling* and *sum equivalence scaling*. The problem of sum equivalence scaling was described in the engineering literature by Kruithof [10] and was considered by Sinkhorn [17], Brualdi [5], Sinkhorn and Knopp [18], Bregman [1], Menon [12], Menon and Schneider [13], Schneider and Zenios [16], and many other authors. The sum symmetric problem was considered by Brualdi [3], [4] and Marshall and Olkin [11].

One important application of sum equivalence scaling concerns the updating of (dynamic) data that is given in matrix form, e.g., traffic intensity between sources and destinations. When new data is not fully observable, but new marginals consisting of corresponding row sums and column sums are observable, a common technique is to replace the old data given by a matrix A by a scaling DAE whose row sums and column sums equal the observed marginals. Max equivalence scaling arises naturally when observations about the new data concern row and column maxima.

We describe our notation in § 2 and list some solvability results for sum symmetric and sum equivalence scalings in § 3. We consider these problems both for a given pattern and for all subpatterns of a given pattern, and we add some new results in the latter case. In § 4 we give nine equivalent conditions for the existence of a solution of the max symmetric scaling. In § 5 we present an algorithm that, for a given nonnegative matrix A, either symmetrically scales A to have prescribed row maxima or determines that no such scaling can exist. In § 6, we apply the results of § 4 to study max equivalence scaling, and in § 7 we restate our results in terms of weighted undirected graphs. Finally, in § 8 we unify max symmetric scaling and sum symmetric scaling by considering scaling problems in which the l_p norms of the rows of the matrix are prescribed.

2. Notation and definitions. For a positive integer n, we use the notation $\langle n \rangle$ to denote the set of integers $\{1, 2, \ldots, n\}$. For a subset $I \subseteq \langle n \rangle$, we use I^c to denote the set $\langle n \rangle \backslash I$, the *complement of I with respet to* $\langle n \rangle$. The identity of n will always be clear from the context. The cardinality of a finite set S is denoted $|S|$. Also, we use the symbols $\subset$ and $\subseteq$ to denote strict and weak containment, respectively.

Let A be an $m \times n$ nonnegative matrix and let I and J be nonempty subsets of $\langle m \rangle$ and $\langle n \rangle$, respectively. We use the notation A_{IJ} to denote the $|I| \times |J|$ submatrix of A corresponding to the rows and columns of A, indexed by I and J, respectively. We identify an index i and the set $\{i\}$. For example, when $I = \{i\}$, we write A_{iJ} for A_{IJ}. By convention, we write $A_{IJ} = 0$ if either I or J equals the empty set.

For a vector $r = (r_1, \ldots, r_n)^T \in \Re^n$ and subset $I \subseteq \langle n \rangle$, we use the notation r_I to denote the subvector of r whose entries are r_i for $i \in I$, and we use $r(I)$ to denote the element sum of r_I. We follow the standard convention that the summation over the empty set is defined to be 0. Also, the value of $\max_{i \in I} r_i$ in the case of $I = \varnothing$ depends on the underlying group to which the elements r_i belong. Specifically, if we are considering the entries of r as entries of the multiplicative group of nonnegative real numbers, then the maximization over the empty set is defined to be 0, whereas if the entries are viewed as elements of the additive group of real numbers, then the maximization over the empty set is defined to be $-\infty$. The additive case arises in § 7 when we consider graph versions of our results. Also, for $\alpha > 0$, we define the operation $\frac{\alpha}{0} = +\infty$. This operation will occur only in minimization expressions over sets containing an element of finite value.

The $n \times n$ diagonal matrix whose diagonal entries are $d_1, d_2, \ldots, d_n$ is denoted $\operatorname{diag}(d_1, d_2, \ldots, d_n)$. A diagonal matrix $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$ is called *positive* if $d_i > 0$ for $i \in \langle n \rangle$. For an $n \times n$ nonnegative symmetric matrix A, a matrix B is called

a *symmetric scaling of* A if $B = DAD$ for some positive diagonal matrix D. For an $m \times n$ nonnegative matrix A, a matrix B is called an *equivalence scaling of* A if $B = DAE$ for some positive diagonal matrices D and E.

An $m \times n$ matrix $P = [p_{ij}]$ is called a *pattern* matrix if every entry of P is either 0 or 1. Given two $m \times n$ pattern matrices P and P', the matrix P' is a *subpattern* of P if $P' \leqq P$. Given a $m \times n$ nonnegative matrix $A = [a_{ij}]$, the *pattern of* A is the $m \times n$ pattern matrix P such that for $i \in \langle m \rangle$ and $j \in \langle n \rangle$

$$p_{ij} = \begin{cases} 1 & \text{if } a_{ij} > 0, \quad \text{and} \\ 0 & \text{otherwise.} \end{cases}$$

For an $m \times n$ pattern matrix P, we define the *pattern class of* P, written $\Pi(P)$, to be the set of all $m \times n$ nonnegative matrices whose pattern is P.

3. Existence conditions for sum scaling. We summarize numerous characterizations for the solvability of sum symmetric scaling and sum equivalence scaling.

We call a matrix, or a vector, *positive* if all of its elements are positive. For a positive vector $r = (r_i, \ldots, r_n)^T \in \mathfrak{R}^n$, let $S(r)$ denote the set of all $n \times n$ nonnegative matrices $A = [a_{ij}]$ such that

$$\sum_{j=1}^{n} a_{ij} = r_i \quad \text{for } i \in \langle n \rangle. \tag{1}$$

Conditions (i), (ii), (iii), and (vi) of the following theorem are contained in Brualdi [3], [4]. Conditions (iv) and (v) are, apparently, new.

THEOREM 1. *Let P be an $n \times n$ symmetric pattern matrix, and let $r \in \mathfrak{R}^n$ be strictly positive. Then the following are equivalent*:

(i) *Each symmetric $A \in \Pi(P)$ has a symmetric scaling B in $S(r)$.*

(ii) *Some symmetric $A \in \Pi(P)$ has a symmetric scaling B in $S(r)$.*

(iii) *The set $\Pi(P) \cap S(r)$ is nonempty.*

(iv) *If I and J are subsets of $\langle n \rangle$ such that $P_{IJ} = 0$, then $r(I) \leqq r(J^c)$ with equality holding if and only if $P_{I^cJ^c} = 0$.*

(v) *If I and J are subsets of $\langle n \rangle$ such that $P_{IJ} = 0$, then $r(I \cap J) \leqq r((I \cup J)^c)$ with equality holding if and only if $P_{(I \cap J)^c,(I \cup J)^c} = 0$.*

(vi) *If $\{K, L, M\}$ is any partition of $\langle n \rangle$ such that $P_{K,K \cup L} = 0$, then $r(K) \leqq r(M)$ with equality holding if and only if $P_{L \cup M,M} = 0$.*

Proof. The equivalence of (i), (ii), (iii), and (vi) is given in Brualdi [3], [4]. The implication (iii) $\Rightarrow$ (iv) is found in [13], and the equivalence of (iv) and (v) follows from the observations that $r(I) = r(I \cap J) + r(I \backslash J)$ and $r(J^c) = r((I \cup J)^c) + r(I \backslash J)$. Finally, to see that (v) $\Rightarrow$ (vi), consider a partition $\{K, L, M\}$ of $\langle n \rangle$ such that $P_{K,K \cup L} = 0$. Then apply (v) to the sets K and $K \cup L$. □

For positive vectors $r = (r_1, \ldots, r_m)^T \in \mathfrak{R}^m$ and $c = (c_1, \ldots, c_n)^T \in \mathfrak{R}^n$, let $S(r, c)$ be the set of all $m \times n$ nonnegative matrices $A = [a_{ij}]$ such that

$$\sum_{j=1}^{n} a_{ij} = r_i \quad \text{for } i \in \langle m \rangle \quad \text{and} \quad \sum_{i=1}^{m} a_{ij} = c_i \quad \text{for } j \in \langle n \rangle. \tag{2}$$

The following theorem summarizes results of Menon [12], Brualdi [2], and Menon and Schneider [13].

THEOREM 2. *Let P be an $m \times n$ pattern matrix, and let $r \in \mathfrak{R}^n$ and $c \in \mathfrak{R}^m$ be strictly positive. Then the following are equivalent*:

(i) *Each matrix $A \in \Pi(P)$ has an equivalence scaling B in $S(r, c)$.*

(ii) *Some matrix $A \in \Pi(P)$ has an equivalence scaling B in $S(r, c)$.*

(iii) *The set* $\Pi(P) \cap S(r, c)$ *is nonempty.*

(iv) *If* I *and* J *are subsets of* $\langle m \rangle$ *and* $\langle n \rangle$, *respectively, such that* $P_{IJ} = 0$, *then* $r(I) \leqq c(J^c)$ *with equality holding if and only if* $P_{I^c,J^c} = 0$.

The following two theorems characterize solvability of sum equivalence scaling and sum symmetric scaling for subpatterns of a given pattern matrix P, respectively.

THEOREM 3. *Let* P *be an* $m \times n$ *pattern matrix, and let* $r \in \mathfrak{R}^m$ *and* $c \in \mathfrak{R}^n$ *be strictly positive. Then the following are equivalent*:

(i) *For some subpattern* P' *of* P, *each* $A \in \Pi(P')$ *has an equivalence scaling in* $S(r, c)$.

(ii) *For some subpattern* P' *of* P, *some matrix* $A \in \Pi(P')$ *has an equivalence scaling in* $S(r, c)$.

(iii) *For some subpattern* P' *of* P, *the set* $\Pi(P') \cap S(r, c)$ *is nonempty.*

(iv) *If* I *and* J *are subsets of* $\langle m \rangle$ *and* $\langle n \rangle$, *respectively, such that* $P_{IJ} = 0$, *then* $r(I) \leqq c(J^c)$.

Proof. The equivalence of (i), (ii), and (iii) is immediate from Theorem 2 applied to a corresponding subpattern P' of P. A proof of the equivalence of (iii) and (iv) of Theorem 3, which is simpler than the one given in [14], can be found in [15]. □

THEOREM 4. *Let* P *be an* $n \times n$ *symmetric pattern matrix, and let* $r \in \mathfrak{R}^n$ *be strictly positive. Then the following are equivalent*:

(i) *For some symmetric subpattern* P' *of* P, *each symmetric* $A \in \Pi(P')$ *has a symmetric scaling in* $S(r, c)$.

(ii) *For some symmetric subpattern* P' *of* P, *some symmetric* $A \in \Pi(P')$ *has a symmetric scaling in* $S(r, c)$.

(iii) *For some symmetric subpattern* P' *of* P, *the set* $\Pi(P') \cap S(r)$ *is nonempty.*

(iv) *If* I *and* J *are subsets of* $\langle n \rangle$ *such that* $P_{IJ} = 0$, *then* $r(I) \leqq r(J^c)$.

(v) *If* I *and* J *are subsets of* $\langle n \rangle$ *such that* $P_{IJ} = 0$, *then* $r(I \cap J) \leqq r((I \cup J)^c)$.

(vi) *If* $\{K, L, M\}$ *is any partition of* $\langle n \rangle$ *such that* $P_{K,K\cup L} = 0$, *then* $r(K) \leqq r(M)$.

Proof. The equivalence of (i), (ii), and (iii) follows directly from Theorem 1, applied to a corresponding subpattern P' of P.

(iii) $\Rightarrow$ (iv): If (iii) is satisfied for subpattern P' of P and I, $J \subseteq \langle n \rangle$ with $P_{IJ} = 0$, then $P'_{IJ} = 0$, and it follows from the implication (iii) $\Rightarrow$ (iv) of Theorem 1 that $r(I) \leqq r(J^c)$, and therefore (iv) holds.

(iv) $\Leftrightarrow$ (v) $\Rightarrow$ (vi): These implications follow from the arguments used to show the analogous implications of Theorem 1.

(vi) $\Rightarrow$ (v): Assume that (vi) holds and that I, $J \subseteq \langle n \rangle$ with $P_{IJ} = 0$. Then $P_{I\cap J,J} = 0$, and by the symmetry of P we have $P_{I\cap J,I\setminus J} = [P_{I\setminus J,I\cap J}]^T = 0$; therefore, $P_{I\cap J,I\cup J} = 0$. Applying condition (vi) to the partition $\{K, L, M\}$ with $K = I \cap J$, $L = (I\setminus J) \cup (J\setminus I)$, and $M = (I \cup J)^c$, it follows that $r(I \cap J) = r(K) \leqq r(M) = r((I \cup J)^c)$.

(iv) $\Rightarrow$ (iii): We prove this implication using a modification of the technique used in the proof of Theorem 3.7 in [4]. Suppose that (iv) holds. It follows from the implication (iv) $\Rightarrow$ (iii) of Theorem 3 with $m = n$ and $c = r$ that for some subpattern Q (which need not be symmetric) of P there exists a matrix $C \in \Pi(Q)$ satisfying (2) with $c = r$. Then $B = \frac{1}{2}(C + C^T)$ satisfies (1) and $B \in \Pi(P')$, where $P' = \frac{1}{2}(Q' + Q^T)$ is a symmetric subpattern of P.

We observe that it suffices to prove that for some subpattern P' (which need not be symmetric) of P there exists a matrix $B \in \Pi(P')$ satisfying (2) with $c = r$ (and, consequently, $m = n$). This follows because if B is such a matrix, then $B' = \frac{1}{2}(B + B^T)$ satisfies (1) and $B' \in \Pi(Q)$, where $Q = \frac{1}{2}(P' + (P')^T)$ is a symmetric subpattern of P. Therefore, the implication follows from the implication (iv) $\Rightarrow$ (iii) of Theorem 3. □

4. Problem statement and existence conditions. We present our main result giving eight equivalent conditions characterizing the existence of a solution for max symmetric scaling.

For a positive vector $r = (r_1, \ldots, r_n)^T \in \Re^n$, let $M(r)$ denote the set of all $n \times n$ nonnegative matrices $A = [a_{ij}]$ such that

$$\max_{j \in \langle n \rangle} a_{ij} = r_i \quad \text{for } i \in \langle n \rangle. \tag{3}$$

Also for a vector $r \in \Re^n$ and scalar $\alpha \in \Re$, let the *α-level set of r*, denoted lev (r, α), be the set $\{i \in \langle n \rangle : r_i \geqq \alpha\}$. Finally, if we have that A is an $n \times n$ symmetric matrix and $I = \text{lev}\,(r, \alpha) \neq \phi$, we call A_{II} an *r-upper principal submatrix* of A.

THEOREM 5. *Let P be an $n \times n$ symmetric pattern matrix, and let $r \in \Re^n$ be strictly positive. Then the following are equivalent*:

(i) *Each symmetric $A \in \Pi(P)$ has a symmetric scaling in $M(r)$.*

(ii) *Some symmetric $A \in \Pi(P)$ has a symmetric scaling in $M(r)$.*

(iii) *The set $\Pi(P) \cap M(r)$ is nonempty.*

(iv) *The set $\Pi(P') \cap M(r)$ is nonempty for some pattern matrix P' satisfying $P' \leqq P$.*

(v) *If $P_{IJ} = 0$ for subsets $I, J \subseteq \langle n \rangle$, then*

$$\max_{i \in I} r_i \leqq \max_{j \in J^c} r_j. \tag{4}$$

(vi) *If $P_{IJ} = 0$ for subsets $I, J \subseteq \langle n \rangle$, then*

$$\max_{i \in I \cap J} r_i \leqq \max_{j \in (I \cup J)^c} r_j. \tag{5}$$

(vii) *If $\{K, L, M\}$ is any partition of $\langle n \rangle$ such that $P_{K,K \cup L} = 0$, then*

$$\max_{i \in K} r_i \leqq \max_{i \in M} r_i. \tag{6}$$

(viii) *If $P_{iJ} = 0$ for $J \subseteq \langle n \rangle$ and $i \in J$, then*

$$r_i \leqq \max_{j \in J^c} r_j. \tag{7}$$

(ix) *No upper r-principal submatrix of P has a zero row.*

Proof. The implication (i) $\Rightarrow$ (ii) is trivial because $\Pi(P)$ is nonempty ($P \in \Pi(P)$), and the implication (ii) $\Rightarrow$ (iii) is straightforward because $DAD \in \Pi(P)$ whenever $A \in \Pi(P)$ and D is a positive diagonal matrix. Also, the implication (iii) $\Rightarrow$ (iv) is trivial.

(iv) $\Rightarrow$ (v): Let $A \in \Pi(P') \cap M(r)$ for some pattern matrix $P' \leqq P$, and let I and J be nonempty subsets of $\langle n \rangle$ such that $P_{IJ} = 0$, and therefore $P'_{IJ} = A_{IJ} = 0$. Because r_i is the maximum of the entries in the ith row and A is symmetric, it follows directly that

$$\max_{i \in I} r_i = \max_{i \in I} \max_{j \in \langle n \rangle} a_{ij} = \max_{i \in I} \max_{j \in J^c} a_{ij} = \max_{j \in J^c} \max_{i \in I} a_{ij} \leqq \max_{j \in J^c} r_j.$$

(v) $\Rightarrow$ (vi): Let I and J satisfy the assumptions of condition (vi). Then $P_{I \cap J, I \cup J} = 0$ (see the proof of (vi) $\Rightarrow$ (v) in Theorem 4) and (5) follows by applying (4) to the sets $I \cap J$ and $I \cup J$.

(vi) $\Rightarrow$ (vii): If K, L, and M satisfy the assumptions of condition (vii), (6) follows by applying (5) to the sets $I = K$ and $J = K \cup L$ and observing that $K \cap (K \cup L) = K$ and $[K \cup (K \cup L)]^c = M$.

(vii) $\Rightarrow$ (viii): If i and J satisfy the assumption of condition (vii), then $\{\{i\}, J \backslash \{i\}, J^c\}$ is a partition of $\langle n \rangle$ satisfying the assumptions of condition (viii), and (7) follows by applying (6).

(viii) $\Rightarrow$ (ix): Suppose that for some $\alpha \in \Re$ and $J = \{j \in \langle n \rangle \mid r_j \geqq \alpha\} \neq \varnothing$, the ith row of the upper principal submatrix P_{JJ} is zero. Then, $i \in J \subseteq \langle n \rangle$, $P_{iJ} = 0$, and $r_i \geqq \alpha > r_j$ for $j \in J^c$. This violates condition (viii) for the partition $\{\{i\}, J \setminus \{i\}, J^c\}$ and therefore proves the implication.

(ix) $\Rightarrow$ (i): We prove this implication constructively in § 5 by exhibiting an algorithm that for a given nonnegative symmetric matrix A and positive vector r either finds a symmetric scaling of A in $M(r)$ or finds an upper principal submatrix of A containing a zero row. □

The following observations compare max symmetric scaling and sum symmetric scaling. First, note that feasibility conditions (iii) and (iv) in Theorem 5 are equivalent. By contrast, there is no such equivalence for sum symmetric scaling, and the assertion that for some subpattern P' of P there is a matrix A in $\Pi(P')$ satisfying (1) is not equivalent to condition (iii) of Theorem 1. In fact Rothblum and Schneider [14] provide separate characterizations of each of these two conditions. As a result of the equivalence of conditions (iii) and (iv) of Theorem 5, conditions (v), (vi), and (vii) of Theorem 5 are simpler than the analogous conditions (iv), (v), and (vi) of Theorem 1. For example, for the nonequivalence for sum scaling consider

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad r = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Then $P \in M(r)$, but there is no symmetric scaling of P whose rows sums are $(1, 1)^T$.

Second, as a consequence of properties of the max operation, the *set* conditions (v), (vi), and (vii) of Theorem 5 are equivalent to the simple *point* conditions (vii) and (viii). No analogous simplification is possible for sum symmetric scaling.

Third, a solution for sum symmetric scaling is unique, whereas a solution for max symmetric scaling need not be unique. For example, let

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \quad \text{and} \quad r = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Then the general symmetric scaling of A that is in $M(r)$ is given by

$$B = \begin{pmatrix} \alpha & 0 \\ 0 & (2\alpha)^{-1} \end{pmatrix} A \begin{pmatrix} \alpha & 0 \\ 0 & (2\alpha)^{-1} \end{pmatrix} = \begin{pmatrix} \alpha^2 & 1 \\ 1 & (4\alpha^2)^{-1} \end{pmatrix},$$

where $\frac{1}{2} \leqq \alpha \leqq 1$.

5. The algorithm. We describe an algorithm for max symmetric scaling. For a given nonnegative symmetric matrix A and strictly positive vector r, our algorithm either finds a symmetric scaling of A in $M(r)$ or certifies that no such scaling exists by showing that condition (ix) of Theorem 5 is violated.

THE MAX SYMMETRIC SCALING ALGORITHM

Input: An $n \times n$ nonnegative symmetric matrix A and a strictly positive vector $r \in \Re^n$.

Output: Either a positive diagonal matrix D such that $DAD \in M(r)$ or a subset $J \subseteq \langle n \rangle$ and an index $i \in J$ such that A_{JJ} is an r-upper principal submatrix of A with zero row A_{iJ}.

Step 0 (Initialization): Let α_i for $i \in \langle m \rangle$ be the distinct values in the vector r listed in decreasing order. That is,

$$\{\alpha_i \mid i \in \langle m \rangle\} = \{r_i \mid i \in \langle n \rangle\} \quad \text{and} \quad \alpha_1 > \alpha_2 > \cdots > \alpha_m.$$

Set $k = 1$, $J = \varnothing$, and $d_i = 1$ for $i = 1, 2, \ldots, n$.

Step 1 (Push down): Set

$$I = \{i \in \langle n \rangle \mid r_i = \alpha_k\} \quad \text{and} \quad J = \{i \in \langle n \rangle \mid r_i \geqq \alpha_k\}. \tag{8}$$

If $A_{iJ} = 0$ for some $i \in I$, output (J, i) and STOP; otherwise, set the values d_i for $i \in I$ so that $d_i \leqq 1$ and

$$d_i a_{ij} d_j \leqq \alpha_k \quad \text{for } i \in I \quad \text{and} \quad j \in J. \tag{9}$$

Step 2 (Pull up): Select any $i \in I$ and set $I = I \backslash i$. Set

$$d_i = \min \left\{ \min_{\substack{j \in J \\ j \neq i}} \left\{ \frac{\alpha_k}{a_{ij} d_j} \right\}, \sqrt{\frac{\alpha_k}{a_{ii}}} \right\}. \tag{10}$$

If $I \neq \varnothing$, repeat Step 2.

Step 3 (Termination): If $k < m$, replace k by $k + 1$ and return to Step 1; if $k = m$, then output $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$ and STOP.

We observe that whenever the algorithm does not stop in Step 1, then A_{IJ} has no zero row and we can achieve (9) by setting

$$d_i = \min \left\{ \min_{j \in J} \left\{ \frac{\alpha_k}{a_{ij}} \right\}, 1 \right\} \quad \text{for } i \in I. \tag{11}$$

The following lemma is crucial for our analysis.

LEMMA 6. *During the Max Symmetric Scaling Algorithm, after each execution of Step* 2, *we have*

$$d_i a_{ij} d_j \leqq r_i \quad \textit{for } i, j \in J \tag{12}$$

and

$$\max_{j \in J} d_i a_{ij} d_j = r_i \quad \textit{for } i \in J \backslash I. \tag{13}$$

Furthermore, the d's are nondecreasing throughout consecutive executions of Step 2.

Proof. We first show that the d's are nondecreasing and that if (12) and (13) hold at the beginning of an execution of Step 2, then they also hold at the end of that execution. Let s be the element selected out of I for the execution of Step 2, let d' and I' be, respectively, the values of d and I at the beginning of the execution of the current Step 2, and let d'' and I'' be the values of d and I at the end of that execution. Thus, we are assuming that (12) and (13) hold for $d = d'$ and $I = I'$. Now the selection of s and the definition of d''_s ensure that

$$\max_{j \in J} d''_s a_{sj} d''_j = \alpha_k = r_s. \tag{14}$$

As $d''_j = d'_j$ for $j \in J \setminus \{s\}$ and as the specialization of (12) with $d = d'$ to $i = s$ ensures that we have

$$d'_s a_{sj} d'_j \leqq r_s \quad \text{for all } j \in J,$$

we conclude that $d''_s \geqq d'_s$. As the remaining coordinates of d are unchanged, it follows that $d'' \geqq d'$.

We next establish (12) and (13) with $d = d''$ and $I = I''$. First, if $i \in J \setminus \{s\}$ and $j \in J \setminus \{s\}$, then $d''_i a_{ij} d''_j = d'_i a_{ij} d'_j \leqq r_s$. Also, (14) ensures that $d''_s a_{sj} d''_j \leqq r_s$ for all $j \in J$. Next, for $i \in J$ we have from the symmetry of A, from (14), and from the fact that $r_s =$

$\alpha_k = \min\{r_j \mid j \in J\}$ that

$$d_i'' a_{is} d_s'' = d_s'' a_{si} d_i'' \leqq r_s \leqq r_i,$$

completing the proof that (12) holds for $d = d''$. Further, as it is assumed that (13) holds for $d = d'$ and $I = I'$, as $d'' \geqq d'$, and as we have seen that (12) holds for $d = d''$, we conclude that (13) is also valid with $d = d''$ and $I = I'$. This fact combines with (14) to show that

$$\max_{j \in J} d_i'' a_{ij} d_j'' = r_i \quad \text{for } i \in J \backslash I'' = (J \backslash I') \cup \{s\}. \tag{15}$$

That is, (13) holds for $d = d''$ and $I = I''$.

It remains to show that (12) and (13) hold upon each entrance of Step 2 from Step 1. This fact is obvious for the first entrance of Step 2 from Step 1 because then $J = I$; hence, (12) follows from (9) and (13) is vacuous. Next, assume that (12) and (13) hold for the kth entrance of Step 2 from Step 1 and consider the $(k+1)$st entrance, assuming that Step 1 leads to Step 2 rather than to termination. Our earlier arguments show that (12) and (13) will stay valid throughout consecutive iterations of Step 2; hence, they will hold at the $(k+1)$st entrance into Step 1. Let d', J', and $I' = \varnothing$ be the values of d, J, and I upon the $(k+1)$st entrance into Step 1, and let d'', J'', and I'' be the updated values of d, J, and I after the $(k+1)$st execution of Step 1. In particular, $J'' \backslash I'' = J' \backslash I' = J'$, and (12) and (13) hold for $d = d'$, $I = I'$, and $J = J'$. Further, as $d_j'' = d_j'$ for $j \in J'' \backslash I'' = J'$, we have from the validity of (13) for $d = d'$, $J = J'$, and $I = I'$ that for $i \in J'' \backslash I'' = J' \backslash I'$

$$\begin{aligned} \max_{j \in J''} d_i'' a_{ij} d_j'' &= \max\left\{\max_{j \in J'} d_i' a_{ij} d_j', \max_{j \in I''} d_i'' a_{ij} d_j''\right\} \\ &= \max\left\{r_i, \max_{j \in I''} d_i'' a_{ij} d_j''\right\}. \end{aligned} \tag{16}$$

Now, for $j \in I''$, $r_j = \alpha_{k+1} = \min\{r_j \mid j \in J''\}$; hence, the symmetry of A and (9) with $d = d''$, $J = J''$, and $I = I''$ imply that for $j \in I'$

$$d_i'' a_{ij} d_j'' = d_j'' a_{ji} d_i'' \leqq \alpha_{k+1} \leqq \alpha_k = r_i. \tag{17}$$

Combining (16) and (17) we conclude that

$$\max_{j \in J''} d_i'' a_{ij} d_j'' = r_i \quad \text{for all } i \in J'' \backslash I''.$$

That is, (13) holds for $d = d''$, $J = J''$, and $I = I''$. This fact and (9) ensure that (12) holds as well. Thus, both (12) and (13) hold at the end of $(k+1)$st execution of Step 1 and therefore at the entrance to Step 2. □

THEOREM 7. *If the Max Symmetric Scaling Algorithm is executed with input $A \in \mathfrak{R}^{n \times n}$ and $r \in \mathfrak{R}^n$, then either the algorithm terminates in Step 3 with a positive diagonal matrix D such that $DAD \in \Pi(r)$ or the algorithm terminates in Step 1 with $J \subseteq \langle n \rangle$ and $i \in J$ such that A_{JJ} is an r-upper principal submatrix of A with $A_{iJ} = 0$. The algorithm requires $O(n \ln n + p)$ comparisons, $O(p)$ multiplications and divisions, and $O(q)$ square root calculations, where p and q are, respectively, the number of nonzero elements and the number of nonzero diagonal elements of the matrix A.*

Proof. The algorithm must terminate as Step 1 is executed at most m times, and each execution of Step 2 reduces $|I|$. If the algorithm terminates in Step 1, then A_{iJ} is a zero row of the r-upper principal submatrix A_{JJ}. It follows from Lemma 6 that each time Step 3 is executed, we have $(DAD)_{JJ} \in \Pi(r_J)$ because $I = \varnothing$ when Step 3 is exe-

cuted. Therefore, if the algorithm terminates in Step 3, then $J = \langle n \rangle$ and $(DAD)_{JJ} = DAD \in \Pi(r)$.

We finally determine the complexity of the algorithm. With $O(n \ln n)$ comparisons, we can sort the values $\{ r_i \mid i \in \langle n \rangle \}$ as required in Step 0. In total, Step 1 can be performed with p comparisons and p divisions, and Step 2 can be performed with $p - n$ comparisons, $p - q$ multiplications, p divisions, and q square root calculations. □

The following example shows that the square root calculations cannot, in general, be eliminated. Let

$$A = [1] \in \Re^{1 \times 1} \quad \text{and} \quad r = (2) \in \Re^1.$$

Then the only scaling DAD of A that is in $\Pi(r)$ has $D = \sqrt{2}$. If the diagonal elements of A are all zero, then because the square-root operation in (9) can be omitted, the algorithm can be executed over any linearly ordered group (multiplicative) Abelian group with zero, that is, a group G together with an element 0 such that $a0 = 0 = 0a$ for any $a \in G$. In particular, if the underlying matrix is nonnegative, then the output elements will be in any subgroup that contains the input elements. The above example shows that this conclusion need not hold when A has nonzero diagonal elements.

We note that a diagonal element a_{ii} is considered twice in the course of an execution of the algorithm. If $r_i = \alpha_k$, we have $a_{ii} \to a_{ii} d_i^2$ in the kth execution of Step 1, and then in one of the following executions of Step 2, $\sqrt{\alpha_k / a_{ii}} = \sqrt{r_i / a_{ii}}$ is determined and is compared with other numbers to update d_i. Thus, the square rooting can be avoided if each original a_{ii} is the product of r_i and the square of a known number. Consequently, the square rooting can be avoided in the "decision problem" where one determines whether or not there exists a scaling corresponding to a given vector r and matrices in a given pattern P. This is achieved by testing any matrix A in $\Pi(P)$ with $a_{ii} = r_i$ for all i with $P_{ii} \neq 0$.

6. Equivalence scaling. We apply our results for max symmetric scaling to max equivalence scaling.

For strictly positive $r = (r_1, \ldots, r_m)^T \in \Re^m$ and $c = (c_1, \ldots, c_n)^T \in \Re^n$, let $M(r, c)$ denote the set of all $m \times n$ nonnegative matrices $A = [a_{ij}]$ such that

$$\max_{j \in \langle n \rangle} a_{ij} = r_i \quad \text{for } i \in \langle m \rangle \quad \text{and} \quad \max_{i \in \langle m \rangle} a_{ij} = c_j \quad \text{for } j \in \langle n \rangle. \tag{18}$$

In the following theorem we characterize the existence of a solution for the max equivalence scaling by reducing it to max symmetric scaling.

THEOREM 8. *Let P be an $m \times n$ pattern matrix, and let $r \in \Re^m$ and $c \in \Re^n$ be strictly positive. Then the following are equivalent*:

(i) *Some $A \in \Pi(P)$ has an equivalence scaling in $M(r, c)$.*
(ii) *Each $A \in \Pi(P)$ has an equivalence scaling in $M(r, c)$.*
(iii) *The set $\Pi(P) \cap M(r, c)$ is nonempty.*
(iv) *The set $\Pi(P') \cap M(r, c)$ is nonempty for some pattern matrix $P' \leqq P$.*
(v) *The vectors r and c satisfy*

$$\max_{i \in \langle m \rangle} r_i = \max_{j \in \langle n \rangle} c_j; \tag{19}$$

furthermore, if $P_{IJ} = 0$ for subsets $I \subseteq \langle m \rangle$ and $J \subseteq \langle n \rangle$, then

$$\max_{i \in I} r_i \leqq \max_{j \in J^c} c_j \quad \textit{and} \quad \max_{j \in J} c_j \leqq \max_{i \in I^c} r_i. \tag{20}$$

(vi) *The following conditions hold*:

(a) *The vectors r and c satisfy* (19).

(b) *If* $P_{iJ} = 0$ *for* $i \in \langle m \rangle$ *and* $J \subseteq \langle n \rangle$, *then*

$$r_i \leqq \max_{j \in J^c} c_j.$$

(c) *If* $P_{Ij} = 0$ *for* $I \subseteq \langle m \rangle$ *and* $j \in \langle n \rangle$, *then*

$$c_j \leqq \max_{i \in I^c} r_i.$$

(vii) *The vectors r and c satisfy* (19), *and for all* $\alpha \in \Re$ *and subsets* $I = \text{lev}(\alpha, r) \subseteq \langle m \rangle$ *and* $J = \text{lev}(\alpha, c) \subseteq \langle n \rangle$, *if* $I, J \neq \varnothing$, *then the submatrix* P_{IJ} *contains neither a zero row nor a zero column.*

Proof. The implications (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (iv) follow from the arguments used to establish the corresponding implications in Theorem 5.

(iv) $\Rightarrow$ (v): Suppose that $A \in \Pi(P') \cap M(r, c)$ for some pattern matrix $P' \leqq P$. Then

$$\max_{i \in \langle m \rangle} r_i = \max_{i \in \langle m \rangle} \max_{j \in \langle n \rangle} a_{ij} = \max_{j \in \langle n \rangle} \max_{i \in \langle m \rangle} a_{ij} = \max_{j \in \langle n \rangle} c_j.$$

Furthermore, if $P_{IJ} = 0$ for some $I \subseteq \langle m \rangle$ and $J \subseteq \langle n \rangle$, then

$$\max_{i \in I} r_i = \max_{i \in I} \max_{j \in J^c} a_{ij} = \max_{j \in J^c} \max_{i \in I} a_{ij} \leqq \max_{j \in J^c} c_j.$$

A symmetric argument proves the second inequality in (20).

(v) $\Rightarrow$ (vi): This implication is trite because parts (b) and (c) of condition (vi) are the specializations of the second part of condition (v) for the cases of $I = i$ and $J = j$, respectively.

(vi) $\Rightarrow$ (vii): Suppose that for some $\alpha \in \Re$, $I = \text{lev}(\alpha, r) \neq \varnothing$ and $J = \text{lev}(\alpha, c) \neq \varnothing$. It follows that

$$r_i \geqq \alpha > \max_{j \in J^c} c_j \quad \text{for } i \in I$$

and

$$c_j \geqq \alpha > \max_{i \in I^c} r_i \quad \text{for } j \in J.$$

Therefore, if P_{IJ} has a zero row or a zero column, we get a violation of parts (b) or (c), respectively, of condition (vi). The implication (vi) $\Rightarrow$ (vii) now follows because the first assertion of (vii) is the same as part a of (vi).

(vii) $\Rightarrow$ (i): For an $m \times n$ nonnegative matrix A, define the $(m + n) \times (m + n)$ matrix A' and the vector $r' \in \Re^{m+n}$ by

$$A' = \left(\begin{array}{c|c} 0 & A \\ \hline A^T & 0 \end{array}\right) \quad \text{and} \quad r' = \begin{pmatrix} r \\ c \end{pmatrix}. \tag{21}$$

It is straightforward to show that max symmetric scaling with input A' and r' has a solution $D' = \text{diag}(d'_1, d'_2, \ldots, d'_{m+n})$ if and only if max equivalence scaling for A, r, and c has a solution $D = \text{diag}(d'_1, d'_2, \ldots, d'_m)$ and $E = \text{diag}(d'_{m+1}, \ldots, d'_{m+n})$. We conclude that condition (i) of Theorem 8 is equivalent to condition (i) of Theorem 5 applied to A' and r'. It is straightforward to show that condition (vii) of Theorem 8 is

equivalent to condition (vii) of Theorem 5 when applied to A' and r'. Therefore, the equivalence of (i) and (vii) follows from Theorem 5. □

We observe that the reduction of max equivalence scaling to max symmetric scaling in the proof of Theorem 8 shows that max equivalence scaling can be solved by the Max Symmetric Scaling Algorithm. Moreover, the diagonal elements of the matrix A' defined in (21) are all zero. Therefore, it follows from the complexity analysis at the end of § 5 that max equivalence scaling can be solved using $O((n+m)\ln(n+m)+p)$ comparisons and $O(p)$ multiplications and divisions, where p is the number of nonzero elements of the matrix A. Further, we observe that max equivalence scaling can be solved over any linearly ordered Abelian group with zero. The example given in § 5 shows that in general max symmetric scaling does not have this property. Also, sum equivalence scaling does not have this property. For example, let

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad r = c = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The only equivalence scaling of A with row and column sums all equality 1 is the matrix

$$B = \begin{pmatrix} \sqrt{2}-1 & 2-\sqrt{2} \\ 2-\sqrt{2} & \sqrt{2}-1 \end{pmatrix} = \begin{pmatrix} 1-2^{-1}\sqrt{2} & 0 \\ 0 & \sqrt{2}-1 \end{pmatrix} A \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 1 \end{pmatrix}.$$

We note that when a solution to max equivalence scaling exists, it need not be unique. For example, let

$$A = \begin{pmatrix} 4 & 1 \\ 2 & 4 \end{pmatrix} \quad \text{and} \quad r = c = \begin{pmatrix} 4 \\ 4 \end{pmatrix}.$$

Then the general equivalence scaling B of A that is in $M(r, c)$ is given by

$$B = \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} A \begin{pmatrix} 1 & 0 \\ 0 & \alpha^{-1} \end{pmatrix} = \begin{pmatrix} 4 & \alpha^{-1} \\ 2\alpha & 4 \end{pmatrix},$$

where $\frac{1}{4} \leqq \alpha \leqq 2$. By contrast, the corresponding equivalence in sum equivalence scaling is unique.

The results about max equivalence scaling were derived from results about max symmetric scaling. Historically, a reverse logic has been applied in the sum case as results about sum equivalence scaling are used to establish results about sum symmetric scaling (see Brualdi [2] and Csima and Datta [7]). The latter arguments use uniqueness (up to multiplicative scalar) of diagonal matrices D and E for which DAE has prescribed row sum vector r and column sum vector c. Hence, if A is symmetric and $r = c$, the fact that DAE and $EA^TD = EAD$ have the same row and column sums can be used to argue that (with proper normalization) $D = E$. But, as we have seen above, no such uniqueness results are available in the max case.

7. Graphs. We observe that max symmetric scaling and the corresponding solvability theorem have an equivalent undirected graph statement. An (*undirected*) graph is an ordered pair $G = (V, E)$, where V is a finite set of *vertices* and E is a set of *edges* composed of unordered pairs of vertices. Given such a graph $G = (V, E)$ and a vertex $v \in V$, we let $N(v)$ denote the set of neighbors of v, i.e., $N(v) \equiv \{u \in V : \{u, v\} \in E\}$. In particular, we say that v is *isolated* if $N(v) = \phi$. Note that by definition a graph may contain *loops* but may not contain repeated edges. For subsets $S, T \subseteq V$, we use $[S, T]$ to denote the set of edges $\{u, v\} \in E$ with $u \in S$ and $v \in T$ or $u \in T$ and $v \in S$.

A *weight function* for a graph $G = (V, E)$ is a real-valued function f defined on the edges E. For convenience, in this case we write f_{uv} for $f(\{u, v\})$. A *weighted graph* is a triple (V, E, f), where (V, E) is a graph and f is a weight function for G. A *potential* for G is a real-valued function defined on the vertices V. For a nonempty subset W of V, we define the *subgraph induced by* W to be the graph (W, E'), where E' contains all edges of E of the form $e = \{u, v\}$ for vertices $u, v \in W$.

Next, we define a mapping Φ from the set of symmetric nonnegative matrices to the set of weighted graphs. For an $n \times n$ symmetric nonnegative matrix $A = [a_{ij}]$, we define the mapping Φ by

$$A \xrightarrow{\Phi} (V, E, f), \tag{22}$$

where

$$V = \langle n \rangle,$$
$$E = \{\{i, j\} \mid a_{ij} > 0\}, \quad \text{and}$$
$$f_{ij} = \ln a_{ij} \quad \text{for } \{i, j\} \in E.$$

It is easy to see that Φ is a bijection between the set of nonnegative symmetric matrices and the set of weighted graphs.

We state the following lemma without proof.

LEMMA 9. *Let A be an $n \times n$ symmetric nonnegative matrix, and let (V, E, f) be the corresponding weighted graph under the mapping Φ in* (22). *Let $r \in \mathfrak{R}^n$ be strictly positive, and let s be the potential defined by $s_i = \ln r_i$ for $i \in \langle n \rangle$. Then the following are equivalent*:

(i) *There exists a positive diagonal matrix $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$ such that $DAD \in M(r)$.*

(ii) *There exists a potential p such that*

$$\max_{u \in N(v)} \{p_u + f_{uv} + p_v\} = s_v \quad \text{for } v \in V. \tag{23}$$

Furthermore, D and p are related by $p_u = \ln d_u$ for $u \in \langle n \rangle = V$.

The next theorem follows directly from Theorem 5 and the correspondence between matrices and graphs described in Lemma 9.

THEOREM 10. *Let $G = (V, E)$ be a graph, and let s be a potential for G. Then the following are equivalent*:

(i) *For every weight function f for G there exists a potential p satisfying* (23).

(ii) *For some weight function f for G there exists a potential p satisfying* (23).

(iii) *There exists some weight function f for G such that*

$$\max_{u \in N(v)} f_{uv} = s_v \quad \text{for } v \in V. \tag{24}$$

(iv) *There exists $E' \subseteq E$ and some weight function $f: E' \mapsto \mathfrak{R}$ such that*

$$\max_{u \in N'(v)} f_{uv} = s_v \quad \text{for } v \in V,$$

where $N'(v) = \{u \in V: \{u, v\} \in E'\}$.

(v) *If $[S, T] = \varnothing$ for $S, T \subseteq V$, then*

$$\max_{v \in S} s_v \leqq \max_{v \in T^c} s_v.$$

(vi) *If* $[S, T] = \varnothing$ *for* $S, T \subseteq V$, *then*

$$\max_{v \in S \cap T} s_v \leqq \max_{v \in (S \cup T)^c} s_v.$$

(vii) *If* $\{S, T, U\}$ *is any partition of* V *such that* $[S, S \cup T] = \varnothing$, *then*

$$\max_{v \in S} s_v \leqq \max_{v \in U} s_v.$$

(viii) *Let* $W \subseteq V$. *If* $v \in W$ *is an isolated vertex of the subgraph induced by* W, *then*

$$s_v \leqq \max_{u \in W^c} s_u.$$

(ix) *For every* $\alpha \in \Re$, *the subgraph of* G *induced by the level set* lev (s, α) *has no isolated vertex*.

We observe that Theorem 8 also has an equivalent graph formulation in terms of bipartite graphs. We have omitted the details because they are straightforward.

8. pth power scaling. For $0 \leqq p \leqq \infty$ and $x \in \Re^n$, we define the l_p norm of x by $\|x\|_p$. We consider the problem of determining a symmetric scaling of a given nonnegative symmetric matrix such that the rows of the resulting matrix have prescribed l_p norms. Of course, the cases of $p = 1$ and $p = \infty$ reduce to sum and max symmetric scaling, respectively. Here, we show that the cases of $0 < p < \infty$ can be reduced easily to the case of $p = 1$.

For an $m \times n$ nonnegative matrix A and $0 < p < \infty$, the pth *Hadamard power of* A, written $A^{(p)}$, is the matrix whose ijth entry is $(a_{ij})^p$. Let A_i denote the ith row of the matrix A. For a strictly positive vector $r \in \Re^n$ and $0 < p \leqq \infty$, let $S^p(r)$ denote the set of all $n \times n$ nonnegative symmetric matrices B such that $\|B_i\|_p = r_i$ for each $i \in \langle n \rangle$.

It is easily seen that $B \in S^p(r)$ if and only if $B^{(p)} \in S(r^{(p)})$. Moreover, $B = DAD$ if and only if $B^{(p)} = D^{(p)} A^{(p)} D^{(p)}$. Thus, as an immediate consequence of Theorem 1, we obtain the following theorem.

THEOREM 11 (l_p-symmetric scaling). *Let* P *be an* $n \times n$ *symmetric pattern matrix, let* $r \in \Re^n$ *be strictly positive, and let* $p \in \Re$ *with* $0 < p < \infty$. *Then the following are equivalent*:

(i) *Each symmetric* $A \in \Pi(P)$ *has a symmetric scaling* $B \in S^p(r)$.

(ii) *Some symmetric* $A \in \Pi(P)$ *has a symmetric scaling* $B \in S^p(r)$.

(iii) *The set* $\Pi(P) \cap S^p(r)$ *is nonempty*.

(iv) *If* $P_{IJ} = 0$ *for* $I, J \subseteq \langle n \rangle$, *then* $\|r_I\|_p \leqq \|r_{J^c}\|_p$ *with equality holding if and only if* $P_{I^c J^c} = 0$.

Conditions (v) and (vi) of Theorem 1 can also be extended in the obvious way. Further, analogous results can also be derived for the cases of $-\infty \leqq p < 0$.

REFERENCES

[1] L. M. BREGMAN, *Proof of the convergence of Sheleikhovskii's method for a problem with transportation constraints*, U.S.S.R. Comput. Math. and Math. Phys. 1 (1967), pp. 191–204.

[2] R. A. BRUALDI, *Convex sets of non-negative matrices*, Canad. J. Math., 20 (1968), pp. 144–157.

[3] ———, *The DAD theorem for arbitrary row sums*, Proc. Amer. Math. Soc., 45 (1974), pp. 189–194.

[4] ———, *Combinatorial properties of symmetric non-negative matrices*, in Colloquio Internazionale sulle Teorie Combinatorie, Accademia Nazional de Lincei, Rome, 1976, pp. 99–120.

[5] R. A. BRUALDI, S. PARTER, AND H. SCHNEIDER, *The diagonal equivalence of a nonnegative matrix to a stochastic matrix*, J. Math. Anal. Appl., 16 (1966), pp. 31–50.

[6] V. CHVÁTAL, *Linear Programming*. W. H. Freeman and Company, New York, 1983.
[7] J. CSIMA AND B. N. DATTA, *The DAD theorem for symmetric non-negative matrices*, J. Combin. Theory Ser. A, 12 (1972), pp. 147–152.
[8] D. GALE, *A theorem on flows in networks*, Pacific J. Math., 7 (1957), pp. 1073–1082.
[9] A. J. HOFFMAN, *Some recent applications of the theory of linear inequalities to extremal combinatorial analysis*, Combinatorial Analysis, in R. Bellman and M. Hall, Jr., eds., American Mathematical Society, Providence, RI, 1960, pp. 113–127.
[10] J. KRUITHOF, *Telefoonverkeersrekening*, De Ingenieur, 3 (1937), pp. 15–25.
[11] A. W. MARSHALL AND I. OLKIN, *Scaling of matrices to achieve specified row and column sums*, Numer. Math., 12 (1968), pp. 83–90.
[12] M. V. MENON, *Matrix links, an extremisation problem and the reduction of a nonnegative matrix to one with prescribed row and column sums*, Canad. J. Math., 20 (1968), pp. 225–232.
[13] M. V. MENON AND H. SCHNEIDER, *The spectrum of a nonlinear operator associated with a matrix*, Linear Algebra Appl., 2 (1969), pp. 321–334.
[14] U. G. ROTHBLUM AND H. SCHNEIDER, *Scaling of matrices which have prespecified row sums and column sums via optimization*, Linear Algebra Appl., 114/115 (1989), pp. 737–764.
[15] U. G. ROTHBLUM, H. SCHNEIDER, AND M. H. SCHNEIDER, *Scaling matrices to prescribed row and column maxima*, Rutgers Research Report, Rutgers University, New Brunswick, NJ, 1990.
[16] M. H. SCHNEIDER AND S. ZENIOS, A comparative study of algorithms for matrix balancing, *Oper. Res.*, 38 (1990), pp. 439–455.
[17] R. SINKHORN, *A relationship between arbitrary positive matrices and doubly stochastic matrices*, Ann. Math. Statist., 35 (1964), pp. 876–879.
[18] R. SINKHORN AND P. KNOPP, *Concerning nonnegative matrices and doubly stochastic matrices*, Pacific J. Math., 212 (1967), pp. 343–348.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 15–58, January 1994

002

A COMPLETED THEORY OF THE UNSYMMETRIC LANCZOS PROCESS AND RELATED ALGORITHMS, PART II*

MARTIN H. GUTKNECHT†

Dedicated to the memory of Heinz Rutishauser (1918–1970).

Abstract. This paper is a continuation of Part I [M. H. Gutknecht, *SIAM J. Matrix Anal. Appl.*, 13 (1992), pp. 594–639], where the theory of the "unsymmetric" Lanczos biorthogonalization (BO) algorithm and the corresponding iterative method BIORES for non-Hermitian linear systems was extended to the nongeneric case. The analogous extension is obtained here for the biconjugate gradient (or BIOMIN) method and for the related BIODIR method. Here, too, the breakdowns of these methods can be cured. As a preparation, mixed recurrence formulas are derived for a pair of sequences of formal orthogonal polynomials belonging to two adjacent diagonals in a nonnormal Padé table, and a matrix interpretation of these recurrences is developed. This matrix interpretation leads directly to a completed formulation of the progressive qd algorithm, valid also in the case of a nonnormal Padé table. Finally, it is shown how the cure for exact breakdown can be extended to near-breakdown in such a way that (in exact arithmetic) the well-conditioned formal orthogonal polynomials and the corresponding Krylov space vectors do not depend on the threshold specifying the near-breakdown.

Key words. Lanczos algorithm, biconjugate gradient algorithm, BIOMIN, BIODIR, breakdown, formal orthogonal polynomial, recurrence, Padé approximation, staircase, quotient difference algorithm, qd algorithm

AMS subject classifications. 65F10, 30E05, 41A21, 65F15

Introduction. In Part I [13] we derived a number of basic results on sequences of *formal orthogonal polynomials* of the first and second kind (FOP1s and FOP2s, respectively). Given a linear functional $\Phi : \mathcal{P} \to \mathbb{C}$ defined on the space $\mathcal{P}$ of complex polynomials by[1]

$$\Phi_l(z^k) := \phi_{k+l} \quad (k \in \mathbb{N}), \tag{1.1}$$

there is a finite or infinite sequence $\{n_j\}_{j=0}^{J}$ ($J \leq \infty$) of indices with $0 =: n_0 < n_1 < n_2 < \cdots (< n_J \text{ if } J < \infty)$ for which a *regular* (monic) FOP1 $P_{n_j} := P_{l;n_j}$ exists. By definition, these are those values of the index n for which a unique monic polynomial $P_n := P_{l;n}$ of exact degree n satisfying

$$\Phi_l(pP_n) = 0 \quad (\forall p \in \mathcal{P}_{n-1}) \tag{1.2}$$

exists. These indices are also characterized by the nonsingularity of the $n \times n$ moment matrix

$$\mathbf{M}_n := \mathbf{M}_{l;n} := \begin{bmatrix} \phi_l & \phi_{l+1} & \cdots & \phi_{l+n-1} \\ \phi_{l+1} & \phi_{l+2} & \cdots & \phi_{l+n} \\ \vdots & \vdots & & \vdots \\ \phi_{l+n-1} & \phi_{l+n} & \cdots & \phi_{l+2n-2} \end{bmatrix}. \tag{1.7}$$

* Received by the editors October 2, 1990; accepted for publication (in revised form) April 22, 1992.

† Interdisciplinary Project Center for Supercomputing, Eidgenössische Technische Hochschule Zürich, ETH-Zentrum, CH-8092 Zürich, Switzerland (mhg@ips.ethz.ch).

[1] Equations copied from Part I are numbered as before. Most of those that are explicitly used in this part are recalled in the introduction. Also note that Part I ended with §4 and that Part II, for the sake of continuity, begins its numbering with §5.

Starting from these regular FOP1s P_{n_j} we have obtained a full sequence $\{P_n\}_{n=0}^{\infty} := \{P_{l;n}\}_{n=0}^{\infty}$ of monic FOP1s by setting

$$P_n(z) := W_{n-n_j}(z)P_{n_j}(z) \quad \text{if } n_j \le n < n_{j+1} =: n_j + h_j. \tag{1.28}$$

Here W_{n-n_j} could be an arbitrary monic polynomial of exact degree $n - n_j$, but in view of actual implementations we have primarily considered the case where W_{n-n_j} is the $n - n_j$ element of a fixed sequence $\{W_m\}$ of monic polynomials satisfying a three-term recurrence

$$W_{m+1}(z) = (z - \alpha_m^W)\ W_m(z) - \beta_m^W\ W_{m-1}(z) \quad (m \in \mathbb{N}) \tag{2.10}$$

(with $W_0(z) :\equiv 1$, $W_{-1}(z) :\equiv 0$, $\beta_0^W := 0$).

The FOP1s P_n of (1.28) satisfy the formal orthogonality conditions

$$\Phi_l(pP_n) = 0 \quad (\forall p \in \mathcal{P}_{\hat{n}-1}), \quad \Phi_l(z^{\hat{n}}P_n) \ne 0, \quad \text{where } \hat{n} := n_j + n_{j+1} - n - 1. \tag{2.1}$$

In particular, when P_n is regular, $n = n_j$, then

$$\hat{n} = n_j + h_j - 1 = n_{j+1} - 1. \tag{2.2}$$

Equivalently, assuming $n_i \le n' \le n_{i+1}$ and $n_j \le n \le n_{j+1}$, we can write

$$\Phi_l(P_{n'}P_n) = 0 \quad \begin{array}{l}\text{if } i \ne j \\ \text{or } i = j \quad \text{and} \quad n' + n < n_j + n_{j+1} - 1,\end{array} \tag{2.5}$$

$$\Phi_l(P_{n'}P_n) =: \delta_j \ne 0 \quad \text{if } i = j \quad \text{and} \quad n' + n = n_j + n_{j+1} - 1, \tag{2.6}$$

where δ_j is independent of $n - n_j$ and $n' - n_j$.

The formula (2.10) implies that the nonregular FOP1s P_n ($n_j < n < n_{j+1}$) can be generated according to

$$P_{n+1}(z) = (z - \alpha_{n-n_j}^W)\, P_n(z) - \beta_{n-n_j}^W\, P_{n-1}(z), \qquad n_j \le n \le n_{j+1} - 2. \tag{2.11}$$

(Likewise, any other recurrence for $\{W_m\}$ leads to one for those P_n.) Less trivial is the fact that the orthogonality relation (2.1) allows us to establish for the regular FOP1s a three-term recurrence

$$P_{n_{j+1}}(z) = (W_{h_j}(z) - a_j(z))P_{n_j}(z) - \beta_j P_{n_{j-1}}(z), \quad j = 0, \ldots, J-1, \tag{2.17}$$

with a monic polynomial coefficient $W_{h_j} - a_j \in \mathcal{P}_{h_j}$ (hence, $a_j \in \mathcal{P}_{h_j-1}$) and a scalar coefficient $\beta_j \in \mathbb{C}$. (The initial values are: $P_{n_{-1}}(z) :\equiv 0$, $P_{n_0}(z) :\equiv 1$, $\beta_0 := 0$.) For the coefficient β_j there is an explicit formula and for the coefficients of $a_j(z) = \sum_{s=0}^{h_j-1} \alpha_{s,j} W_s(z)$ we have found a recursive formula based on the solution of a lower triangular system:

$$\beta_j \Phi_l(P_{n_j-1}P_{n_{j-1}}) = \Phi_l(zP_{n_j-1}P_{n_{j+1}-1}), \tag{2.23a}$$

$$\begin{aligned}\Phi_l(a_j P_{n_j+k}P_{n_j}) = {} & \Phi_l(zP_{n_j+k}P_{n_{j+1}-1}) - \alpha_{h_j-1}^W \Phi_l(P_{n_j+k}P_{n_{j+1}-1}) \\ & - \beta_{h_j-1}^W \Phi_l(P_{n_j+k}P_{n_{j+1}-2}), \quad k = 0, \ldots, h_j - 1.\end{aligned} \tag{2.23b}$$

After introducing the infinite row vector $\mathbf{p} := [P_0, P_1, \ldots]$, we can express the recurrences (2.11) and (2.17) as

$$z\mathbf{p}(z) = \mathbf{p}(z)\mathbf{H}, \tag{3.11}$$

where the Gragg matrix $\mathbf{H}$ is an infinite block tridiagonal unit upper Hessenberg matrix

$$\mathbf{H} := \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_1 & & & \\ \mathbf{C}_0 & \mathbf{A}_1 & \mathbf{B}_2 & & \\ & \mathbf{C}_1 & \mathbf{A}_2 & \ddots & \\ & & \ddots & \ddots & (\mathbf{B}_J) \\ & & & (\mathbf{C}_{J-1}) & (\mathbf{A}_J) \end{bmatrix}. \tag{3.6}$$

Under the assumption (2.10), the diagonal blocks $\mathbf{A}_j$ are $h_j \times h_j$ comrade matrices containing on the diagonal and the first superdiagonal the coefficients α_m^W and β_m^W from (2.10), and in the last column the coefficients $\alpha_{j,i}$ of the polynomial a_j. The off-diagonal blocks $\mathbf{C}_j$ and $\mathbf{B}_j$ are zero except for the element in the upper right corner, which is 1 in $\mathbf{C}_j$ and β_j in $\mathbf{B}_j$. If $J < \infty$, $\mathbf{B}_J$ is the $h_{J-1} \times \infty$ zero matrix, and $\mathbf{A}_J$ is the infinite tridiagonal matrix $\mathbf{T}^W$ representing the recurrence (2.10) in the form $z\mathbf{w}(z) = \mathbf{w}(z)\mathbf{T}^W$ (where $\mathbf{w} := [W_0, W_1, \ldots]$).

By using matrix notation, we can express the orthogonality properties (2.5)–(2.6) in compact form:

$$\Phi_l(\mathbf{p}^T\mathbf{p}) = \mathbf{D}, \tag{3.22}$$

where $\mathbf{D}$ is a block diagonal matrix whose blocks $\mathbf{D}_j$ are $h_j \times h_j$ lower right triangular matrices with all antidiagonal elements equal to δ_j.

The FOP1s P_n and the associated FOP2s Q_n ($n \in \mathbb{N}$) are essentially the denominators and the numerators, respectively, of the proper parts of the Padé approximant lying on the lth diagonal of the Padé table of the formal Laurent series

$$f(z) = \sum_{k=-\infty}^{\infty} \phi_k z^k. \tag{1.25}$$

More exactly, the $(m,n) := (l+n-1, n)$ Padé approximant of f is equal to

$$r_{m,n}(z) := \sum_{k=-\infty}^{m-n} \phi_k z^k + z^{m-n} \frac{Q_n(z^{-1})}{P_n(z^{-1})},$$

cf. (1.21), (1.22), and (1.34). The rational function $r_{m,n}$ is the (m,n) entry of the Padé table.

An important feature of the Padé table is its block structure: Identical entries occur in finite or infinite square blocks, cf. Corollary 1.6. The regular FOP1s belong to entries on the first row or the first column of such a square block. This Block Structure Theorem is important in this second part, where we now consider pairs of sequences of FOP1s, $\{P_n\}_{n=0}^{\infty} := \{P_{l;n}\}_{n=0}^{\infty}$ and $\{P'_n\}_{n=0}^{\infty} := \{P_{l+1;n}\}_{n=0}^{\infty}$, which belong to *two adjacent diagonals* of the Padé table. We mark the quantities corresponding to the second sequence by a prime, writing for example $\mathbf{M}'_n$, $\mathbf{H}'$, $\mathbf{p}'$. We also set $\Phi := \Phi_l$ and $\Phi' := \Phi_{l+1}$. Note that this usage of primes differs from the one in Part I, where they indicated quantities belonging to the FOP2s, the polynomials of the second kind.

In §5 we define a new sequence of regular FOP1s whose elements are alternatively taken from the two above-mentioned sequences and belong to all distinct Padé approximants that lie on the two diagonals. We call the corresponding sequence of

Padé approximants a *block staircase sequence*. In analogy to the three-term recurrence (2.17) for the regular FOP1s that belong to one diagonal, we derive a pair of three-term recurrence formulas (with one polynomial coefficient in each) for the new sequence of FOP1s. Since FOP1s from both diagonals appear in these formulas, we call them *mixed* recurrences. Two equivalent but different matrix formulations for them are given in §6. Actually, these matrix formulations involve all the polynomials from the two sequences $\{P_n\}_{n=0}^{\infty}$ and $\{P_n'\}_{n=0}^{\infty}$ and not just the regular ones.

By eliminating either the first or the second sequence, we rediscover in §7 the matrix formulations of the separate recurrences for the second and the first sequence, respectively, which are both of the type discussed in Part I, i.e., they are determined by Gragg matrices of the form (3.6). It turns out that the Gragg matrix of the second sequence is obtained from the one of the first sequence by executing one step of a *block LR algorithm*, i.e., we have to compute a particular block LU decomposition and then multiply the factors together in reverse order. The factors, which are block bidiagonal (but none of which is chosen with unit block diagonal), are exactly the matrices that describe the mixed recurrences of the block staircase. This block LR algorithm generalizes Rutishauser's LR algorithm for tridiagonal matrices, and hence also his (equivalent) qd algorithm [23]. It is the key to a *nongeneric progressive qd algorithm*, which, in contrast to the classical (generic) progressive qd algorithm, never breaks down in exact arithmetic.

In §4 of Part I we applied the results on (diagonal) sequences of FOP1s to the unsymmetric Lanczos process. Let $\mathbf{A} : \mathcal{H} \to \mathcal{H}$ be a bounded linear operator mapping a separable real or complex Hilbert space into itself. The standard inner product in $\mathcal{H}$ is denoted by $\langle .\,,.\rangle$, but we use instead a formal inner product $\langle .\,,.\rangle_{\mathbf{B}}$ defined by $\langle \mathbf{y}\,,\mathbf{x}\rangle_{\mathbf{B}} := \langle \mathbf{y}\,,\mathbf{B}\mathbf{x}\rangle$, which is induced by another bounded linear operator $\mathbf{B} : \mathcal{H} \to \mathcal{H}$ that commutes with $\mathbf{A}$. (The cases of practical interest are $\mathbf{B} = \mathbf{I}$, $\mathbf{B} = \mathbf{A}$, and $\mathbf{B} = \mathbf{A}^{-1}$.) Orthogonality with respect to this indefinite inner product is referred to as formal orthogonality. Associated with $\mathbf{A}$, $\mathbf{x}_0$, $\mathbf{y}_0$, and this inner product are the *Schwarz constants* or *moments*

$$\phi_k := \langle \mathbf{y}_0, \mathbf{A}^k\mathbf{x}_0\rangle_{\mathbf{B}} := \langle \mathbf{y}_0, \mathbf{B}\mathbf{A}^k\mathbf{x}_0\rangle \quad (k \in \mathbb{N}). \tag{4.1}$$

The link to the above-described theory of FOPs is based on the identification of these moments with the values that the linear functional $\Phi = \Phi_0$ of (1.1) takes on the monomials.

Starting from $\mathbf{A}$, $\mathbf{x}_0$, $\mathbf{y}_0$, the classical (generic) *Lanczos biorthogonalization (BO) algorithm* [19], [15], [11] generates the two sequences $\{\mathbf{x}_n\}_{n=0}^{\nu-1}$ and $\{\mathbf{y}_n\}_{n=0}^{\nu-1}$ such that for $n = 0, 1, \ldots, \nu - 1$

$$\mathbf{x}_n \in \mathcal{K}_{n+1} := \text{span}\,(\mathbf{x}_0, \mathbf{A}\mathbf{x}_0, \mathbf{A}^2\mathbf{x}_0, \ldots, \mathbf{A}^n\mathbf{x}_0), \tag{4.6a}$$

$$\mathbf{y}_n \in \mathcal{L}_{n+1} := \text{span}\,(\mathbf{y}_0, \mathbf{A}^H\mathbf{y}_0, (\mathbf{A}^H)^2\mathbf{y}_0, \ldots, (\mathbf{A}^H)^n\mathbf{y}_0), \tag{4.6b}$$

and

$$\langle \mathbf{y}_m, \mathbf{x}_n\rangle_{\mathbf{B}} \begin{cases} = 0 & \text{if} \quad m \neq n, \\ \neq 0 & \text{if} \quad m = n. \end{cases} \tag{4.7}$$

In view of (4.6), $\mathbf{x}_n$ must be equal to a polynomial in $\mathbf{A}$ times $\mathbf{x}_0$, and $\mathbf{y}_n$ must be equal to a polynomial in $\mathbf{A}^H$ times $\mathbf{y}_0$. From the orthogonality condition (4.7) and the uniqueness of the regular FOPs it follows easily that actually

$$\mathbf{x}_n = P_n(\mathbf{A})\mathbf{x}_0\Gamma_n, \qquad \mathbf{y}_n = \overline{P_n}(\mathbf{A}^H)\mathbf{y}_0\bar{\Gamma}_n, \tag{4.14}$$

where P_n is the monic regular FOP1 of degree n, $\overline{P_n}$ is the polynomial with the complex conjugate coefficients, and Γ_n and $\bar{\Gamma}_n$ are scale factors. As we know, such a regular FOP1 need not exist, and that is when (4.7) no longer holds and the generic BO algorithm breaks down. Our remedy for this breakdown was to use (4.14), with P_n being any FOP1 of degree n (regular or not). The *nongeneric BO algorithm* (Algorithm 1) was then obtained by translating the recurrences for the FOP1s via (4.14) to recurrences for $\mathbf{x}_n$ and $\mathbf{y}_n$. In these recurrences the scale factors Γ_n and $\bar{\Gamma}_n$ are replaced by the relative scale factors

$$\gamma_{n,i} := \Gamma_n/\Gamma_{n-i}\,, \quad \bar{\gamma}_{n,i} := \bar{\Gamma}_n/\bar{\Gamma}_{n-i} \quad (n \in \mathbb{N}, i \in \mathbb{N}). \tag{4.20a}$$

The application of the BO algorithm to solving linear systems of equations $\mathbf{Ax} = \mathbf{b}$ is based on defining a sequence of approximants $\mathbf{z}_n$ in such a way that $\mathbf{x}_n$ is the residual vector for $\mathbf{z}_n$,

$$\mathbf{x}_n = \mathbf{b} - \mathbf{A}\mathbf{z}_n, \quad n = 0, 1, 2, \ldots, \tag{4.61}$$

(*normalized* BIORES *algorithm*) or such that $\mathbf{x}_n$ is the residual of $\mathbf{z}_n$ in a system with scaled right-hand side

$$\mathbf{x}_n = \mathbf{b}\rho_n - \mathbf{A}\mathbf{z}_n, \quad n = 0, 1, 2, \ldots, \tag{4.62}$$

(*unnormalized* BIORES *algorithm*).

The BO algorithm terminates when $\mathbf{x}_n = \mathbf{0}$ or $\mathbf{y}_n = \mathbf{0}$. But while the generic BO algorithm (and thus also BIORES) breaks down seriously whenever $\langle \mathbf{y}_n, \mathbf{x}_n \rangle_{\mathbf{B}} = 0$, our generalization fails only if the inner product vanishes for *all* n beyond some bound n_J. This is then called an incurable breakdown [22], [21]. Unfortunately, in order to detect such an incurable breakdown, we theoretically have to work with exact arithmetic and to iterate until n reaches the rank of $\mathbf{A}$. When the algorithm is applied to solving a linear system, the hope is that in theory $\mathbf{x}_n = \mathbf{0}$ for some n, and that in practice the residual $\mathbf{x}_n$ is sufficiently small even much earlier. There is, however, the additional difficulty that $\mathbf{y}_n = \mathbf{0}$ causes the algorithm to stop, and this requires that we find a nonzero replacement for $\mathbf{y}_n$ that is orthogonal to $\mathcal{K}_n$ and can be used to proceed.

Here, we discuss in §8 three iterative linear system solvers that are closely related to the unnormalized nongeneric BIORES algorithm and have in fact the same breakdown behavior. (For the generic versions this is not true [11].) The first two, *normalized* and *unnormalized nongeneric* BIOMIN, are extensions of the well-known and highly successful *biconjugate gradient (BCG) method* [20], [7], [11]. The third is (*normalized*) *nongeneric* BIODIR, which, independently, has also been developed by Joubert [17], [18]. These three methods generate relevant subsequences of essentially the same sequences of approximants $\mathbf{z}_n$, residuals $\mathbf{x}_n$, and corresponding $\mathbf{B}$-biorthogonal vectors $\mathbf{y}_n$ as nongeneric BIORES; but additionally they produce two $\mathbf{BA}$-biorthogonal sequences $\{\mathbf{u}_n\}$ and $\{\mathbf{v}_n\}$. The elements of the first serve as *direction vectors*, i.e., they specify the direction of the correction for $\mathbf{z}_n$. While the vectors $\mathbf{x}_n$ and $\mathbf{y}_n$ correspond according to (4.14) to a diagonal sequence of FOP1s, the vectors $\mathbf{u}_n$ and $\mathbf{v}_n$ correspond similarly to the FOP1s on the adjacent diagonal. And while nongeneric BIORES is based on the recurrence for the first block diagonal sequence of FOP1s, nongeneric BIOMIN is based on the recursion for the block staircase sequence, and nongeneric BIODIR makes use of the one for the block diagonal sequence on the adjacent diagonal.

Finally, in §§9 and 10, we present a theoretically clean approach to treating near-breakdowns. Of course, such an approach is of great importance in practice, where

the occurrence of exact breakdown is very unlikely, but near-breakdown may cause severe numerical effects. We formulate this theory for polynomials, but the application to the above-mentioned iterative linear solvers is straightforward. (The actual implementation still requires a careful treatment of many nontrivial details. This is the subject of joint work with Roland Freund and Noël Nachtigal [8].)

This treatment of near-breakdown is based on defining appropriate *clusters* or *blocks* of polynomials in such a way that formal orthogonality is maintained between the blocks, but not within the blocks. The first polynomial in each block is still a regular FOP1, and, moreover, it is well conditioned if the blocks are suitably chosen. To construct these polynomials (or the corresponding sequences of Krylov space vectors) we apply a block orthogonalization process, which is just the appropriate generalization of the Gram–Schmidt process.[2] The resulting algorithm, which is described here in terms of polynomials and in [8] in terms of Krylov space vectors, can be considered as a generalization of the nongeneric BO algorithm of §4 and of the similar algorithms proposed by Parlett, Taylor, and Liu [22], [21], [24] and by Boley et al. [1]. Although mainly exact breakdowns were considered in [22], [24], we suggest applying Parlett's adjective *"look-ahead,"* which is by now well established, to the near-breakdown versions of all of the above-mentioned algorithms, while the adjective *"nongeneric"* should be reserved for the versions curing exact breakdown.

In Parlett, Taylor, and Liu [22] the discussion was actually restricted to 2×2 blocks. Several options of block LDU decomposition of the moment matrix were considered for this case, but the resulting generalizations of the Lanczos algorithm differ in detail considerably from the proposals made here, even for exact breakdown. In particular, the left Lanczos vectors are chosen differently; thus, in the relations (4.14) the polynomial $\overline{P_n}$ is there in general *not* the complex conjugate polynomial of P_n. Moreover, as we will see in §9, the above-mentioned "appropriate generalization" of the Gram–Schmidt process for near-breakdowns is not the straightforward one, which would not yield a "block three-term" recurrence. Finally, here we not only treat block diagonal sequences (§9), but also block staircase sequences (§10), which present some additional difficulties.

Upon revision of this paper we learned of nongeneric algorithms developed by Hegedüs [14] for applying conjugate gradients to a particular indefinite problem. From his treatment one must conclude that he was probably also aware of the possibility of developing some of the nongeneric algorithms given here; but he did not specify them.

5. Block staircase sequences and corresponding recurrences. In this section we consider two *adjacent* sequences of FOP1s, $\{P_n\} := \{P_{l;n}\}$ and $\{P'_n\} := \{P_{l+1;n}\}$, and their associated sequences of FOP2s, $\{Q_n\} := \{Q_{l;n}\}$ and $\{Q'_n\} := \{Q_{l+1;n}\}$, and derive a recurrence for a particular sequence formed alternatively from regular elements of $\{P_n\}$ and $\{P'_n\}$. We denote those elements of this sequence that are taken from $\{P_n\}$ by $P_{n_j^\wedge}$, and those taken from $\{P'_n\}$ by $P'_{n_j^\vee}$. The index sequences

$$\{n_j^\wedge\}_{j=0}^{J^\wedge} \quad \text{and} \quad \{n_j^\vee\}_{j=0}^{J^\vee}$$

still indicate the degrees of the corresponding polynomials and are subsequences of the two index sequences $\{n_j\}$ and $\{n'_j\}$, respectively, that belong to the regular FOP1s in

[2] We use the notion "block orthogonal" here, although there is a danger of confusion with the different meaning the word "block" has in "block Lanczos" or "block Gram–Schmidt."

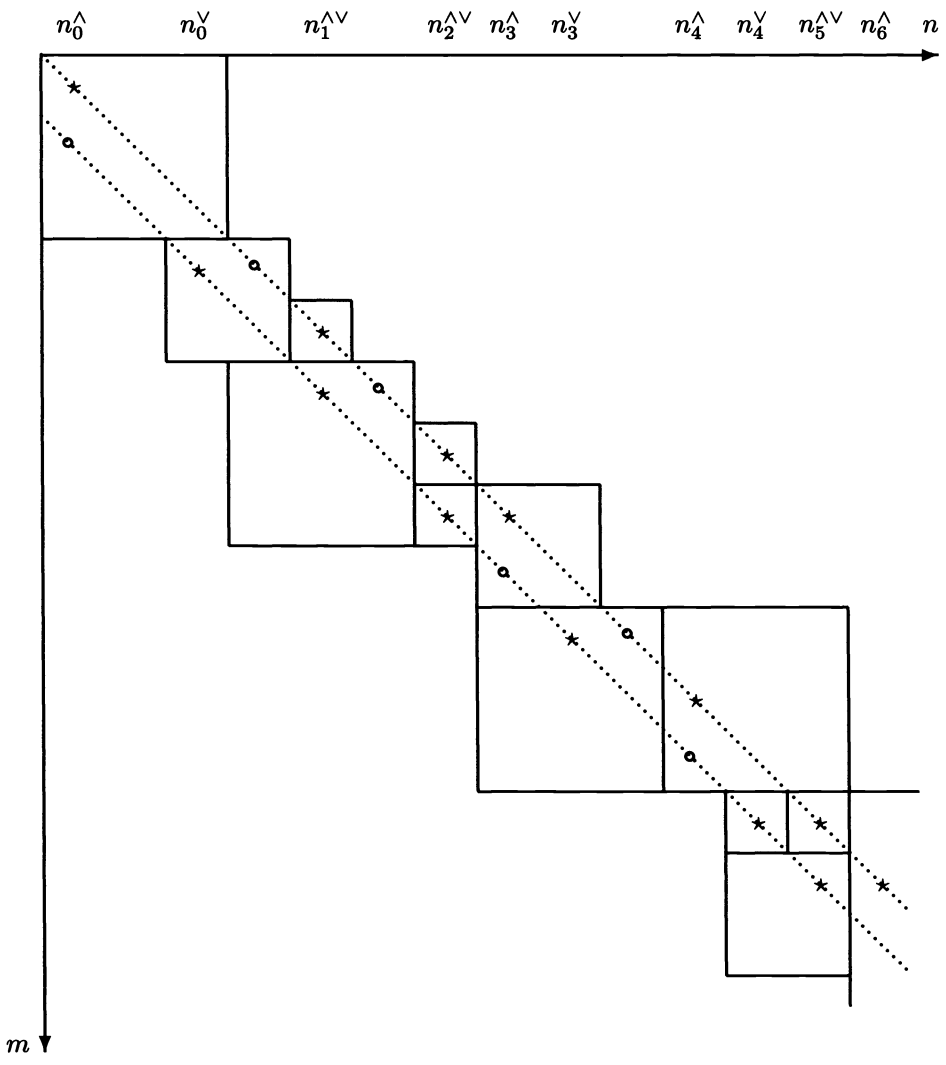


FIG. 3. *A block staircase sequence. The elements of the staircase are marked by* $\star$*, while the other regular elements on the two adjacent diagonals are marked by* $\circ$*. The notation* $n_j^{\wedge\vee}$ *means* $n_j^\wedge = n_j^\vee$.

$\{P_n\}$ and $\{P'_n\}$, respectively. These subsequences are chosen such that $n_0^\wedge := 0$ and

$$n_j^\wedge \le n_j^\vee \ (j = 0, \ldots, J^\vee), \quad n_j^\vee < n_{j+1}^\wedge \ (j = 0, \ldots, J^\wedge - 1), \tag{5.1}$$

and such that they contain as many indices as possible; in the cases where the interlacing condition (5.1) does not determine $n_j^\vee$ or $n_{j+1}^\wedge$ uniquely, we make the last of the choices allowed by (5.1). From the Block Structure Theorem 1.6 for the Padé table, it is seen that such ambiguities occur in connection with blocks that contain elements from both $\{P_n\}$ and $\{P'_n\}$, i.e., which are intersected both by the upper and the lower diagonal (on which $m - n = l$ and $m - n = l + 1$, respectively), cf. Fig. 3. In such cases, either the first column or the first row of the block contains regular elements out of both $\{P_n\}$ and $\{P_{l+1;n}\}$, and then the lower or the right, respectively,

of these elements is dropped, while the other becomes an element of $\{P_{n_j^\wedge}\}$ or $\{P'_{n_j^\vee}\}$, respectively. Note that

$$J^\vee \le J^\wedge \le J^\vee + 1 \le \infty\,, \tag{5.2}$$

i.e., there can be at most one more marked element on the upper diagonal, and the number of elements may be infinite.

Let us also set

$$m_j^\wedge := n_j^\wedge + l - 1\,, \quad m_j^\vee := n_j^\vee + l\,, \tag{5.3}$$

$$h_j^\wedge := n_j^\vee - n_j^\wedge + 1 = m_j^\vee - m_j^\wedge\,, \quad h_j^\vee := n_{j+1}^\wedge - n_j^\vee = m_{j+1}^\wedge - m_j^\vee + 1\,. \tag{5.4}$$

Both $h_j^\wedge$ and $h_j^\vee$ are ≥ 1. If $J^\wedge$, $J^\vee < \infty$, then $h_{J^\wedge}^\wedge := h_{J^\vee}^\vee := n_{J^\wedge}^\vee := n_{J^\vee+1}^\wedge := \infty$.

The index pair sequences $\{(m_j^\wedge, n_j^\wedge)\}_{j=0}^{J^\wedge}$ and $\{(m_j^\vee, n_j^\vee)\}_{j=0}^{J^\vee}$ define the *block staircase sequence* for the two adjacent diagonals of the Padé table. (For the more general situation of the Newton–Padé table, block staircase sequences have been introduced in [12].) Note that the index pairs specify for each block an entry of minimum degrees m and n on one of the two adjacent diagonals, but do not indicate the upper left corner of the block.

The orthogonality result of Theorem 2.1 yields the following lemma for such block staircase sequences.

LEMMA 5.1. *The following formal orthogonality properties hold:*

$$\Phi(pP_{n_j^\wedge}) = 0 \quad (\forall p \in \mathcal{P}_{n_j^\wedge + h_j^\wedge - 2} = \mathcal{P}_{n_j^\vee - 1})\,, \tag{5.5a}$$

$$\delta_j^\wedge := \Phi(z^{n_j^\vee} P_{n_j^\wedge}) \ne 0\,, \tag{5.5b}$$

and

$$\Phi'(pP'_{n_j^\vee}) = 0 \quad (\forall p \in \mathcal{P}_{n_j^\vee + h_j^\vee - 2} = \mathcal{P}_{n_{j+1}^\wedge - 2})\,, \tag{5.6a}$$

$$\delta_j^\vee := \Phi'(z^{n_{j+1}^\wedge - 1} P'_{n_j^\vee}) \ne 0\,. \tag{5.6b}$$

Proof. Apply (2.1) and (2.2) to the current situation and note that the regular elements following $P_{n_j^\wedge}$ and $P'_{n_j^\vee}$ on the same diagonal have the indices

$$n_j^\vee + 1 = n_j^\wedge + h_j^\wedge \quad \text{and} \quad n_{j+1}^\wedge = n_j^\vee + h_j^\vee\,, \tag{5.7}$$

respectively, independently of whether this following regular element is part of the block staircase or not. This is due to the fact that, in case of a dropped element, the index of the next element differs from that of the dropped one only by 1, as can be seen from Fig. 3. □

COROLLARY 5.2. *The following formal orthogonality properties hold:*

$$\Phi'(pP_{n_j^\wedge}) = 0 \quad (\forall p \in \mathcal{P}_{n_j^\wedge + h_j^\wedge - 3} = \mathcal{P}_{n_j^\vee - 2}), \tag{5.8a}$$

$$\delta_j^\wedge = \Phi'(z^{n_j^\vee - 1} P_{n_j^\wedge}) \ne 0, \tag{5.8b}$$

and

$$\Phi_l(zpP'_{n_j^\vee}) = 0 \quad (\forall p \in \mathcal{P}_{n_j^\vee + h_j^\vee - 2} = \mathcal{P}_{n_{j+1}^\wedge - 2})\,, \tag{5.9a}$$

$$\delta_j^\vee = \Phi_l(z^{n_{j+1}^\wedge} P'_{n_j^\vee}) \ne 0\,. \tag{5.9b}$$

Proof. In the same way that (1.2) was written as the homogeneous linear system (1.6), (5.5a) can be written as

$$\sum_{j=0}^{n} \phi_{i+j-n}\pi_{l;j,n} = 0, \quad i = l+n, \dots, l+2n+h-2, \tag{5.10}$$

where $n := n_j^{\wedge}$, $h := h_j^{\wedge}$. When the first equation is deleted, this system represents (5.8a). Both (5.5b) and (5.8b) express that the "next equation after (5.10)," with $i := l+2n+h-1$, is not homogeneous.

Similarly, (5.6a) and (5.9a) are represented by the same homogeneous linear system, and (5.6b) and (5.9b) yield the same strictly inhomogeneous equation. □

Lemma 5.1 and Corollary 5.2 allow us now to establish recurrence formulas for block staircase sequences. By analogy to the derivation of the recurrence formula for diagonal sequences in §2, we start from representations of $P_{n_{j+1}^{\wedge}}$ and $P'_{n_{j+1}^{\vee}}$ in terms of the previous elements of the staircase. Clearly, in view of (5.4), there are polynomials $t_{s,j}^{\wedge} \in \mathcal{P}_{h_s^{\wedge}-1}$ and $t_{s,j}^{\vee} \in \mathcal{P}_{h_s^{\vee}-2}$ $(s = 0, 1, \dots, j)$ such that

$$P_{n_{j+1}^{\wedge}}(z) = zW_{h_j^{\vee}-1}(z)P'_{n_j^{\vee}}(z) - \sum_{s=0}^{j} \left[zt_{s,j}^{\vee}(z)P'_{n_s^{\vee}}(z) + t_{s,j}^{\wedge}(z)P_{n_s^{\wedge}}(z) \right]. \tag{5.11}$$

(If $h_s^{\vee} = 1$, we set $t_{s,j}^{\vee} := 0$.) We multiply this relation with the monomials of degree at most $n_j^{\vee} + h_j^{\vee} - 1 = n_{j+1}^{\wedge} - 1$. Each of these monomials can be written as either $z^{n_i^{\wedge}+k}$ with $0 \le k \le h_i^{\wedge} - 2$, $0 \le i \le j$, or $z^{n_i^{\vee}+k}$ with $0 \le k \le h_i^{\vee} - 1$, $0 \le i \le j$. (Note that the range of k in the first loop is empty if $h_i^{\wedge} = 1$, in which case the multiplier $z^{n_i^{\wedge}+k}$ is not used.) Then, we apply Φ to the resulting $n_{j+1}^{\wedge}$ relations in order to obtain a linear system of $n_{j+1}^{\wedge}$ equations for the polynomials $t_{s,j}^{\wedge}$ and $t_{s,j}^{\vee}$, $s = 0, \dots, j$. In view of (5.4), these polynomials have a total of $n_{j+1}^{\wedge}$ coefficients. In view of (5.8a), all expressions $\Phi(z^{n_i^{\wedge}+k} P_{n_{j+1}^{\wedge}})$ and $\Phi(z^{n_i^{\vee}+k} P_{n_{j+1}^{\wedge}})$ vanish, so that $P_{n_{j+1}^{\wedge}}$ does not appear in the system. In order to verify the structure of this system, we list a number of results following directly from (5.4), (5.5), and (5.9); k is always assumed to lie in the given ranges.

$$\Phi(z^{n_i^{\wedge}+k+1} W_{h_j^{\vee}-1} P'_{n_j^{\vee}}) = 0 \text{ if } i \le j. \tag{5.12a}$$

(Here, for the case $i = j-1$, we have used that $n_{j-1}^{\wedge} + k \le n_{j-1}^{\wedge} + h_{j-1}^{\wedge} - 2 = n_{j-1}^{\vee} - 1 \le n_j^{\wedge} - 2 \le n_j^{\vee} - 2$.)

$$\Phi(z^{n_i^{\vee}+k+1} W_{h_j^{\vee}-1} P'_{n_j^{\vee}}) \begin{cases} = 0 & \text{if } i \le j-1, \\ \neq 0 & \text{if } i = j,\ k = 0. \end{cases} \tag{5.12b}$$

(Here, we have used that $n_{j-1}^{\vee} + h_{j-1}^{\vee} + h_j^{\wedge} - 2 = n_j^{\wedge} + h_j^{\wedge} - 2 = n_j^{\vee} - 1$.)

$$\Phi(z^{n_i^{\wedge}+k+1} t_{s,j}^{\vee} P'_{n_s^{\vee}}) = 0 \text{ if } i \le s; \tag{5.13a}$$

$$\Phi(z^{n_i^{\vee}+k+1} t_{s,j}^{\vee} P'_{n_s^{\vee}}) \begin{cases} = 0 & \text{if } i < s, \\ = 0 & \text{if } i = s,\ k + \partial t_{s,j}^{\vee} < h_s^{\vee} - 1, \\ \neq 0 & \text{if } i = s,\ k + \partial t_{s,j}^{\vee} = h_s^{\vee} - 1; \end{cases} \tag{5.13b}$$

$$\Phi(z^{n_i^\wedge+k}\, t_{s,j}^\wedge\, P_{n_s^\wedge}) \begin{cases} =0 & \text{if } i<s, \\ =0 & \text{if } i=s,\ k+\partial t_{s,j}^\wedge < h_s^\wedge - 1; \\ \neq 0 & \text{if } i=s,\ \partial t_{s,j}^\wedge > 0,\ k+\partial t_{s,j}^\wedge = h_s^\wedge - 1; \end{cases} \tag{5.13c}$$

$$\Phi(z^{n_i^\vee+k}\, t_{s,j}^\wedge\, P_{n_s^\wedge}) \begin{cases} =0 & \text{if } i<s, \\ \neq 0 & \text{if } i=s,\ \partial t_{s,j}^\wedge = 0,\ k=0. \end{cases} \tag{5.13d}$$

By capitalizing on these formulas, the above-mentioned linear system (which was obtained by multiplying (5.11), for $i = 0, 1, \ldots, j$, with $z^{n_i^\wedge+k}$ $(k = 0, \ldots, h_i^\wedge - 2)$ and $z^{n_i^\vee+k}$ $(k = 0, \ldots, h_i^\vee - 1)$, and then applying Φ) reduces to the following set of equations:

$$\sum_{s=0}^{i-1} \Phi(z^{n_i^\wedge+k+1}\, t_{s,j}^\vee\, P'_{n_s^\vee}) + \sum_{s=0}^{i} \Phi(z^{n_i^\wedge+k}\, t_{s,j}^\wedge\, P_{n_s^\wedge}) = 0\,, \qquad k = 0, \ldots, h_i^\wedge - 2;\ i = 0, \ldots, j\,; \tag{5.14a}$$

$$\sum_{s=0}^{i} \Phi(z^{n_i^\vee+k+1}\, t_{s,j}^\vee\, P'_{n_s^\vee}) + \sum_{s=0}^{i} \Phi(z^{n_i^\vee+k}\, t_{s,j}^\wedge\, P_{n_s^\wedge}) = \begin{cases} 0 \text{ if } i \le j-1, \\ \Phi(z^{n_j^\vee+k+1}\, W_{h_j^\vee-1}\, P'_{n_j^\vee}) \neq 0 \text{ if } i=j,\ k=0, \\ \Phi(z^{n_j^\vee+k+1}\, W_{h_j^\vee-1}\, P'_{n_j^\vee}) \text{ if } i=j, \end{cases} \qquad k = 0, \ldots, h_i^\vee - 1;\ i = 0, \ldots, j\,. \tag{5.14b}$$

This system is of triangular structure, like the one for diagonal sequences, which consists of (2.13) and (2.14). Again, except for the last few equations, the system is homogeneous and we can conclude that "most" of the unknown polynomials are zero. In fact, let us assume that $j > 0$ and, at first, that $t_{0,j}^\wedge \neq 0$. Then, if $\partial t_{0,j}^\wedge > 0$, the equation with $i = 0$ and $k = h_0^\wedge - \partial t_{0,j}^\wedge - 1$ in (5.14a), which is homogeneous but contains in view of (5.13c) exactly one nonzero term, yields a contradiction. Likewise, if $\partial t_{0,j}^\wedge = 0$, we let $i = k = 0$ and use (5.14b), (5.13b), and (5.13d) to obtain a contradiction. Next, suppose that $t_{0,j}^\wedge = t_{0,j}^\vee = t_{1,j}^\wedge = \cdots = t_{i-1,j}^\vee = t_{i,j}^\wedge = 0$, but $t_{i,j}^\vee \neq 0$ for some $i \le j-1$. Then, (5.14b) and (5.13b) with $k = h_i^\vee - \partial t_{i,j}^\vee - 1 \in \{1, 2, \ldots, h_i^\vee - 1\}$ lead again to a contradiction; hence $t_{i,j}^\vee = 0$. If $t_{0,j}^\wedge = t_{0,j}^\vee = t_{1,j}^\wedge = \cdots = t_{i-1,j}^\vee = 0$ for some $i \le j-1$, we can likewise conclude from (5.14a) and (5.13c) or from (5.14b), (5.13b), and (5.13d) that $t_{i,j}^\wedge = 0$. In summary, if $j > 0$, there holds

$$t_{0,j}^\wedge = t_{0,j}^\vee = t_{1,j}^\wedge = \cdots = t_{j-2,j}^\vee = t_{j-1,j}^\wedge = t_{j-1,j}^\vee = 0\,. \tag{5.15}$$

By the same arguments we conclude from (5.14a) and (5.13c) that $\partial t_{j,j}^\wedge \le 0$; from (5.14b), (5.13b), and (5.13d), by choosing $i = j$ and $k = 0$, we conclude that

$$t_{j,j}^\wedge(z) \equiv: \varphi_j^\vee \neq 0, \tag{5.16}$$

the constant $\varphi_j^\vee$ being given by

$$\varphi_j^\vee = \delta_j^\vee / \delta_j^\wedge, \quad \text{where } \delta_j^\wedge = \Phi(z^{n_j^\vee}\, P_{n_j^\wedge}),\ \ \delta_j^\vee = \Phi(z^{n_{j+1}^\wedge}\, P'_{n_j^\vee})\,. \tag{5.17a}$$

For $e_j^\vee := t_{j,j}^\vee$ we get from (5.14b) (with $i = j$) the additional $h_j^\vee - 1$ equations

$$\Phi(z^{n_j^\vee+k+1}\, e_j^\vee\, P'_{n_j^\vee}) + \varphi_j^\vee \Phi(z^{n_j^\vee+k}\, P_{n_j^\wedge}) = \Phi(z^{n_j^\vee+k+1}\, W_{h_j^\vee-1}\, P'_{n_j^\vee}), \qquad k = 1, \ldots, h_j^\vee - 1\,. \tag{5.17b}$$

If the polynomial $e_j^\vee$ is expressed in powers of z or in terms of the polynomials W_m, it follows from (5.9b) that the matrix of the resulting linear system for the coefficients is right lower triangular and regular, since its antidiagonal elements are all equal to $\delta_j^\vee \neq 0$. In case of the monomial basis, the matrix is Hankel.

Summarizing, we have shown so far that the general representation (5.11) reduces actually to a mixed three-term recurrence formula

$$P_{n_{j+1}^\wedge}(z) = [zW_{h_j^\vee - 1}(z) - ze_j^\vee(z)]P'_{n_j^\vee}(z) - \varphi_j^\vee P_{n_j^\wedge}(z), \quad j = 0, 1, \ldots, J^\wedge - 1. \tag{5.18}$$

In a completely analogous manner we can derive a mixed three-term recurrence formula for computing $P'_{n_j^\vee}$. We start from the representation

$$P'_{n_j^\vee}(z) = W_{h_j^\wedge - 1}(z)P_{n_j^\wedge}(z) - \sum_{s=0}^{j} t_{s,j}^\wedge(z)P_{n_s^\wedge}(z) - \sum_{s=0}^{j-1} t_{s,j}^\vee(z)P'_{n_s^\vee}(z) \tag{5.19}$$

with *new* polynomials $t_{s,j}^\wedge \in \mathcal{P}_{h_s^\wedge - 2}$ $(s = 0, 1, \ldots, j)$ and $t_{s,j}^\vee \in \mathcal{P}_{h_s^\vee - 1}$ $(s = 0, 1, \ldots, j-1)$. (If $h_s^\wedge = 1$, we set $t_{s,j}^\wedge := 0$.)

This time, we multiply this relation with the monomials $z^{n_i^\wedge + k}$ $(1 \leq k \leq h_i^\wedge - 1,$ $0 \leq i \leq j)$ and $z^{n_i^\vee + k}$ $(1 \leq k \leq h_i^\vee,\ 0 \leq i \leq j-1)$, which together are all the monomials with degrees between 1 and $n_j^\wedge + h_j^\wedge - 1 = n_j^\vee$. Again, we then apply Φ to both sides. Due to (5.9a), $P'_{n_j^\vee}$ does not appear in the resulting linear system of $n_j^\vee$ equations, and in view of (5.4), the total number of coefficients of $t_{s,j}^\wedge$ and $t_{s,j}^\vee$ in (5.19) is also $n_j^\vee$. To simplify the system we need the following formulas, which are analogous to (5.12) and (5.13). (Note that the ranges of k and the maximum degrees of $\partial t_{s,j}^\wedge$ and $\partial t_{s,j}^\vee$ have changed.)

$$\Phi(z^{n_i^\wedge + k} W_{h_j^\wedge - 1} P_{n_j^\wedge}) = 0 \ \text{ if } \ i \leq j - 1; \tag{5.20a}$$

$$\Phi(z^{n_i^\vee + k} W_{h_j^\wedge - 1} P_{n_j^\wedge}) \begin{cases} = 0 & \text{if} \quad i < j - 1, \\ = 0 & \text{if} \quad i = j - 1, \ k \leq h_{j-1}^\vee - 1, \\ \neq 0 & \text{if} \quad i = j - 1, \ k = h_{j-1}^\vee - 1; \end{cases} \tag{5.20b}$$

$$\Phi(z^{n_i^\wedge + k} t_{s,j}^\vee P'_{n_s^\vee}) = 0 \ \text{ if } \ i \leq s; \tag{5.21a}$$

$$\Phi(z^{n_i^\vee + k} t_{s,j}^\vee P'_{n_s^\vee}) \begin{cases} = 0 & \text{if} \quad i < s, \\ = 0 & \text{if} \quad i = s, \ k + \partial t_{s,j}^\vee < h_s^\vee, \\ \neq 0 & \text{if} \quad i = s, \ k + \partial t_{s,j}^\vee = h_s^\vee; \end{cases} \tag{5.21b}$$

$$\Phi(z^{n_i^\wedge + k} t_{s,j}^\wedge P_{n_s^\wedge}) \begin{cases} = 0 & \text{if} \quad i < s, \\ = 0 & \text{if} \quad i = s, \ k + \partial t_{s,j}^\wedge < h_s^\wedge - 1, \\ \neq 0 & \text{if} \quad i = s, \ k + \partial t_{s,j}^\wedge = h_s^\wedge - 1; \end{cases} \tag{5.21c}$$

$$\Phi(z^{n_i^\vee + k} t_{s,j}^\wedge P_{n_s^\wedge}) = 0 \ \text{ if } \ i < s. \tag{5.21d}$$

By applying these formulas, we can rewrite the linear system as

$$\text{(5.22a)}\qquad \sum_{s=0}^{i}\Phi(z^{n_i^\vee+k}\,t_{s,j}^\vee\,P'_{n_s^\vee})+\sum_{s=0}^{i}\Phi(z^{n_i^\vee+k}\,t_{s,j}^\wedge\,P_{n_s^\wedge})$$
$$=\begin{cases}0 \text{ if } i<j-1,\\ 0 \text{ if } i=j-1,\ k\le h_{j-1}^\vee-1,\\ \Phi(z^{n_{j-1}^\vee+k}\,W_{h_j^\wedge-1}\,P_{n_j^\wedge})\neq 0 \text{ if } i=j-1,\ k=h_{j-1}^\vee-1,\end{cases}$$
$$k=1,\dots,h_i^\vee;\ i=0,\dots,j-1;$$

$$\text{(5.22b)}\qquad \sum_{s=0}^{i-1}\Phi(z^{n_i^\wedge+k}\,t_{s,j}^\vee\,P'_{n_s^\vee})+\sum_{s=0}^{i}\Phi(z^{n_i^\wedge+k}\,t_{s,j}^\wedge\,P_{n_s^\wedge})$$
$$=\begin{cases}0 \text{ if } i\le j-1,\\ \Phi(z^{n_j^\wedge+k}\,W_{h_j^\wedge-1}\,P_{n_j^\wedge}) \text{ if } i=j,\end{cases}$$
$$k=1,\dots,h_i^\wedge-1;\ i=0,\dots,j.$$

Again, the system has a triangular structure. For let us assume that $j>0$ and $t_{0,j}^\wedge\neq 0$. Then (5.21c) and (5.22b) with $i=0$ and $k=h_0^\wedge-\partial t_{0,j}^\wedge-1\in\{1,2,\dots,h_0^\wedge-1\}$ yield a contradiction, showing that $t_{0,j}^\wedge=0$. By the same argument, $t_{0,j}^\wedge=t_{0,j}^\vee=\cdots=t_{i-1,j}^\vee=0$ (for some $i\le j-1$) implies that $t_{i,j}^\wedge=0$. On the other hand, assuming $t_{0,j}^\wedge=t_{0,j}^\vee=\cdots=t_{i,j}^\vee=0$, $t_{i,j}^\wedge\neq 0$ (for some $i\le j-1$), we conclude from (5.21b) and (5.22a), by choosing $k=h_i^\vee-\partial t_{i,j}^\vee\in\{1,2,\dots,h_i^\vee\}$ there, that $t_{i,j}^\vee=0$, unless $i=j-1$ and $k=h_{j-1}^\vee$. In the latter case there holds

$$\text{(5.23)}\qquad t_{j-1,j}^\vee(z)\equiv:\varphi_j^\wedge\neq 0,$$

where the constant $\varphi_j^\wedge$ is given by

$$\text{(5.24a)}\qquad \varphi_j^\wedge=\delta_j^\wedge/\delta_{j-1}^\vee,\quad \text{where } \delta_{j-1}^\vee=\Phi(z^{n_j^\wedge}P'_{n_{j-1}^\vee}),\ \ \delta_j^\wedge=\Phi(z^{n_j^\vee-1}P_{n_j^\wedge}).$$

Finally, choosing $i=j$ in (5.22b) yields a linear system of $h_j^\wedge-1$ equations for $e_j^\wedge:=t_{j,j}^\wedge$:

$$\text{(5.24b)}\qquad \Phi(z^{n_j^\wedge+k}\,e_j^\wedge\,P_{n_j^\wedge})+\varphi_j^\wedge\Phi(z^{n_j^\wedge+k}\,P'_{n_{j-1}^\vee})=\Phi(z^{n_j^\wedge+k}\,W_{h_j^\wedge-1}\,P_{n_j^\wedge}),$$
$$k=1,\dots,h_j^\wedge-1.$$

Here also, if $e_j^\wedge$ is expressed in a basis of polynomials with ascending degrees, the coefficient matrix of this system is right lower triangular and regular, with antidiagonal elements $\delta_j^\wedge\neq 0$. If the monomial basis is used, the matrix is again Hankel.

Summarizing, we get the following theorem.

THEOREM 5.3. *The regular* FOP1*s* $P_{n_j^\wedge}$, $j=0,\dots,J^\wedge$, *and* $P'_{n_j^\vee}$, $j=0,\dots,J^\vee$, *of the block staircase sequence starting at* $(0,n_0^\wedge)=(0,l)$ *satisfy a pair of mixed three-term recurrence formulas:*

$$\text{(5.25a)}\quad P'_{n_j^\vee}(z)=[W_{h_j^\wedge-1}(z)-e_j^\wedge(z)]P_{n_j^\wedge}(z)-\varphi_j^\wedge P'_{n_{j-1}^\vee}(z),\quad j=0,1,\dots,J^\vee,$$
$$\text{(5.25b)}\quad P_{n_{j+1}^\wedge}(z)=[zW_{h_j^\vee-1}(z)-ze_j^\vee(z)]P'_{n_j^\vee}(z)-\varphi_j^\vee P_{n_j^\wedge}(z),\quad j=0,1,\dots,J^\wedge-1,$$

with initial values $P'_{n_{-1}^\vee}(z):\equiv 0$, $P_{n_0^\wedge}(z):\equiv 1$, $\varphi_0^\wedge:=0$. $\{W_m\}_{m=0}^\infty$ *is an arbitrary prescribed sequence of monic polynomials of respective degree* m; $\{\varphi_j^\wedge\}_{j=0}^{J^\vee}$ *and* $\{\varphi_j^\vee\}_{j=0}^{J^\wedge-1}$

are sequences of uniquely determined nonzero complex constants, each of which is given by (5.24a) *or* (5.17a), *respectively; and* $\{e_j^\wedge\}_{j=0}^{J^\vee}$ *and* $\{e_j^\vee\}_{j=0}^{J^\wedge-1}$ *are sequences of complex polynomials of respective degree* $\partial e_j^\wedge \le h_j^\wedge - 2$ *and* $\partial e_j^\vee \le h_j^\vee - 2$, *each of which is the unique solution of a linear system* (5.24b) *or* (5.17b), *respectively.* ($e_j^\wedge(z) \equiv 0$ *if* $h_j^\wedge = 1$, *and* $e_j^\vee(z) \equiv 0$ *if* $h_j^\vee = 1$.) *The integers* $h_j^\wedge$ *and* $h_j^\vee$ *are determined by* (5.5) *and* (5.9), *respectively, i.e.,*

$$h_j^\wedge := \min\{k \in \mathbb{N}^+;\ \Phi(z^{n_j^\wedge + k - 1} P_{n_j^\wedge}) \neq 0\} \tag{5.26a}$$
$$= \min\{k \in \mathbb{N}^+;\ \Phi(W_{k-1}(P_{n_j^\wedge})^2) \neq 0\},$$

$$h_j^\vee := \min\{k \in \mathbb{N}^+;\ \Phi(z^{n_j^\vee + k} P'_{n_j^\vee}) \neq 0\} \tag{5.26b}$$
$$= \min\{k \in \mathbb{N}^+;\ \Phi(zW_{k-1}(P'_{n_j^\vee})^2) \neq 0\}.$$

Similar recurrences, which, however, involve all regular FOP1s on both diagonals, have been given by Draux [6, pp. 394–398].

As in §2 we could state as corollaries the special results obtained for the cases where the polynomials W_m satisfy a three-term recurrence and where the polynomials $e_j^\wedge$ and $e_j^\vee$ are expressed as linear combinations of these W_m. At this point, the formulation of these results is left to the reader, but in §6 we will give their matrix formulation.

Theorem 5.3 allows us to construct recursively the sequence $P_{n_0^\wedge}$, $P'_{n_0^\vee}$, $P_{n_1^\wedge}$, $P'_{n_1^\vee}$, $P_{n_2^\wedge}, \ldots$ that contains all essentially distinct regular FOP1s on two adjacent diagonals. However, as is apparent from Fig. 3, each of the sequences $\{P_{n_j^\wedge}\}_{j=0}^{J^\wedge}$ and $\{P'_{n_j^\vee}\}_{j=0}^{J^\vee}$ in general does not contain all regular FOP1s of the corresponding diagonal. If $n_j^\vee > n_j^\wedge$, then $P_{n_j^\wedge}$ is also a regular element of the lower ($^\vee$) diagonal, and if $n_{j+1}^\wedge > n_j^\vee + 1$, then $zP'_{n_j^\vee}$ is also a regular element of the upper ($^\wedge$) diagonal. To obtain two full sequences of FOP1s, we use additionally the definition (1.28):

$$P_n(z) := \begin{cases} W_{n-n_j^\wedge}(z) P_{n_j^\wedge}(z) & \text{if } n_j^\wedge \le n \le n_j^\vee \quad (= n_j^\wedge + h_j^\wedge - 1), \\ zW_{n-n_j^\vee - 1}(z) P'_{n_j^\vee}(z) & \text{if } n_j^\vee < n < n_{j+1}^\wedge \quad (= n_j^\vee + h_j^\vee); \end{cases} \tag{5.27a}$$

$$P'_n(z) := \begin{cases} W_{n-n_j^\wedge}(z) P_{n_j^\wedge}(z) & \text{if } n_j^\wedge \le n < n_j^\vee \quad (= n_j^\wedge + h_j^\wedge - 1), \\ W_{n-n_j^\vee}(z) P'_{n_j^\vee}(z) & \text{if } n_j^\vee \le n < n_{j+1}^\wedge \quad (= n_j^\vee + h_j^\vee). \end{cases} \tag{5.27b}$$

Then, clearly,

$$P'_n(z) = P_n(z) \quad \text{if } n_j^\wedge \le n < n_j^\vee, \tag{5.28a}$$
$$P_n(z) = zP'_{n-1}(z) \quad \text{if } n_j^\vee < n < n_{j+1}^\wedge. \tag{5.28b}$$

Equations (5.27) and (5.28) allow us to modify the recurrences (5.25) in several ways.

If the three-term recurrence (2.10) is assumed to hold, we have moreover

$$P_{n+1}(z) = (z - \alpha^W_{n-n_j^\wedge}) P_n(z) - \beta^W_{n-n_j^\wedge} P_{n-1}(z), \quad n_j^\wedge \le n \le n_j^\vee - 1, \tag{5.29a}$$
$$P'_{n+1}(z) = (z - \alpha^W_{n-n_j^\vee}) P'_n(z) - \beta^W_{n-n_j^\vee} P'_{n-1}(z), \quad n_j^\vee \le n \le n_{j+1}^\wedge - 2. \tag{5.29b}$$

Here, by (5.28), the terms $zP_n(z)$ and $zP'_n(z)$ on the right-hand side can be replaced by $zP'_n(z)$ and $P_{n+1}(z)$, respectively:

$$zP'_n(z) = P_{n+1}(z) + \alpha^W_{n-n_j^\wedge} P_n(z) + \beta^W_{n-n_j^\wedge} P_{n-1}(z), \quad n_j^\wedge \le n \le n_j^\vee - 1, \tag{5.30a}$$
$$P_{n+1}(z) = P'_{n+1}(z) + \alpha^W_{n-n_j^\vee} P'_n(z) + \beta^W_{n-n_j^\vee} P'_{n-1}(z), \quad n_j^\vee \le n \le n_{j+1}^\wedge - 2. \tag{5.30b}$$

It is also worth noting that the terms $e_j^\wedge(z)P_{n_j^\wedge}(z)$ and $ze_j^\vee(z)P'_{n_j^\vee}(z)$, which appear in (5.25) and are only nonzero if $h_j^\wedge > 1$ and $h_j^\vee > 1$, respectively, are, according to (5.28), equal to $e_j^\wedge(z)P'_{n_j^\wedge}(z)$ and $e_j^\vee(z)P_{n_j^\vee+1}(z)$, respectively. In view of the definitions (5.27), the terms $W_{h_j^\wedge-1}(z)P_{n_j^\wedge}(z)$ and $W_{h_j^\vee-1}(z)P'_{n_j^\vee}(z)$ can be written as $P_{n_j^\vee}(z)$ and $P'_{n_{j+1}^\wedge-1}(z)$, so that the recurrences (5.25) become

$$\text{(5.31a)}\quad P'_{n_j^\vee}(z) = P_{n_j^\vee}(z) - e_j^\wedge(z)P_{n_j^\wedge}^{(\prime)}(z) - \varphi_j^\wedge P'_{n_{j-1}}(z), \quad j = 0,1,\dots,J^\vee,$$

$$\text{(5.31b)}\quad P_{n_{j+1}^\wedge}(z) = zP'_{n_{j+1}^\wedge-1}(z) - e_j^\vee(z)P_{n_j^\vee+1}(z) - \varphi_j^\vee P_{n_j^\wedge}(z), \quad j = 0,1,\dots,J^\wedge - 1.$$

The notation $P_n^{(\prime)}$ indicates here and below that one can use either P_n or P'_n.

Finally, in view of the later application to the nongeneric BCG process, we give the analog of Theorem 2.9. By an argument based on the orthogonality relations (5.5) and (5.9), which is analogous to the one for establishing (2.22), and by using (5.27), we obtain:

$$\text{(5.32a)}\quad \varphi_j^\wedge = \delta_j^\wedge/\delta_{j-1}^\vee, \quad \text{where } \delta_{j-1}^\vee = \Phi(zP_{n_j^\wedge-1}^{(\prime)}P'_{n_{j-1}^\vee}),\ \delta_j^\wedge = \Phi(P_{n_j^\vee}P_{n_j^\wedge}),$$

$$\text{(5.32b)}\quad \Phi(e_j^\wedge P_{n_j^\wedge+k}P_{n_j^\wedge}) = \Phi(P_{n_j+k}P_{n_j^\vee}), \quad k = 1,\dots,h_j^\wedge - 1;$$

$$\text{(5.32c)}\quad \varphi_j^\vee = \delta_j^\vee/\delta_j^\wedge, \quad \text{where } \delta_j^\wedge = \Phi(P_{n_j^\vee}^{(\prime)}P_{n_j^\wedge}),\ \delta_j^\vee = \Phi(zP_{n_{j+1}^\wedge-1}^{(\prime)}P'_{n_j^\vee}),$$

$$\text{(5.32d)}\quad \Phi(ze_j^\vee P'_{n_j^\vee+k}P'_{n_j^\vee}) = \Phi(zP'_{n_j^\vee+k}P'_{n_{j+1}^\wedge-1}), \quad k = 1,\dots,h_j^\vee - 1.$$

If (2.10) holds, we may insert $P_{n_j^\vee}$ and $P'_{n_{j+1}^\wedge-1}$ into (5.32b) and (5.32d), according to (5.29). We may also use (5.28b), although, in contrast to the situation in §2, there is no need to do that:

$$\text{(5.32e)}\quad \Phi(e_j^\wedge P_{n_j^\wedge+k}P_{n_j^\wedge}) = \Phi(P_{n_j^\wedge+k}(zP_{n_j^\vee-1} - \alpha_{h_j^\wedge-2}^W P_{n_j^\vee-1} - \beta_{h_j^\wedge-2}^W P_{n_j^\vee-2})), \quad k = 1,\dots,h_j^\wedge - 1,$$

$$\text{(5.32f)}\quad \Phi(e_j^\vee P'_{n_j^\vee+k}P_{n_j^\vee+1}) = \Phi(P'_{n_j^\vee+k}(zP_{n_{j+1}^\wedge-1} - \alpha_{h_j^\vee-2}^W P_{n_{j+1}^\wedge-1} - \beta_{h_j^\vee-2}^W P_{n_{j+1}^\wedge-2})), \quad k = 1,\dots,h_j^\vee - 1.$$

THEOREM 5.4. *The linear systems* (5.24) *and* (5.17) *for computing the polynomials* $e_j^\wedge$, $e_j^\vee$ *and the constants* $\varphi_j^\wedge$, $\varphi_j^\vee$ *can be replaced by the equivalent system* (5.32a)–(5.32d) *consisting of single equations for* $\varphi_j^\wedge$ *and* $\varphi_j^\vee$, *and of a right lower triangular systems for the coefficients of* $e_j^\wedge$ *and* $e_j^\wedge$. *If* (2.10) *holds, we may replace* (5.32b) *by* (5.32e), *and* (5.32d) *by* (5.32f).

6. Matrix interpretations of staircase recurrences. In this section we give a matrix formulation of Theorem 5.3 on the mixed three-term recurrences for block staircase sequences of orthogonal polynomials. It is analogous to the matrix formulation in Theorem 3.1 for the diagonal three-term recurrence (Theorem 2.7 and Corollary 2.8). Whereas a block tridiagonal matrix $\mathbf{H}$ emerged there, we first find here two block diagonal matrices, $\mathbf{E}^\vee$ and $\mathbf{E}^\wedge$, and a lower and an upper block bidiagonal matrix, $\mathbf{F}^\vee$ and $\mathbf{F}^\wedge$, respectively. They give rise to two further block bidiagonal matrices $\mathbf{G}^\vee := \mathbf{F}^\vee(\mathbf{E}^\vee)^{-1}$ and $\mathbf{G}^\wedge := \mathbf{F}^\wedge(\mathbf{E}^\wedge)^{-1}$, which turn out to be block LU factors of $\mathbf{H}$, but also block UL factors of another matrix $\mathbf{H}'$ of the same structure, which belong to the functional Φ' instead of to Φ.

Again, we assume in view of the later application to the BCG method that the polynomial basis $\{W_m\}$ satisfies the three-term recurrence (2.10) whose matrix formulation is given by (3.1)–(3.3). Then Theorem 5.3, the definitions (5.27), and the relations (5.28) and (5.30) easily yield the following theorem.

THEOREM 6.1. *Let $\{P_n\}$ and $\{P'_n\}$ be the sequences of* FOP1s *corresponding to the functionals Φ and Φ', respectively, uniquely specified by* (5.27), *where $\{W_m\}$ is a sequence satisfying the three-term recurrence* (2.10) (*possibly with $\alpha_m^W = \beta_m^W = 0$, $\forall m \in \mathbb{N}$*). *Define the infinite row vectors*

$$\mathbf{p} := [P_0, P_1, \ldots], \quad \mathbf{p}' := [P'_0, P'_1, \ldots], \tag{6.1}$$

and, for finite or infinite value of $J^\wedge$, the infinite matrices

$$\mathbf{E}^\vee := \text{block diag}\,[\mathbf{E}_0^\vee, \mathbf{E}_1^\vee, \ldots, \mathbf{E}_{J^\wedge}^\vee], \quad \mathbf{E}^\wedge := \text{block diag}\,[\mathbf{E}_0^\wedge, \mathbf{E}_1^\wedge, \ldots, \mathbf{E}_{J^\wedge}^\wedge], \tag{6.2}$$

with square blocks $\mathbf{E}_j^\vee$ and $\mathbf{E}_j^\wedge$ of order $h_j^\wedge + h_j^\vee - 1$ that for $j = 0, \ldots, J^\wedge - 1$ are given by

$$\mathbf{E}_j^\vee := \left[\begin{array}{c|c|c} \mathbf{I}_j^\wedge & \mathbf{O} & \mathbf{0} \\ \hline \mathbf{O} & \mathbf{I}_j^\vee & -\mathbf{e}_j^\vee \\ \hline \mathbf{0}^{\mathbf{T}} & \mathbf{0}^{\mathbf{T}} & 1 \end{array}\right], \quad \mathbf{E}_j^\wedge := \left[\begin{array}{c|c|c} \mathbf{I}_j^\wedge & -\mathbf{e}_j^\wedge & \mathbf{O} \\ \hline \mathbf{0}^{\mathbf{T}} & 1 & \mathbf{0}^{\mathbf{T}} \\ \hline \mathbf{O} & \mathbf{0} & \mathbf{I}_j^\vee \end{array}\right]. \tag{6.3}$$

Here, $\mathbf{I}_j^\wedge$ and $\mathbf{I}_j^\vee$ are the unit matrices of order $h_j^\wedge - 1$ and $h_j^\vee - 1$, respectively, and the row vectors

$$\mathbf{e}_j^\vee := [\varepsilon_{0,j}^\vee, \varepsilon_{1,j}^\vee, \ldots, \varepsilon_{h_j^\vee - 2,j}^\vee]^T, \quad \mathbf{e}_j^\wedge := [\varepsilon_{0,j}^\wedge, \varepsilon_{1,j}^\wedge, \ldots, \varepsilon_{h_j^\wedge - 2,j}^\wedge]^T \tag{6.4}$$

contain the coefficients of the polynomials

$$e_j^\vee(z) = \sum_{i=0}^{h_j^\vee - 2} \varepsilon_{i,j}^\vee W_i(z), \quad e_j^\wedge(z) = \sum_{i=0}^{h_j^\wedge - 2} \varepsilon_{i,j}^\wedge W_i(z) \tag{6.5}$$

from Theorem 5.3, *expressed in terms of the basis $\{W_m\}$. If $h_j^\wedge = 1$ or $h_j^\vee = 1$, the rows and columns containing $\mathbf{I}_j^\wedge$ and $\mathbf{I}_j^\vee$, respectively, are missing in $\mathbf{E}_j^\vee$ and $\mathbf{E}_j^\wedge$. If $J^\wedge < \infty$ (and thus $J^\vee < \infty$ also), then $\mathbf{E}_{J^\wedge}^\vee := \mathbf{I}$ (the infinite unit matrix); if $J^\vee = J^\wedge - 1$, $\mathbf{E}_{J^\wedge}^\wedge := \mathbf{I}$ also, while, if $J^\vee = J^\wedge$, then $\mathbf{E}_{J^\wedge}^\wedge$ has the same structure as in* (6.3), *with $\mathbf{I}_{J^\wedge}^\vee := \mathbf{I}$.*

Also define the infinite block bidiagonal matrices

$$\mathbf{F}^\vee := \begin{bmatrix} \mathbf{F}_0^\vee & & & & \\ \mathbf{L}_0^\vee & \mathbf{F}_1^\vee & & & \\ & \mathbf{L}_1^\vee & \mathbf{F}_2^\vee & & \\ & & \ddots & \ddots & \\ & & & \mathbf{L}_{J^\wedge-1}^\vee & \mathbf{F}_{J^\wedge}^\vee \end{bmatrix}, \quad \mathbf{F}^\wedge := \begin{bmatrix} \mathbf{L}_0^\wedge & \mathbf{F}_1^\wedge & & & \\ & \mathbf{L}_1^\wedge & \mathbf{F}_2^\wedge & & \\ & & \mathbf{L}_2^\wedge & \ddots & \\ & & & \ddots & \mathbf{F}_{J^\wedge}^\wedge \\ & & & & \mathbf{L}_{J^\wedge}^\wedge \end{bmatrix}, \tag{6.6}$$

with square blocks $\mathbf{F}_j^\vee$ and $\mathbf{L}_j^\wedge$ of order $h_j^\wedge + h_j^\vee - 1$, whose upper left corner is the $(n_j^\wedge, n_j^\wedge)$-entry[3] *of $\mathbf{F}^\vee$ and $\mathbf{F}^\wedge$, respectively, and which for $j = 0, 1, \ldots, J^\wedge - 1$ are*

[3] The entries in the upper left corner of $\mathbf{F}^\vee$ and $\mathbf{F}^\wedge$ are considered as (0,0)-entries. The entries in the subblocks are in this text identified by their indices in the full matrix, e.g., the $(n_j^\wedge, n_j^\wedge)$ entry of $\mathbf{F}_j^\vee$ is the same as the $(n_j^\wedge, n_j^\wedge)$ entry of $\mathbf{F}^\vee$. Likewise, the $n_j^\wedge$th row (column) of $\mathbf{F}_j^\vee$ is the $n_j^\wedge$th row (column) of $\mathbf{F}^\vee$.

given as follows: If $h_j^\wedge > 1$ *and* $h_j^\vee > 1$,

$$\text{(6.7a)}\qquad \mathbf{F}_j^\vee := \left[\begin{array}{c|c|c} \mathbf{T}_j^\wedge & \mathbf{O} & \mathbf{f}_j^\vee \\ \hline \mathbf{l}_j^{\wedge T} & \mathbf{0}^T & 0 \\ \hline \mathbf{O} & \mathbf{I}_j^\vee & \mathbf{0} \end{array}\right], \quad \mathbf{L}_j^\wedge := \left[\begin{array}{c|c|c} \mathbf{I}_j^\wedge & \mathbf{0} & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{c}_j^\vee & \mathbf{T}_j^\vee \\ \hline \mathbf{0}^T & 0 & \mathbf{l}_j^{\vee T} \end{array}\right];$$

whereas, if $h_j^\wedge = 1$ *and* $h_j^\vee > 1$,

$$\text{(6.7b)}\qquad \mathbf{F}_j^\vee := \left[\begin{array}{c|c} \mathbf{0}^T & \varphi_j^\vee \\ \hline \mathbf{I}_j^\vee & \mathbf{0} \end{array}\right], \quad \mathbf{L}_j^\wedge := \left[\begin{array}{c|c} \mathbf{c}_j^\vee & \mathbf{T}_j^\vee \\ \hline 0 & \mathbf{l}_j^{\vee T} \end{array}\right];$$

and, if $h_j^\wedge > 1$ *and* $h_j^\vee = 1$,

$$\text{(6.7c)}\qquad \mathbf{F}_j^\vee := \left[\begin{array}{c|c} \mathbf{T}_j^\wedge & \mathbf{f}_j^\vee \\ \hline \mathbf{l}_j^{\wedge T} & 0 \end{array}\right], \quad \mathbf{L}_j^\wedge := \left[\begin{array}{c|c} \mathbf{I}_j^\wedge & \mathbf{0} \\ \hline \mathbf{0}^T & 1 \end{array}\right],$$

which reduces even further to

$$\text{(6.7d)}\qquad \mathbf{F}_j^\vee := [\varphi_j^\vee], \quad \mathbf{L}_j^\wedge := [1],$$

if both $h_j^\wedge = 1$ *and* $h_j^\vee = 1$. *If* $\mathbf{T}_m^W$ *denotes the tridiagonal matrix of order* $m+1$ *with entries* β_k^W, α_k^W *and* 1 *in its* $(k+1)$*th column, cf.* (3.2), *then the blocks* $\mathbf{F}_j^\vee$, $\mathbf{F}_j^\wedge$, $\mathbf{L}_j^\vee$, *and* $\mathbf{L}_j^\wedge$ *contain*

$$\text{(6.8a)}\qquad \mathbf{T}_j^\wedge := \mathbf{T}_{h_j^\wedge - 2}^W, \quad \mathbf{T}_j^\vee := \mathbf{T}_{h_j^\vee - 2}^W,$$

$$\text{(6.8b)}\qquad \mathbf{f}_j^\wedge := [\varphi_j^\wedge, 0, \ldots, 0]^T \in \mathbf{C}^{h_j^\vee - 1}, \quad \mathbf{f}_j^\vee := [\varphi_j^\vee, 0, \ldots, 0]^T \in \mathbf{C}^{h_j^\wedge - 1},$$

$$\text{(6.8c)}\qquad \mathbf{c}_j^\wedge := [1, 0, \ldots, 0]^T \in \mathbf{C}^{h_j^\wedge - 1}, \quad \mathbf{c}_j^\vee := [1, 0, \ldots, 0]^T \in \mathbf{C}^{h_j^\vee - 1},$$

$$\text{(6.8d)}\qquad \mathbf{l}_j^{\wedge T} := [0, \ldots, 0, 1] \in \mathbf{C}^{h_j^\wedge - 1}, \quad \mathbf{l}_j^{\vee T} := [0, \ldots, 0, 1] \in \mathbf{C}^{h_j^\vee - 1}.$$

If $J^\vee = J^\wedge - 1 < \infty$,

$$\text{(6.8e)}\qquad \mathbf{F}_{J^\wedge}^\vee := \mathbf{T}^W, \quad \mathbf{L}_{J^\wedge}^\wedge := \mathbf{I},$$

while, if $J^\vee = J^\wedge < \infty$,

$$\text{(6.8f)}\qquad \mathbf{F}_{J^\wedge}^\vee := \left[\begin{array}{c|c} \mathbf{T}_{J^\wedge}^\wedge & \mathbf{O} \\ \mathbf{l}_{J^\wedge}^{\wedge T} & \mathbf{0}^T \\ \mathbf{O} & \mathbf{I} \end{array}\right], \quad \mathbf{L}_{J^\wedge}^\wedge := \left[\begin{array}{c|c|c} \mathbf{I}_{J^\wedge}^\wedge & \mathbf{0} & \mathbf{O} \\ \mathbf{O} & \mathbf{c}_{J^\wedge}^\vee & \mathbf{T}^W \end{array}\right],$$

where $\mathbf{c}_{J^\wedge}^\vee = [1, 0, 0, \ldots]^T$ *has infinitely many components.*

The off-diagonal blocks $\mathbf{F}_j^\wedge$ *and* $\mathbf{L}_j^\vee$ *are rank-one matrices of size* $(h_{j-1}^\wedge + h_{j-1}^\vee - 1) \times (h_j^\wedge + h_j^\vee - 1)$ *and* $(h_{j+1}^\wedge + h_{j+1}^\vee - 1) \times (h_j^\wedge + h_j^\vee - 1)$, *respectively. Each has a single nonzero element, namely, the* $(n_{j-1}^\vee, n_j^\vee)$ *entry* $\varphi_j^\wedge$ *of* $\mathbf{F}^\wedge$ *and the* $(n_{j+1}^\wedge, n_{j+1}^\wedge - 1)$ *entry* 1 *of* $\mathbf{F}^\vee$, *which lies in the upper right corner of* $\mathbf{L}_j^\vee$. *If* $j < J^\wedge$ *and* $h_{j-1}^\wedge, h_{j-1}^\vee, h_j^\wedge, h_j^\vee$, $h_{j+1}^\wedge$, $h_{j+1}^\vee > 1$, *then* $\mathbf{F}_j^\wedge$ *and* $\mathbf{L}_j^\vee$ *have the structure*

$$\text{(6.9)}\qquad \mathbf{F}_j^\wedge := \left[\begin{array}{c|c|c} \mathbf{O} & \mathbf{0} & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{f}_j^\wedge & \mathbf{O} \\ \hline \mathbf{0}^T & 0 & \mathbf{0}^T \end{array}\right], \quad \mathbf{L}_j^\vee := \left[\begin{array}{c|c|c} \mathbf{O} & \mathbf{O} & \mathbf{c}_j^\wedge \\ \hline \mathbf{0}^T & \mathbf{0}^T & 0 \\ \hline \mathbf{O} & \mathbf{O} & \mathbf{0} \end{array}\right],$$

with diagonal blocks of order $(h^{\wedge}_{j-1}-1)\times(h^{\wedge}_{j}-1)$, $(h^{\vee}_{j-1}-1)\times 1$, $1\times(h^{\vee}_{j}-1)$, *and* $(h^{\wedge}_{j+1}-1)\times(h^{\wedge}_{j}-1)$, $1\times(h^{\vee}_{j}-1)$, $(h^{\vee}_{j+1}-1)\times 1$, *respectively. If one or several of these sizes are zero, the structure is again modified, but the nonzero entry of* $\mathbf{L}^{\vee}_{j}$ *is always in the upper right corner; the one of* $\mathbf{F}^{\wedge}_{j}$ *is in the first column if* $h^{\wedge}_{j}=1$, *in the last column if* $h^{\vee}_{j}=1$, *in the first row if* $h^{\wedge}_{j-1}$, *and in the last row if* $h^{\vee}_{j-1}=1$. *If* $J^{\vee}=J^{\wedge}-1<\infty$, $\mathbf{F}^{\wedge}_{J^{\wedge}}=\mathbf{O}$; *and if* $J^{\vee}=J^{\wedge}<\infty$, $\mathbf{F}^{\wedge}_{J^{\wedge}}$ *has the same structure as in* (6.9), *but the last block column is infinitely wide.*

In terms of the above-defined quantities, the mixed recurrences (5.25) *for the sequences* $\{P_{n^{\wedge}_{j}}\}$ *and* $\{P'_{n^{\vee}_{j}}\}$ *can be written as*

$$\mathbf{p}(z)\mathbf{E}^{\wedge}=\mathbf{p}'(z)\mathbf{F}^{\wedge}, \tag{6.10a}$$

$$z\mathbf{p}'(z)\mathbf{E}^{\vee}=\mathbf{p}(z)\mathbf{F}^{\vee}. \tag{6.10b}$$

Likewise, if $\mathbf{E}^{\vee}_{[n]},\mathbf{E}^{\wedge}_{[n]},\mathbf{F}^{\vee}_{[n]},\mathbf{F}^{\wedge}_{[n]}$ *denote the principal submatrices of order* $n+1$ *of* $\mathbf{E}^{\vee},\mathbf{E}^{\wedge},\mathbf{F}^{\vee}$, *and* $\mathbf{F}^{\wedge}$, *respectively, and if*

$$\mathbf{p}_n:=[P_0,P_1,\dots,P_n],\quad \mathbf{p}'_n:=[P'_0,P'_1,\dots,P'_n], \tag{6.11}$$

then we have

$$\mathbf{p}_n(z)\mathbf{E}^{\wedge}_{[n]}=\mathbf{p}'_n(z)\mathbf{F}^{\wedge}_{[n]}, \tag{6.12a}$$

$$z\mathbf{p}'_n(z)\mathbf{E}^{\vee}_{[n]}=\mathbf{p}_n(z)\mathbf{F}^{\vee}_{[n]}+[0,\dots,0,P_{n+1}(z)]\,. \tag{6.12b}$$

Remarks. (i) $\mathbf{E}^{\vee}$ and $\mathbf{E}^{\wedge}$ are unit upper triangular and block diagonal, and the same holds for their inverses, which have the blocks

$$(\mathbf{E}^{\vee}_{j})^{-1}=\left[\begin{array}{c|c|c}\mathbf{I}^{\wedge}_{j} & \mathbf{O} & \mathbf{0}\\ \hline \mathbf{O} & \mathbf{I}^{\vee}_{j} & \mathbf{e}^{\vee}_{j}\\ \hline \mathbf{0}^{T} & \mathbf{0}^{T} & 1\end{array}\right],\quad (\mathbf{E}^{\wedge}_{j})^{-1}=\left[\begin{array}{c|c|c}\mathbf{I}^{\wedge}_{j} & \mathbf{e}^{\wedge}_{j} & \mathbf{O}\\ \hline \mathbf{0}^{T} & 1 & \mathbf{0}^{T}\\ \hline \mathbf{O} & \mathbf{0} & \mathbf{I}^{\vee}_{j}\end{array}\right], \tag{6.13}$$

differing from $\mathbf{E}^{\vee}_{j}$ and $\mathbf{E}^{\wedge}_{j}$ by the missing minus sign in front of $\mathbf{e}^{\vee}_{j}$ and $\mathbf{e}^{\wedge}_{j}$ only.

(ii) $\mathbf{F}^{\vee}$ is unit upper Hessenberg, and $\mathbf{F}^{\wedge}$ is unit upper triangular.

(iii) If $h^{\wedge}_{j}=h^{\vee}_{j}=1$ $(\forall j)$, $\mathbf{E}^{\vee}$ and $\mathbf{E}^{\wedge}$ are the infinite unit matrix, and $\mathbf{F}^{\vee}$ and $\mathbf{F}^{\wedge}$ are lower and upper bidiagonal,

$$\mathbf{F}^{\vee}=\begin{bmatrix}\varphi^{\vee}_{0} & & & \\ 1 & \varphi^{\vee}_{1} & & \\ & 1 & \varphi^{\vee}_{2} & \\ & & \ddots & \ddots\end{bmatrix},\quad \mathbf{F}^{\wedge}=\begin{bmatrix}1 & \varphi^{\wedge}_{1} & & \\ & 1 & \varphi^{\wedge}_{2} & \\ & & 1 & \ddots\\ & & & \ddots\end{bmatrix}. \tag{6.14}$$

By using (6.13), we can readily turn the relations (6.10) into formally explicit formulas for $\mathbf{p}$ and $\mathbf{p}'$, which, however, are used in an implicit way as recurrences for the elements of $\mathbf{p}'$ and $\mathbf{p}$, respectively, appearing on the right-hand side.

COROLLARY 6.2. *Under the assumptions of Theorem* 6.1, *there holds*

$$\mathbf{p}(z)=\mathbf{p}'(z)\,\mathbf{G}^{\wedge} \tag{6.15a}$$

and

$$z\mathbf{p}'(z)=\mathbf{p}(z)\,\mathbf{G}^{\vee}, \tag{6.15b}$$

where

$$\mathbf{G}^{\vee} = \mathbf{F}^{\vee}(\mathbf{E}^{\vee})^{-1} := \begin{bmatrix} \mathbf{G}_0^{\vee} & & & & \\ \mathbf{L}_0^{\vee} & \mathbf{G}_1^{\vee} & & & \\ & \mathbf{L}_1^{\vee} & \mathbf{G}_2^{\vee} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{L}_{J^{\wedge}-1}^{\vee} & \mathbf{G}_{J^{\wedge}}^{\vee} \end{bmatrix} \tag{6.16a}$$

and

$$\mathbf{G}^{\wedge} = \mathbf{F}^{\wedge}(\mathbf{E}^{\wedge})^{-1} := \begin{bmatrix} \mathbf{G}_0^{\wedge} & \mathbf{F}_1^{\wedge} & & & \\ & \mathbf{G}_1^{\wedge} & \mathbf{F}_2^{\wedge} & & \\ & & \mathbf{G}_2^{\wedge} & \ddots & \\ & & & \ddots & \mathbf{F}_{J^{\wedge}}^{\wedge} \\ & & & & \mathbf{G}_{J^{\wedge}}^{\wedge} \end{bmatrix} \tag{6.16b}$$

have the same off-diagonal blocks as $\mathbf{F}^{\vee}$ *and* $\mathbf{F}^{\wedge}$. *The diagonal blocks* $\mathbf{G}_j^{\vee}$ *and* $\mathbf{G}_j^{\wedge}$, $j = 0, 1, \ldots, J^{\wedge} - 1$, *are defined as follows: If* $h_j^{\wedge} > 1$ *and* $h_j^{\vee} > 1$,

$$\mathbf{G}_j^{\vee} := \left[\begin{array}{c|c|c} \mathbf{T}_j^{\wedge} & \mathbf{O} & \mathbf{f}_j^{\vee} \\ \hline \mathbf{l}_j^{\wedge T} & \mathbf{0}^T & 0 \\ \hline \mathbf{O} & \mathbf{I}_j^{\vee} & \mathbf{e}_j^{\vee} \end{array}\right], \quad \mathbf{G}_j^{\wedge} := \left[\begin{array}{c|c|c} \mathbf{I}_j^{\wedge} & \mathbf{e}_j^{\wedge} & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{c}_j^{\vee} & \mathbf{T}_j^{\vee} \\ \hline \mathbf{0}^T & 0 & \mathbf{l}_j^{\vee T} \end{array}\right]; \tag{6.17a}$$

whereas, if $h_j^{\wedge} = 1$ *and* $h_j^{\vee} > 1$,

$$\mathbf{G}_j^{\vee} := \left[\begin{array}{c|c} \mathbf{0}^T & \varphi_j^{\vee} \\ \hline \mathbf{I}_j^{\vee} & \mathbf{e}_j^{\vee} \end{array}\right], \quad \mathbf{G}_j^{\wedge} := \left[\begin{array}{c|c} \mathbf{c}_j^{\vee} & \mathbf{T}_j^{\vee} \\ \hline 0 & \mathbf{l}_j^{\vee T} \end{array}\right] = \mathbf{L}_j^{\wedge}; \tag{6.17b}$$

if $h_j^{\wedge} > 1$ *and* $h_j^{\vee} = 1$,

$$\mathbf{G}_j^{\vee} := \left[\begin{array}{c|c} \mathbf{T}_j^{\wedge} & \mathbf{f}_j^{\vee} \\ \hline \mathbf{l}_j^{\wedge T} & 0 \end{array}\right] = \mathbf{F}_j^{\vee}, \quad \mathbf{G}_j^{\wedge} := \left[\begin{array}{c|c} \mathbf{I}_j^{\wedge} & \mathbf{e}_j^{\wedge} \\ \hline \mathbf{0}^T & 1 \end{array}\right]; \tag{6.17c}$$

and, if both $h_j^{\wedge} = 1$ *and* $h_j^{\vee} = 1$,

$$\mathbf{G}_j^{\vee} := [\varphi_j^{\vee}] = \mathbf{F}_j^{\vee}, \quad \mathbf{G}_j^{\wedge} := [1] = \mathbf{L}_j^{\wedge}. \tag{6.17d}$$

If $J^{\vee} = J^{\wedge} - 1 < \infty$,

$$\mathbf{G}_{J^{\wedge}}^{\vee} := \mathbf{F}_{J^{\wedge}}^{\vee} = \mathbf{T}^W, \quad \mathbf{G}_{J^{\wedge}}^{\wedge} := \mathbf{L}_{J^{\wedge}}^{\wedge} = \mathbf{I}, \tag{6.17e}$$

while, if $J^{\vee} = J^{\wedge} < \infty$,

$$\mathbf{G}_{J^{\wedge}}^{\vee} := \mathbf{F}_{J^{\wedge}}^{\vee} := \left[\begin{array}{c|c} \mathbf{T}_{J^{\wedge}}^{\wedge} & \mathbf{O} \\ \mathbf{l}_{J^{\wedge}}^{\wedge T} & 0^T \\ \mathbf{O} & \mathbf{I} \end{array}\right], \quad \mathbf{G}_{J^{\wedge}}^{\wedge} := \left[\begin{array}{c|c|c} \mathbf{I}_{J^{\wedge}}^{\wedge} & \mathbf{e}_{J^{\wedge}}^{\wedge} & \mathbf{O} \\ \mathbf{O} & \mathbf{c}_{J^{\wedge}}^{\vee} & \mathbf{T}^W \end{array}\right]. \tag{6.17f}$$

Note that the two relations (6.15) are just the recurrences (5.30) and (5.31) expressed in matrix notation.

TABLE 1
Computation of $\mathbf{H}$ *from* $\mathbf{G}^\vee$ *and* $\mathbf{G}^\wedge$ *by equating the elements of* $\mathbf{H}$ *with those of* $\mathbf{G}^\vee\mathbf{G}^\wedge$.

$h^\vee_{j-1}$	$h^\wedge_j$	$h^\vee_j$	n_{i-1}	n_i	n_{i+1}	n_{i+2}
>1	>1	>1	$n^\vee_{j-1}+1$	$n^\wedge_j$	$n^\vee_j+1$	$n^\wedge_{j+1}$
$=1$	>1	>1	$n^\wedge_{j-1}$	$n^\wedge_j=n^\vee_{j-1}+1$	$n^\vee_j+1$	$n^\wedge_{j+1}$
>1	$=1$	>1	$n^\vee_{j-1}+1$	$n^\wedge_j=n^\vee_j$	$n^\vee_j+1$	$n^\wedge_{j+1}$
$=1$	$=1$	>1	$n^\wedge_{j-1}$	$n^\wedge_j=n^\vee_{j-1}+1=n^\vee_j$	$n^\vee_j+1$	$n^\wedge_{j+1}$
>1	>1	$=1$	$n^\vee_{j-1}+1$	$n^\wedge_j$	$n^\wedge_{j+1}=n^\vee_j+1$	$n^\vee_{j+1}+1$
$=1$	>1	$=1$	$n^\wedge_{j-1}$	$n^\wedge_j=n^\vee_{j-1}+1$	$n^\wedge_{j+1}=n^\vee_j+1$	$n^\vee_{j+1}+1$
>1	$=1$	$=1$	$n^\vee_{j-1}+1$	$n^\wedge_j=n^\vee_j$	$n^\wedge_{j+1}=n^\vee_j+1$	$n^\vee_{j+1}+1$
$=1$	$=1$	$=1$	$n^\wedge_{j-1}$	$n^\wedge_j=n^\vee_{j-1}+1=n^\vee_j$	$n^\wedge_{j+1}=n^\vee_j+1$	$n^\vee_{j+1}+1$

$h^\vee_{j-1}$	$h^\wedge_j$	$h^\vee_j$	h_i	h_{i+1}	$\mathbf{a}_i$	β_i	$\mathbf{a}_{i+1}$	β_{i+1}
>1	>1	>1	$h^\wedge_j$	$h^\vee_j-1$	$\Omega_j(\mathbf{e}^\wedge_j,0)$	$\varphi^\wedge_j$	$\mathbf{e}^\vee_j$	$\varphi^\vee_j$
$=1$	>1	>1	$h^\wedge_j$	$h^\vee_j-1$	$\Omega_j(\mathbf{e}^\wedge_j,\varphi^\wedge_j)$	$\varphi^\wedge_{j-1}\varphi^\wedge_j$	$\mathbf{e}^\vee_j$	$\varphi^\vee_j$
>1	$=1$	>1	$h^\wedge_j$	$h^\vee_j-1$	$[0]$	$\varphi^\wedge_j$	$\mathbf{e}^\vee_j$	$\varphi^\vee_j$
$=1$	$=1$	>1	$h^\wedge_j$	$h^\vee_j-1$	$[\varphi^\wedge_j]$	$\varphi^\wedge_{j-1}\varphi^\wedge_j$	$\mathbf{e}^\vee_j$	$\varphi^\vee_j$
>1	>1	$=1$	$h^\wedge_j$	$h^\wedge_{j+1}$	$\Omega_j(\mathbf{e}^\wedge_j,\varphi^\vee_j)$	$\varphi^\wedge_j$	–	–
$=1$	>1	$=1$	$h^\wedge_j$	$h^\wedge_{j+1}$	$\Omega_j(\mathbf{e}^\wedge_j,\varphi^\vee_j+\varphi^\wedge_j)$	$\varphi^\vee_{j-1}\varphi^\wedge_j$	–	–
>1	$=1$	$=1$	$h^\wedge_j$	$h^\wedge_{j+1}$	$[\varphi^\vee_j]$	$\varphi^\wedge_j$	–	–
$=1$	$=1$	$=1$	$h^\wedge_j$	$h^\wedge_{j+1}$	$[\varphi^\vee_j+\varphi^\wedge_j]$	$\varphi^\vee_{j-1}\varphi^\wedge_j$	–	–

7. Matrix relations between diagonal and staircase recurrences: The nongeneric qd algorithm. From the two relations (6.15) we can eliminate either $\mathbf{p}$ or $\mathbf{p}'$ to obtain

$$z\mathbf{p}(z) = \mathbf{p}(z)\mathbf{G}^\vee\mathbf{G}^\wedge \tag{7.1a}$$

and

$$z\mathbf{p}'(z) = \mathbf{p}'(z)\mathbf{G}^\wedge\mathbf{G}^\vee. \tag{7.1b}$$

However, these are—in matrix notation—just the recurrences for the FOP1s P_n and those for the FOP1s P'_n. Therefore, they must be identical to (3.11), which describes this recurrence for P_n, and with the corresponding relation for the set $\{P'_n\}$, respectively. This leads to the following result.

THEOREM 7.1. *Under the assumptions of Theorem* 6.1 *let* $\mathbf{H}$ *and* $\mathbf{H}'$ *be the block tridiagonal matrices from* (3.6) *that describe, according to* (3.11), *the recurrences for the* FOP1s $\{P_n\}$ *and* $\{P'_n\}$, *respectively, that correspond to the linear functionals* Φ *and* Φ'. *Then* $\mathbf{H}$ *has the block* LU *factorization*

$$\mathbf{H} = \mathbf{G}^\vee\mathbf{G}^\wedge, \tag{7.2a}$$

and $\mathbf{H}'$ *has the block* UL *factorization*

$$\mathbf{H}' = \mathbf{G}^\wedge\mathbf{G}^\vee, \tag{7.2b}$$

where $\mathbf{G}^\vee$ *and* $\mathbf{G}^\wedge$ *are the block bidiagonal matrices defined by* (6.16)–(6.17) *and* (6.9). *The resulting relation between the entries of* $\mathbf{G}^\vee$, $\mathbf{G}^\wedge$, *and* $\mathbf{H}$ *or* $\mathbf{H}'$ *are listed in Tables*

TABLE 2
Computation of $\mathbf{H}'$ *from* $\mathbf{G}^\vee$ *and* $\mathbf{G}^\wedge$ *by equating the elements of* $\mathbf{H}'$ *with those of* $\mathbf{G}^\wedge\mathbf{G}^\vee$.

$h_j^\wedge$	$h_j^\vee$	$h_{j+1}^\wedge$	n'_{i-1}	n'_i	n'_{i+1}	n'_{i+2}
>1	>1	>1	$n_j^\wedge$	$n_j^\vee$	$n_{j+1}^\wedge$	$n_{j+1}^\vee$
$=1$	>1	>1	$n_{j-1}^\vee$	$n_j^\vee = n_j^\wedge$	$n_{j+1}^\wedge$	$n_{j+1}^\vee$
>1	$=1$	>1	$n_j^\wedge$	$n_j^\vee$	$n_{j+1}^\wedge = n_j^\vee + 1$	$n_{j+1}^\vee$
$=1$	$=1$	>1	$n_{j-1}^\vee$	$n_j^\vee = n_j^\wedge$	$n_{j+1}^\wedge = n_j^\vee + 1$	$n_{j+1}^\vee$
>1	>1	$=1$	$n_j^\wedge$	$n_j^\vee$	$n_{j+1}^\vee = n_{j+1}^\wedge$	$n_{j+2}^\wedge$
$=1$	>1	$=1$	$n_{j-1}^\vee$	$n_j^\wedge = n_j^\vee$	$n_{j+1}^\vee = n_{j+1}^\wedge$	$n_{j+2}^\wedge$
>1	$=1$	$=1$	$n_j^\wedge$	$n_j^\vee$	$n_{j+1}^\vee = n_{j+1}^\wedge = n_j^\vee + 1$	$n_{j+2}^\wedge$
$=1$	$=1$	$=1$	$n_{j-1}^\vee$	$n_j^\wedge = n_j^\vee$	$n_{j+1}^\vee = n_{j+1}^\wedge = n_j^\vee + 1$	$n_{j+2}^\wedge$

$h_j^\wedge$	$h_j^\vee$	$h_{j+1}^\wedge$	h'_i	h'_{i+1}	$\mathbf{a}'_{i-1}$	β'_{i-1}	$\mathbf{a}'_i$	β'_i
>1	>1	>1	$h_j^\vee$	$h_{j+1}^\wedge - 1$	$\mathbf{e}_j^\wedge$	$\varphi_j^\wedge$	$\Omega'_j(\mathbf{e}_j^\vee, 0)$	$\varphi_j^\vee$
$=1$	>1	>1	$h_j^\vee$	$h_{j+1}^\wedge - 1$	–	–	$\Omega'_j(\mathbf{e}_j^\vee, \varphi_j^\vee)$	$\varphi_j^\vee\varphi_j^\wedge$
>1	$=1$	>1	$h_j^\vee$	$h_{j+1}^\wedge - 1$	$\mathbf{e}_j^\wedge$	$\varphi_j^\wedge$	$[0]$	$\varphi_j^\vee$
$=1$	$=1$	>1	$h_j^\vee$	$h_{j+1}^\wedge - 1$	–	–	$[\varphi_j^\vee]$	$\varphi_j^\vee\varphi_j^\wedge$
>1	>1	$=1$	$h_j^\vee$	$h_{j+1}^\vee$	$\mathbf{e}_j^\wedge$	$\varphi_j^\wedge$	$\Omega'_j(\mathbf{e}_j^\vee, \varphi_{j+1}^\wedge)$	$\varphi_j^\vee$
$=1$	>1	$=1$	$h_j^\vee$	$h_{j+1}^\vee$	–	–	$\Omega'_j(\mathbf{e}_j^\vee, \varphi_j^\vee + \varphi_{j+1}^\wedge)$	$\varphi_j^\vee\varphi_j^\wedge$
>1	$=1$	$=1$	$h_j^\vee$	$h_{j+1}^\vee$	$\mathbf{e}_j^\wedge$	$\varphi_j^\wedge$	$[\varphi_{j+1}^\wedge]$	$\varphi_j^\vee$
$=1$	$=1$	$=1$	$h_j^\vee$	$h_{j+1}^\vee$	–	–	$[\varphi_j^\vee + \varphi_{j+1}^\wedge]$	$\varphi_j^\vee\varphi_j^\wedge$

1 *and* 2. *There* n_i *and* n'_i *denote the sequences of the indices of the regular* FOP1*s out of* $\{P_n\}$ *and* $\{P'_n\}$, *respectively; the entries of* $\mathbf{H}'$ *are also distinguished by a prime from those of* $\mathbf{H}$. *The functions* Ω_j *and* Ω'_j *are defined by*

$$\Omega_j(\mathbf{e}_j^\wedge, \varphi) := \mathbf{T}^W_{h_j^\wedge - 1}\left[\begin{array}{c}\mathbf{e}_j^\wedge\\ \hline -1\end{array}\right] + \begin{bmatrix}\varphi\\ 0\\ \vdots\\ 0\end{bmatrix}, \tag{7.3a}$$

$$\Omega'_j(\mathbf{e}_j^\vee, \varphi) := \mathbf{T}^W_{h_j^\vee - 1}\left[\begin{array}{c}\mathbf{e}_j^\vee\\ \hline -1\end{array}\right] + \begin{bmatrix}\varphi\\ 0\\ \vdots\\ 0\end{bmatrix}. \tag{7.3b}$$

Remark. In the case $W_m(z) = z^m$, the definitions (7.3) reduce to

$$\Omega_j(\mathbf{e}_j^\wedge, \varphi) := \left[\frac{\varphi}{\mathbf{e}_j^\wedge}\right], \quad \Omega'_j(\mathbf{e}_j^\vee, \varphi) := \left[\frac{\varphi}{\mathbf{e}_j^\vee}\right]. \tag{7.4}$$

Proof. We have already derived (7.2), so it remains to relate the entries of $\mathbf{H}$ and $\mathbf{H}'$ to those of $\mathbf{G}^\vee$ and $\mathbf{G}^\wedge$. For this we need to compute explicitly the elements of the products $\mathbf{G}^\vee\mathbf{G}^\wedge$ and $\mathbf{G}^\wedge\mathbf{G}^\vee$. The task is complicated by the fact that the structure of the blocks depends on the quantities $h_j^\wedge$ and $h_j^\vee$ being larger than or equal to 1.

Let us start with the off-diagonal blocks of the products. First, the $(j+1, j)$-blocks of $\mathbf{G}^\vee\mathbf{G}^\wedge$ and $\mathbf{G}^\wedge\mathbf{G}^\vee$ are $\mathbf{L}_j^\vee\mathbf{G}_j^\wedge = \mathbf{L}_j^\vee$ and $\mathbf{G}_{j+1}^\wedge\mathbf{L}_j^\vee = \mathbf{L}_j^\vee$, respectively. The

$(j-1,1)$-blocks are $\mathbf{G}^{\vee}_{j-1}\mathbf{F}^{\wedge}_{j}$ and $\mathbf{F}^{\wedge}_{j}\mathbf{G}^{\vee}_{j}$, and their structure depends on the value of $h^{\vee}_{j-1}$ and $h^{\wedge}_{j}$, respectively. Since $\mathbf{F}^{\wedge}_{j}$ has only one nonzero entry, namely, the $(n^{\vee}_{j-1}, n^{\vee}_{j})$ entry $\varphi^{\wedge}_{j}$ of $\mathbf{G}^{\wedge}$, only the $n^{\vee}_{j-1}$th column of $\mathbf{G}^{\vee}_{j-1}$ and the $n^{\vee}_{j}$th row of $\mathbf{G}^{\vee}_{j}$ matter. (Columns and rows of the blocks of $\mathbf{G}^{\vee}$ and $\mathbf{G}^{\wedge}$ are here again numbered according to their indices in the whole matrix.) The $n^{\vee}_{j-1}$th column of $\mathbf{G}^{\vee}_{j-1}$ contains 1 as its $(n^{\vee}_{j-1}+1, n^{\vee}_{j-1})$ entry if $h^{\vee}_{j-1} > 1$, and $\varphi^{\vee}_{j-1}$ as its $(n^{\wedge}_{j-1}, n^{\vee}_{j-1})$ entry if $h^{\vee}_{j-1} = 1$. Consequently,

$$[\mathbf{G}^{\vee}_{j-1}\mathbf{F}^{\wedge}_{j}]_{m,n} = \left\{ \begin{array}{ll} \varphi^{\wedge}_{j} & \text{if } (m,n) = (n^{\vee}_{j-1}+1, n^{\vee}_{j}) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } h^{\vee}_{j-1} > 1, \tag{7.5a}$$

and

$$[\mathbf{G}^{\vee}_{j-1}\mathbf{F}^{\wedge}_{j}]_{m,n} = \left\{ \begin{array}{ll} \varphi^{\vee}_{j-1}\varphi^{\wedge}_{j} & \text{if } (m,n) = (n^{\wedge}_{j-1}, n^{\vee}_{j}) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } h^{\vee}_{j-1} = 1. \tag{7.5b}$$

The $n^{\vee}_{j}$th row of $\mathbf{G}^{\vee}_{j}$ contains 1 as its $(n^{\vee}_{j}, n^{\vee}_{j}-1)$ entry if $h^{\wedge}_{j} > 1$, and $\varphi^{\vee}_{j}$ as its $(n^{\wedge}_{j}, n^{\wedge}_{j+1}-1)$ entry if $h^{\wedge}_{j} = 1$, so that

$$[\mathbf{F}^{\wedge}_{j}\mathbf{G}^{\vee}_{j}]_{m,n} = \left\{ \begin{array}{ll} \varphi^{\wedge}_{j} & \text{if } (m,n) = (n^{\vee}_{j-1}, n^{\vee}_{j}-1) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } h^{\wedge}_{j} > 1, \tag{7.6a}$$

and

$$[\mathbf{F}^{\wedge}_{j}\mathbf{G}^{\vee}_{j}]_{m,n} = \left\{ \begin{array}{ll} \varphi^{\vee}_{j}\varphi^{\wedge}_{j} & \text{if } (m,n) = (n^{\vee}_{j-1}, n^{\wedge}_{j+1}-1) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } h^{\wedge}_{j} = 1. \tag{7.6b}$$

The diagonal blocks of $\mathbf{G}^{\vee}\mathbf{G}^{\wedge}$ and $\mathbf{G}^{\wedge}\mathbf{G}^{\vee}$ are $\mathbf{G}^{\vee}_{j}\mathbf{G}^{\wedge}_{j} + \mathbf{L}^{\vee}_{j-1}\mathbf{F}^{\wedge}_{j}$ and $\mathbf{G}^{\wedge}_{j}\mathbf{G}^{\vee}_{j} + \mathbf{F}^{\wedge}_{j+1}\mathbf{L}^{\vee}_{j}$. Here we obtain

$$\mathbf{L}^{\vee}_{j-1}\mathbf{F}^{\wedge}_{j} = \mathbf{O} \text{ if } h^{\vee}_{j-1} > 1, \tag{7.7a}$$

$$[\mathbf{L}^{\vee}_{j-1}\mathbf{F}^{\wedge}_{j}]_{m,n} = \left\{ \begin{array}{ll} \varphi^{\wedge}_{j} & \text{if } (m,n) = (n^{\wedge}_{j}, n^{\vee}_{j}) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } h^{\vee}_{j-1} = 1, \tag{7.7b}$$

$$\mathbf{F}^{\wedge}_{j+1}\mathbf{L}^{\vee}_{j} = \mathbf{O} \text{ if } h^{\wedge}_{j+1} > 1, \tag{7.8a}$$

$$[\mathbf{F}^{\wedge}_{j+1}\mathbf{L}^{\vee}_{j}]_{m,n} = \left\{ \begin{array}{ll} \varphi^{\wedge}_{j+1} & \text{if } (m,n) = (n^{\vee}_{j}, n^{\wedge}_{j+1}-1) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } h^{\wedge}_{j+1} = 1. \tag{7.8b}$$

Furthermore, if $h^{\wedge}_{j} > 1$ and $h^{\vee}_{j} > 1$,

$$\mathbf{G}^{\vee}_{j}\mathbf{G}^{\wedge}_{j} = \left[\begin{array}{c|c|c} \mathbf{T}^{\wedge}_{j} & \mathbf{T}^{\wedge}_{j}\mathbf{e}^{\wedge}_{j} & \mathbf{f}^{\vee}_{j}\mathbf{l}^{\vee T}_{j} \\ \hline \mathbf{l}^{\wedge T}_{j} & \varepsilon^{\wedge}_{h^{\wedge}_{j}-2,j} & \mathbf{0}^{T} \\ \hline \mathbf{O} & \mathbf{c}^{\vee}_{j} & \mathbf{T}^{\vee}_{j} + \mathbf{e}^{\vee}_{j}\mathbf{l}^{\vee T}_{j} \end{array} \right], \tag{7.9a}$$

$$\mathbf{G}^{\wedge}_{j}\mathbf{G}^{\vee}_{j} = \left[\begin{array}{c|c|c} \mathbf{T}^{\wedge}_{j} + \mathbf{e}^{\wedge}_{j}\mathbf{l}^{\wedge T}_{j} & \mathbf{O} & \mathbf{f}^{\vee}_{j} \\ \hline \mathbf{c}^{\vee}_{j}\mathbf{l}^{\wedge T}_{j} & \mathbf{T}^{\vee}_{j} & \mathbf{T}^{\vee}_{j}\mathbf{e}^{\vee}_{j} \\ \hline \mathbf{0}^{T} & \mathbf{l}^{\vee T}_{j} & \varepsilon^{\vee}_{h^{\vee}_{j}-2,j} \end{array} \right]; \tag{7.9b}$$

if $h_j^\wedge = 1$ and $h_j^\vee > 1$,

$$\text{(7.9b)}\quad \mathbf{G}_j^\vee \mathbf{G}_j^\wedge = \left[\begin{array}{c|c} 0 & \varphi_j^\vee \mathbf{l}_j^{\vee T} \\ \hline \mathbf{c}_j^\vee & \mathbf{T}_j^\vee + \mathbf{e}_j^\vee \mathbf{l}_j^{\vee T} \end{array}\right], \quad \mathbf{G}_j^\wedge \mathbf{G}_j^\vee = \left[\begin{array}{c|c} \mathbf{T}_j^\vee & \mathbf{c}_j^\vee \varphi_j^\vee + \mathbf{T}_j^\vee \mathbf{e}_j^\vee \\ \hline \mathbf{l}_j^{\vee T} & \varepsilon_{h_j^\vee - 2, j}^\vee \end{array}\right];$$

if $h_j^\wedge > 1$ and $h_j^\vee = 1$,

$$\text{(7.9c)}\quad \mathbf{G}_j^\vee \mathbf{G}_j^\wedge = \left[\begin{array}{c|c} \mathbf{T}_j^\wedge & \mathbf{T}_j^\wedge \mathbf{e}_j^\wedge + \mathbf{f}_j^\vee \\ \hline \mathbf{l}_j^{\wedge T} & \varepsilon_{h_j^\wedge - 2, j}^\wedge \end{array}\right], \quad \mathbf{G}_j^\wedge \mathbf{G}_j^\vee = \left[\begin{array}{c|c} \mathbf{T}_j^\wedge + \mathbf{e}_j^\wedge \mathbf{l}_j^{\wedge T} & \mathbf{f}_j^\vee \\ \hline \mathbf{l}_j^{\wedge T} & 0 \end{array}\right];$$

and, if $h_j^\wedge = h_j^\vee = 1$,

$$\text{(7.9d)}\qquad \mathbf{G}_j^\vee \mathbf{G}_j^\wedge = [\varphi_j^\vee], \quad \mathbf{G}_j^\wedge \mathbf{G}_j^\vee = [\varphi_j^\vee].$$

To relate the entries of $\mathbf{H}$ and $\mathbf{H}'$ with those of $\mathbf{G}^\vee\mathbf{G}^\wedge$ and $\mathbf{G}^\wedge\mathbf{G}^\vee$, we need to associate the blocks $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ of $\mathbf{H}$ and the blocks $\mathbf{A}'_i, \mathbf{B}'_i, \mathbf{C}'_i$ of $\mathbf{H}'$ with those of $\mathbf{G}^\vee\mathbf{G}^\wedge$ and $\mathbf{G}^\wedge\mathbf{G}^\vee$, respectively. However, depending on $h_j^\vee$ and $h_j^\wedge$ being equal to or greater than 1, a diagonal block

$$\text{(7.10a)}\qquad \mathbf{H}_j^\wedge := \mathbf{G}_j^\vee \mathbf{G}_j^\wedge + \mathbf{L}_{j-1}^\vee \mathbf{F}_j^\wedge$$

of order $h_j^\wedge + h_j^\vee - 1$ of $\mathbf{G}^\vee\mathbf{G}^\wedge = \mathbf{H}$ and such a block

$$\text{(7.10b)}\qquad \mathbf{H}_j^\vee := \mathbf{G}_j^\wedge \mathbf{G}_j^\vee + \mathbf{F}_{j+1}^\wedge \mathbf{L}_j^\vee$$

of $\mathbf{G}^\wedge\mathbf{G}^\vee = \mathbf{H}'$ corresponds either to a single block $\mathbf{A}_i$ or $\mathbf{A}'_i$ or to a 2×2 block matrix

$$\text{(7.11)}\qquad \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_{i+1} \\ \mathbf{C}_i & \mathbf{A}_{i+1} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \mathbf{A}'_{i-1} & \mathbf{B}'_i \\ \mathbf{C}'_{i-1} & \mathbf{A}'_i \end{bmatrix},$$

respectively. Likewise, the off-diagonal blocks may correspond to a $1 \times 1, 1 \times 2, 2 \times 1$, or 2×2 block matrix.

The indices of the regular elements of the sequences $\{P_n\}$ and $\{P'_n\}$ are now denoted by n_i and n'_i, respectively, while $\{n_j^\wedge\} \subseteq \{n_i\}$ and $\{n_j^\vee\} \subseteq \{n'_i\}$ still denote the index subsequences of the regular elements in the generalized staircase. Since the upper left corners of $\mathbf{A}_i$ and $\mathbf{G}_j^\vee\mathbf{G}_j^\wedge$ are at (n_i, n_i) and $(n_j^\wedge, n_j^\wedge)$, the association of the blocks is based on the identification $n_i = n_j^\wedge$. Likewise, the upper left corner of $\mathbf{A}'_i$ at (n'_i, n'_i) corresponds to the $(n_j^\vee, n_j^\vee)$ element of $\mathbf{G}_j^\wedge\mathbf{G}_j^\vee$, hence we have $n'_i = n_j^\vee$. From (7.3)–(7.9) we can then read off the associations listed in Tables 1 and 2. □

Tables 1 and 2 describe the structure and the entries of the products $\mathbf{H} = \mathbf{G}^\vee\mathbf{G}^\wedge$ and $\mathbf{H}' = \mathbf{G}^\wedge\mathbf{G}^\vee$ in terms of the entries of $\mathbf{G}^\vee$ and $\mathbf{G}^\wedge$. Next, we are interested in inverting these two operations, i.e., in computing the lower block bidiagonal matrix $\mathbf{G}^\vee$ and the upper block bidiagonal matrix $\mathbf{G}^\wedge$, either from $\mathbf{H}$ or from $\mathbf{H}'$. Of course, $\mathbf{G}^\vee$ and $\mathbf{G}^\wedge$ are required to have the structure specified by (6.16) and (6.17). While $\mathbf{H} = \mathbf{G}^\vee\mathbf{G}^\wedge$ is a block LU decomposition, $\mathbf{H}' = \mathbf{G}^\wedge\mathbf{G}^\vee$ is a (not so often encountered) block UL decomposition. Theoretically, in view of

$$\text{(7.12)}\qquad (\mathbf{H}')^{-1} = (\mathbf{G}^\vee)^{-1}(\mathbf{G}^\wedge)^{-1},$$

the latter could be obtained via a block LU decomposition of $(\mathbf{H}')^{-1}$ followed by the inversion of the factors, but we can directly determine the block UL decomposition

with ease. From Theorem 7.1 we know that these two block decompositions exist, but it is not yet clear how the sizes of the blocks of $\mathbf{G}^\vee$ and $\mathbf{G}^\wedge$ can be determined from those in $\mathbf{H}$ or from those in $\mathbf{H}'$.

In the jth step of the block LU decomposition, $\mathbf{H} = \mathbf{G}^\vee\mathbf{G}^\wedge$, the block pivot element is obtained by subtracting $\mathbf{L}_{j-1}^\vee\mathbf{F}_j^\wedge$ from a diagonal block $\mathbf{H}_j^\wedge$ of $\mathbf{H}$ of the appropriate size. It must then be split into $\mathbf{G}_j^\vee\mathbf{G}_j^\wedge$:

$$\mathbf{H}_j^\wedge - \mathbf{L}_{j-1}^\vee\mathbf{F}_j^\wedge = \mathbf{G}_j^\vee\mathbf{G}_j^\wedge. \tag{7.13}$$

However, given $\mathbf{H}$ structured according to (3.6)–(3.10), we do not know a priori the sizes of these diagonal blocks, since they may correspond to a single block $\mathbf{A}_i$ or to a 2×2 block matrix given on the left-hand side of (7.11). (This is also the reason for calling these blocks $\mathbf{H}_j^\wedge$ and not $\mathbf{H}_j$; the block sizes are the same as in $\mathbf{G}^\wedge$ and $\mathbf{G}^\vee$.) But any block pivot has to be nonsingular, and from this requirement we can determine the correct size of the block $\mathbf{H}_j^\wedge$. Any tentative 1×1 block pivot is either a 1×1 matrix obtained by updating $[\alpha_{0,i}]$ or a unit upper Hessenberg matrix. The latter is a companion matrix if we assume for the moment that $W_m(z) = z^m$. Hence it is nonsingular if and only if the element in its upper right corner is nonzero. Thus, identifying $n_i = n_j^\wedge$, we then have $\mathbf{H}_j^\wedge := \mathbf{A}_i$ if and only if $\alpha_{0,i}$ minus the $(n_i, n_{i+1}-1)$ element of $\mathbf{L}_{j-1}^\vee\mathbf{F}_j^\wedge$ does not vanish, i.e., in view of (7.7) and $n_j^\vee = n_{i+1} - 1$ (cf. Table 1), if and only if

$$\tilde{\varphi}_j^\vee := \alpha_{0,i} \neq 0 \text{ in case } h_{j-1}^\vee > 1, \tag{7.14a}$$

$$\tilde{\varphi}_j^\vee := \alpha_{0,i} - \varphi_j^\wedge \neq 0 \text{ in case } h_{j-1}^\vee = 1\,. \tag{7.14b}$$

Note that $\varphi_j^\wedge = \beta_i/\varphi_{j-1}^\wedge$ if $h_{j-1}^\vee = 1$ (cf. Table 1), and that this quantity can be computed at this moment. Moreover, in the case of a 1×1 block pivot, i.e., when (7.14) holds, we conclude from Table 1 and (7.4) that $\varphi_j^\vee = \tilde{\varphi}_j^\vee$. Hence, (7.14) means that we test whether the tentative value $\tilde{\varphi}_j^\vee$ of $\varphi_j^\vee$ does not vanish. This value does not depend on the basis $\{W_m\}$ chosen, as long as this basis consists of monic polynomials, since $\varphi_j^\vee$ is the coefficient of a regular FOP1 in one of our mixed three-term recurrence formulas (5.25). Therefore, in the general case, we can replace (7.14) by a test for the nonvanishing of the tentative value of $\varphi_j^\vee$, which can be found by inversion of the function Ω_j, cf. Table 1.

If the test fails, we end up with a 2×2 block pivot,

$$\mathbf{H}_j^\wedge := \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_{i+1} \\ \mathbf{C}_i & \mathbf{A}_{i+1} \end{bmatrix}, \tag{7.15}$$

cf. (7.11), for which the unit Hessenberg matrix $\mathbf{H}_j^\wedge - \mathbf{L}_{j-1}^\vee\mathbf{F}_j^\wedge$ is always nonsingular, since $\beta_{i+1} \neq 0$, while the $(n_i', n_{i+2}' - 1)$-element of $\mathbf{L}_{j-1}^\vee\mathbf{F}_j^\wedge$ is always zero because $n_j^\vee < n_{i+2}' - 1$.

The matrices $\mathbf{G}^\vee$ and $\mathbf{G}^\wedge$ can thus be built up by successive determination of the sizes of the block pivots and simultaneous computation of the entries by using the formulas of Table 1. The result is summarized in the following theorem and in Table 3.

THEOREM 7.2. *Given* $\mathbf{H}$, *one can compute the block* LU *factorization* $\mathbf{H} = \mathbf{G}^\vee\mathbf{G}^\wedge$ *by a Gauss block elimination process. In step* j, *where the upper left corner of the block pivot* $\mathbf{H}_j^\wedge - \mathbf{L}_{j-1}^\vee\mathbf{F}_j^\wedge$ *is at* $(n_j^\wedge, n_j^\wedge) = (n_i, n_i)$, *the block* $\mathbf{H}_j^\wedge$ *of* $\mathbf{H}$ *is defined by identifying it with either the* 1×1 *block* $\mathbf{A}_i$ *or the* 2×2 *block* (7.15) (*hence the size of the block pivot is either* h_i *or* $h_i + h_{i+1}$), *depending on whether the tentative value*

$\tilde{\varphi}_j^\vee$ of $\varphi_j^\vee$ given in Table 3 is nonzero or zero. The relevant entries $\mathbf{e}_j^\vee, \varphi_j^\vee, \mathbf{e}_j^\wedge$, and $\varphi_j^\wedge$ of $\mathbf{G}^\vee$ and $\mathbf{G}^\wedge$ are then obtained according to Table 3. There the functions $\Omega_j^\wedge$ and $\omega_j^\wedge$ are together the inverse of Ω_j defined in (7.3a)*; they are computed as follows: Partition the matrix $\mathbf{T}_{h_j^\wedge-1}^W$ and the vector $\mathbf{a}_i$ according to*

$$\mathbf{T}_{h_j^\wedge-1}^W =: \left[\begin{array}{c|c} \mathbf{s}^T & \tau \\ \hline \mathbf{S} & \mathbf{t} \end{array}\right], \quad \mathbf{a}_i =: \left[\begin{array}{c} \alpha \\ \hline \mathbf{a} \end{array}\right]; \tag{7.16}$$

then set

$$\Omega_j^\wedge(\mathbf{a}_i) := \mathbf{S}^{-1}(\mathbf{a}+\mathbf{t}), \tag{7.17a}$$

$$\omega_j^\wedge(\mathbf{a}_i) := \alpha + \tau - \mathbf{s}^T\Omega_j^\wedge(\mathbf{a}_i)\,. \tag{7.17b}$$

The process starts at $j = i = 0$ with $n_0^\wedge := n_0$, $\varphi_0^\wedge := 0$, $\mathbf{L}_{-1}^\vee\mathbf{F}_0^\wedge := \mathbf{O}$. ($h_{-1}^\vee$ does not matter.)

Remark. In the case where $W_m(z) = z^m$, the definitions (7.17) reduce to

$$\Omega_j^\wedge(\mathbf{a}_i) := \mathbf{a} = \mathbf{J}_{h_j^\wedge-1}\mathbf{a}_i, \tag{7.18a}$$

$$\omega_j^\wedge(\mathbf{a}_i) := \alpha = \alpha_{0,i}\,. \tag{7.18b}$$

TABLE 3
Formulas for the block LU *factorization* $\mathbf{H} = \mathbf{G}^\vee\mathbf{G}^\wedge$.

$h_{j-1}^\vee$	h_i	$\tilde{\varphi}_j^\vee$	$n_j^\wedge$	$n_j^\vee$	$n_{j+1}^\wedge$
>1	>1	$\omega_j^\wedge(\mathbf{a}_i) = 0$	n_i	$n_{i+1}-1$	n_{i+2}
$=1$	>1	$\omega_j^\wedge(\mathbf{a}_i) - \varphi_j^\wedge = 0$	n_i	$n_{i+1}-1$	n_{i+2}
>1	$=1$	$\alpha_{0,i} = 0$	n_i	$n_i = n_{i+1}-1$	n_{i+2}
$=1$	$=1$	$\alpha_{0,i} - \varphi_j^\wedge = 0$	n_i	$n_i = n_{i+1}-1$	n_{i+2}
>1	>1	$\omega_j^\wedge(\mathbf{a}_i) \neq 0$	n_i	$n_{i+1}-1$	n_{i+1}
$=1$	>1	$\omega_j^\wedge(\mathbf{a}_i) - \varphi_j^\wedge \neq 0$	n_i	$n_{i+1}-1$	n_{i+1}
>1	$=1$	$\alpha_{0,i} \neq 0$	n_i	$n_i = n_{i+1}-1$	n_{i+1}
$=1$	$=1$	$\alpha_{0,i} - \varphi_j^\wedge \neq 0$	n_i	$n_i = n_{i+1}-1$	n_{i+1}

$h_{j-1}^\vee$	h_i	$\tilde{\varphi}_j^\vee$	$h_j^\wedge$	$h_j^\vee$	$\mathbf{e}_j^\wedge$	$\varphi_j^\wedge$	$\mathbf{e}_j^\vee$	$\varphi_j^\vee$
>1	>1	$=0$	h_i	$h_{i+1}+1$	$\Omega_j^\wedge(\mathbf{a}_i)$	β_i	$\mathbf{a}_{i+1}$	β_{i+1}
$=1$	>1	$=0$	h_i	$h_{i+1}+1$	$\Omega_j^\wedge(\mathbf{a}_i)$	$\beta_i/\varphi_{j-1}^\vee$	$\mathbf{a}_{i+1}$	β_{i+1}
>1	$=1$	$=0$	h_i	$h_{i+1}+1$	$\emptyset$	β_i	$\mathbf{a}_{i+1}$	β_{i+1}
$=1$	$=1$	$=0$	h_i	$h_{i+1}+1$	$\emptyset$	$\beta_i/\varphi_{j-1}^\vee$	$\mathbf{a}_{i+1}$	β_{i+1}
>1	>1	$\neq 0$	h_i	1	$\Omega_j^\wedge(\mathbf{a}_i)$	β_i	$\emptyset$	$\tilde{\varphi}_j^\vee$
$=1$	>1	$\neq 0$	h_i	1	$\Omega_j^\wedge(\mathbf{a}_i)$	$\beta_i/\varphi_{j-1}^\vee$	$\emptyset$	$\tilde{\varphi}_j^\vee$
>1	$=1$	$\neq 0$	h_i	1	$\emptyset$	β_i	$\emptyset$	$\tilde{\varphi}_j^\vee$
$=1$	$=1$	$\neq 0$	h_i	1	$\emptyset$	$\beta_i/\varphi_{j-1}^\vee$	$\emptyset$	$\tilde{\varphi}_j^\vee$

For the block UL decomposition $\mathbf{H}' = \mathbf{G}^\wedge\mathbf{G}^\vee$ we must likewise find a diagonal block $\mathbf{H}_j^\vee$ of $\mathbf{H}'$ of appropriate size, so that $\mathbf{H}_j^\vee - \mathbf{F}_{j+1}^\wedge\mathbf{L}_j^\vee$ is nonsingular and can be split into $\mathbf{G}_j^\wedge\mathbf{G}_j^\vee$:

$$\mathbf{H}_j^\vee - \mathbf{F}_{j+1}^\wedge\mathbf{L}_j^\vee = \mathbf{G}_j^\wedge\mathbf{G}_j^\vee\,. \tag{7.19}$$

In the case $W_m(z) = z^m$ any tentative 1×1 block pivot is again either a 1×1 matrix or a companion matrix. The condition for it to be nonsingular is now that $\alpha'_{0,i}$ minus the $(n'_i, n'_{i+1} - 1)$-element of $\mathbf{F}^{\wedge}_{j+1}\mathbf{L}^{\vee}_j$ does not vanish; i.e., in view of (7.8), $n'_i = n^{\vee}_j$ and $n'_{i+1} = n^{\wedge}_{j+1}$ (cf. Table 2) if and only if

$$\alpha'_{0,i} \neq 0 \text{ in case } h^{\wedge}_{j+1} > 1\,, \tag{7.20a}$$

$$\alpha'_{0,i} - \varphi^{\wedge}_{j+1} \neq 0 \text{ in case } h^{\wedge}_{j+1} = 1\,. \tag{7.20b}$$

However, $\varphi^{\wedge}_{j+1}$ is not known at this moment. But we can apply a different argument.

Let $n^{\vee}_j := n'_i$ and assume that the elements of $\mathbf{G}^{\vee}$ and $\mathbf{G}^{\wedge}$ have already been determined up to the $(n^{\vee}_j - 1)$th column. Then $h^{\wedge}_j$, $\mathbf{e}^{\wedge}_j$, and $\varphi^{\wedge}_j$ are known, and $h^{\vee}_j$ is equal to the dimension h'_i of $\mathbf{a}'_i$. From Table 2 we see that $\varphi^{\vee}_j$ is determined by β'_i and $\varphi^{\wedge}_j$. Moreover, by analogy to (7.17) and (7.18) there are functions $\Omega^{\vee}_j$ and $\omega^{\vee}_j$ inverting Ω'_j; the first one yields $\mathbf{e}^{\vee}_j$ if $h^{\vee}_j > 1$.

Next, let us first assume that $h^{\wedge}_j > 1$ and consider

$$\tilde{\varphi}^{\wedge}_{j+1} := \begin{cases} \omega^{\vee}_j(\mathbf{a}'_i) & \text{in case } h^{\vee}_j > 1, \\ \alpha'_{0,i} & \text{in case } h^{\vee}_j = 1 \end{cases} \tag{7.21}$$

as a tentative value for $\varphi^{\wedge}_{j+1}$. If nonvanishing, we let it be the true value of $\varphi^{\wedge}_{j+1}$ and set $h^{\wedge}_{j+1} := 1$ ($\mathbf{e}^{\wedge}_{j+1}$ is then void). Otherwise, $h^{\wedge}_{j+1} > 1$, hence $n^{\vee}_{j+1} := n'_{i+2}$, $\mathbf{e}^{\wedge}_{j+1} := \mathbf{a}'_{i+1}$, $\varphi^{\wedge}_{j+1} := \beta'_{i+1}$.

If $h^{\wedge}_j = 1$, we instead let

$$\tilde{\varphi}^{\wedge}_{j+1} := \begin{cases} \omega^{\vee}_j(\mathbf{a}'_i) - \varphi^{\vee}_j & \text{in case } h^{\vee}_j > 1, \\ \alpha'_{0,i} - \varphi^{\vee}_j & \text{in case } h^{\vee}_j = 1, \end{cases} \tag{7.22}$$

be the tentative value for $\varphi^{\wedge}_{j+1}$. The rest of the step is the same as before.

To complete the definition of the procedure we have to describe its start. At this point we must note that although the functional Φ' is uniquely determined by Φ, the converse is not true, since Φ depends on $\Phi(1) = \phi_l$, while Φ' is independent of ϕ_l. Hence, the set $\{P_n\}$ of FOP1s determined by Φ cannot be uniquely determined by the set $\{P'_n\}$ corresponding to Φ'. Moreover, the value of ϕ_l determines whether the $(l-1, 0)$ and the $(l, 0)$ Padé approximants of $f(z) = \sum \phi_k z^k$ belong to the same block of the table or not. In fact, they do if and only if $\phi_l = 0$. (Recall that these Padé approximants are polynomial interpolants.)

Therefore, given the recurrence formulas for $\{P'_n\}$ (i.e., given the matrix $\mathbf{H}'$), those for $\{P_n\}$ and those for the mixed recurrence (i.e., the matrices $\mathbf{H}$, $\mathbf{G}^{\vee}$ and $\mathbf{G}^{\wedge}$) are only determined after ϕ_l has been specified.

According to (5.17a), $\varphi^{\vee}_0$ satisfies

$$\varphi^{\vee}_0 \Phi(z^{n^{\vee}_0}) = \Phi(z^{n^{\vee}_0 + 1} W_{h^{\vee}_0 - 1} P'_{n^{\vee}_0}). \tag{7.23}$$

In the case $h^{\wedge}_0 = 1$, where $n^{\wedge}_0 := n^{\vee}_0 := n'_0 = 0$ (i.e., $i = j = 0$), $h^{\vee}_0 := h'_0, n^{\wedge}_1 := n'_1$ and where both $\mathbf{e}^{\wedge}_0$ and $\varphi^{\wedge}_0$ are void, we obtain

$$\varphi^{\vee}_0 := \Phi(z W_{n'_1 - 1})/\phi_l, \tag{7.24}$$

which for the monomial basis $W_m(z) = z^m$ reduces to

$$\varphi^{\vee}_0 := \phi_{l+n'_1}/\phi_l. \tag{7.25}$$

In the case $h_0^\wedge > 1$, where $n_0^\wedge := n_0' = 0, n_0^\vee := n_1'$ (i.e., $i = 1, j = 0$), $h_0^\wedge := h_0' + 1, h_0^\vee := h_1'$, we have $\mathbf{e}_0^\wedge := \mathbf{a}_0'$ (cf. (7.9a) and (7.9c) with (3.7)) and, from (7.23),

$$\varphi_0^\vee := \Phi(z^{n_1'+1} W_{h_1'-1} P_{n_1'}')/\phi_{l+n_1'}, \tag{7.26}$$

which for $W_m(z) = z^m$ reduces to

$$\varphi_0^\vee := \phi_{l+n_2'}/\phi_{l+n_1'}\,. \tag{7.27}$$

In both cases $\mathbf{e}_0^\vee, \mathbf{e}_1^\wedge$, and $\varphi_1^\wedge$ are obtained according to the general formulas in Table 4. Altogether we get the following analog of Theorem 7.2.

THEOREM 7.3. *Given* $\mathbf{H}'$ *and either* $\phi_l \neq 0$ *and* $\phi_{l+n_1'}$ *or* $\phi_l = 0$, $\phi_{l+n_1'}$, *and* $\phi_{l+n_2'}$, *we can compute the block* UL *factorization* $\mathbf{H}' = \mathbf{G}^\wedge \mathbf{G}^\vee$ *by the following process. In step* j, *where we compute columns* $n_j^\vee = n_i'$ *through* $n_{j+1}^\vee - 1$ *of* $\mathbf{G}^\vee$ *and* $\mathbf{G}^\wedge$, *the diagonal block* $\mathbf{H}_j^\vee$ *of* $\mathbf{H}'$ (*containing rows and columns* $n_j^\wedge$ *through* $n_{j+1}^\wedge - 1$) *is defined as either the* 1×1 *block* $\mathbf{H}_j^\vee := \mathbf{A}_j'$ *or the* 2×2 *block*

$$\mathbf{H}_j^\vee := \begin{bmatrix} \mathbf{A}_{i-1}' & \mathbf{B}_i' \\ \mathbf{C}_{i-1}' & \mathbf{A}_i' \end{bmatrix}, \tag{7.28}$$

depending on whether the tentative value $\tilde{\varphi}_{j+1}^\wedge$ *of* $\varphi_{j+1}^\wedge$ *given in Table* 4 *is nonzero or zero. The relevant entries* $\mathbf{e}_j^\wedge, \varphi_j^\wedge, \mathbf{e}_{j+1}^\wedge$, *and* $\varphi_{j+1}^\vee$ *of* $\mathbf{G}^\vee$ *and* $\mathbf{G}^\wedge$ *are then obtained according to Table* 4. *There the functions* $\Omega_j^\vee$ *and* $\omega_j^\vee$ *are together the inverse of* Ω_j' *defined in* (7.3b); *they are computed as follows: Partition the matrix* $\mathbf{T}_{h_j^\vee - 1}^W$ *and the vector* $\mathbf{a}_i'$ *according to*

$$\mathbf{T}_{h_j^\vee - 1}^W =: \left[\begin{array}{c|c} \mathbf{s}^T & \tau \\ \hline \mathbf{S} & \mathbf{t} \end{array}\right], \quad \mathbf{a}_i' =: \left[\begin{array}{c} \alpha \\ \hline \mathbf{a} \end{array}\right]; \tag{7.29}$$

then

$$\Omega_j^\vee(\mathbf{a}_i') := \mathbf{S}^{-1}(\mathbf{a} + \mathbf{t}), \tag{7.30a}$$

$$\omega_j^\vee(\mathbf{a}_i') := \alpha + \tau - \mathbf{s}^T \Omega_j^\vee(\mathbf{a}_i')\,. \tag{7.30b}$$

The first step depends on the value $\phi_l = \Phi(1)$ (*on which* $\mathbf{H}'$ *does not depend, while* $\mathbf{H}$ *does*):

If $\phi_l \neq 0$, *then*

$$h_0^\wedge := 1, \quad h_0^\vee := h_0', \tag{7.31a}$$

$$n_0^\wedge := n_0^\vee := n_0' = 0, \quad n_1^\wedge := n_1', \tag{7.31b}$$

$\mathbf{e}_0^\wedge$ *is void, and* $\varphi_0^\vee$ *is given by* (7.24); $\mathbf{e}_0^\vee$, $\mathbf{e}_1^\wedge$, $\varphi_0^\wedge$ *are then obtained from the general formulas in Table* 4, *with* $i = j = 0$.

If $\phi_l = 0$, *then*

$$h_0^\wedge := h_0' + 1, \quad h_0^\vee := h_1', \tag{7.32a}$$

$$n_0^\wedge := n_0' = 0, \quad n_0^\vee := n_1', \quad n_2^\wedge := n_2', \tag{7.32b}$$

$$\mathbf{e}_0^\wedge := \mathbf{a}_0', \tag{7.32c}$$

and $\varphi_0^\vee$ *is given by* (7.26); $\mathbf{e}_0^\vee$, $\mathbf{e}_1^\wedge$, $\varphi_0^\wedge$ *are obtained from Table* 4 *by setting* $i = 1$, $j = 0$.

TABLE 4
Formulas for the block UL *factorization* $\mathbf{H}' = \mathbf{G}^{\wedge}\mathbf{G}^{\vee}$.

$h_j^{\wedge}$	h_i'	$\tilde{\varphi}_{j+1}^{\wedge}$	$n_j^{\vee}$	$n_{j+1}^{\wedge}$	$n_{j+1}^{\vee}$
>1	>1	$\omega_j^{\vee}(\mathbf{a}_i') = 0$	n_i'	n_{i+1}'	n_{i+2}'
$=1$	>1	$\omega_j^{\vee}(\mathbf{a}_i') - \varphi_j^{\vee} = 0$	n_i'	n_{i+1}'	n_{i+2}'
>1	$=1$	$\alpha_{0,i}' = 0$	n_i'	$n_{i+1}' = n_i' + 1$	n_{i+2}'
$=1$	$=1$	$\alpha_{0,i}' - \varphi_j^{\vee} = 0$	n_i'	$n_{i+1}' = n_i' + 1$	n_{i+2}'
>1	>1	$\omega_j^{\vee}(\mathbf{a}_i') \neq 0$	n_i'	n_{i+1}'	n_{i+1}'
$=1$	>1	$\omega_j^{\vee}(\mathbf{a}_i') - \varphi_j^{\vee} \neq 0$	n_i'	n_{i+1}'	n_{i+1}'
>1	$=1$	$\alpha_{0,i}' \neq 0$	n_i'	$n_{i+1}' = n_i' + 1$	n_{i+1}'
$=1$	$=1$	$\alpha_{0,i}' - \varphi_j^{\vee} \neq 0$	n_i'	$n_{i+1}' = n_i' + 1$	n_{i+1}'

$h_j^{\wedge}$	h_i'	$\tilde{\varphi}_{j+1}^{\wedge}$	$h_j^{\vee}$	$h_{j+1}^{\wedge}$	$\mathbf{e}_j^{\vee}$	$\varphi_j^{\vee}$	$\mathbf{e}_{j+1}^{\wedge}$	$\varphi_{j+1}^{\wedge}$
>1	>1	$=0$	h_i'	$h_{j+1}' + 1$	$\Omega_j^{\vee}(\mathbf{a}_i')$	β_i'	$\mathbf{a}_i'$	β_i'
$=1$	>1	$=0$	h_i'	$h_{j+1}' + 1$	$\Omega_j^{\vee}(\mathbf{a}_i')$	$\beta_i'/\varphi_j^{\wedge}$	$\mathbf{a}_i'$	β_i'
>1	$=1$	$=0$	h_i'	$h_{j+1}' + 1$	$\emptyset$	β_i'	$\mathbf{a}_i'$	β_i'
$=1$	$=1$	$=0$	h_i'	$h_{j+1}' + 1$	$\emptyset$	$\beta_i'/\varphi_j^{\wedge}$	$\mathbf{a}_i'$	β_i'
>1	>1	$\neq 0$	h_i'	1	$\Omega_j^{\vee}(\mathbf{a}_i')$	β_i'	$-$	$\tilde{\varphi}_{j+1}^{\wedge}$
$=1$	>1	$\neq 0$	h_i'	1	$\Omega_j^{\vee}(\mathbf{a}_i')$	$\beta_i'/\varphi_j^{\wedge}$	$-$	$\tilde{\varphi}_{j+1}^{\wedge}$
>1	$=1$	$\neq 0$	h_i'	1	$\emptyset$	β_i'	$-$	$\tilde{\varphi}_{j+1}^{\wedge}$
$=1$	$=1$	$\neq 0$	h_i'	1	$\emptyset$	$\beta_i'/\varphi_j^{\wedge}$	$-$	$\tilde{\varphi}_{j+1}^{\wedge}$

Starting from the recurrence coefficients for some sequence $\{P_n\}_{n=0}^{\infty}$ of FOP1s, say from $\{P_{n;0}\}_{n=0}^{\infty}$, inductive application of Theorem 7.2 and of the $\mathbf{H}' = \mathbf{G}^{\wedge}\mathbf{G}^{\vee}$ part of Theorem 7.1 allows us to compute the recurrence coefficients of any sequence $\{P_{n;l}\}_{n=0}^{\infty}$, $l = 1, 2, \ldots$. These two theorems therefore define the *progressive qd algorithm*, even for nongeneric situations. In the generic case, for which the algorithm is due to Rutishauser [23], only the formulas for $h_j^{\wedge} = h_j^{\vee} = h_i = h_i' = 1$ $(\forall i, j)$ are used, i.e., only those in the last rows of Tables 2 and 3. In this generic case the *qd table* contains the recurrence coefficients for every diagonal sequence of FOP1s, i.e., our coefficients $\alpha_{0,i}$ $(i = 0, 1, \ldots)$ and β_i $(i = 1, 2, \ldots)$ for every diagonal $(l = 0, 1, \ldots)$. The progressive qd algorithm allows us to build up the qd table from its main diagonal, where $l = 0$. (More generally, one can proceed downwards from any diagonal or row.) Rutishauser had some heuristic rules for dealing with nongeneric situations, namely, rules for filling the then appearing gaps in the qd table with zeros and ∞ symbols. Draux [6] also formulated and established such rules. However, according to the above result, we can define a qd table that is valid in every nongeneric situation and contains as entries on its lth diagonal the nontrivial entries $\mathbf{a}_i$ $(i = 0, 1, \ldots)$ and β_i $(i = 1, 2, \ldots)$ of the matrix $\mathbf{H}$ for this l.

Likewise, starting from the recurrence coefficients of $\{P_{n;0}\}_{n=0}^{\infty}$, inductive application of Theorem 7.3 and of the $\mathbf{H} = \mathbf{G}^{\vee}\mathbf{G}^{\wedge}$ part of Theorem 7.1 allows us to compute those of any sequence $\{P_{n;l}\}_{n=0}^{\infty}$, $l = -1, -2, \ldots$. In each step a new "moment" ϕ_l, $l = -1, -2, \ldots$, has to be provided. Hence, we can proceed from the main (or any other) diagonal upwards and to the right. For the generic case this process is well known. We call this the *backward qd algorithm.*

The progressive qd algorithm enables us in particular to compute the moments

ϕ_l $(l = 0, 1, \ldots)$ from the coefficients $\mathbf{a}_i$ $(i = 0, 1, \ldots)$ and β_i $(i = 1, 2, \ldots)$ of the main diagonal. Conversely, given the moments, the ordinary qd algorithm yields in the generic case the recurrence coefficients on the main diagonal. This process is known to be highly unstable. The same task can be done with the *Chebyshev algorithm*, which is less likely to break down or to be unstable, since it requires only that all FOP1s $P_{0;n}$ on the main diagonal are regular. However, often the problem itself is ill conditioned, and there is no chance for numerically stable computations. It has, therefore, been proposed to replace the moments by modified moments if possible. Golub and Gutknecht [9] have extended the corresponding *modified Chebyshev algorithm* to the nongeneric case. The nongeneric Chebyshev algorithm is included there as a special case.

The progressive and the backward qd algorithms are well known to have interesting convergence properties. Basically, by proceeding downwards in the qd table we obtain the poles of f, and by moving to the right we find its zeros, see, e.g., [3].

8. The nongeneric biconjugate gradient algorithm (BCG or BIOMIN) and nongeneric BIODIR. The biconjugate gradient (BCG) algorithm is closely related to the Lanczos biorthogonalization (BO) method. It can be traced back to Lanczos [20], where it was introduced as "the complete algorithm for minimized iterations." More than 20 years later, Fletcher [7] revived and popularized it. It generates the same biorthogonal vector sequences $\{\mathbf{x}_n\}$, $\{\mathbf{y}_n\}$ characterized by (4.6)–(4.7), and the same iterates $\{\mathbf{z}_n\}$ satisfying (4.61) as the normalized BIORES algorithm [11], but additionally it generates two biconjugate vector sequences $\{\mathbf{u}_n\}$ and $\{\mathbf{v}_n\}$ taken from the same nested sequences of Krylov spaces as $\{\mathbf{x}_n\}$ and $\{\mathbf{y}_n\}$:

$$\mathbf{u}_n \in \mathcal{K}_{n+1} := \text{span}\,(\mathbf{x}_0, \mathbf{A}\mathbf{x}_0, \mathbf{A}^2\mathbf{x}_0, \ldots, \mathbf{A}^n\mathbf{x}_0), \tag{8.1a}$$

$$\mathbf{v}_n \in \mathcal{L}_{n+1} := \text{span}\,(\mathbf{y}_0, \mathbf{A}^H\mathbf{y}_0, (\mathbf{A}^H)^2\mathbf{y}_0, \ldots, (\mathbf{A}^H)^n\mathbf{y}_0) \tag{8.1b}$$

with

$$\langle \mathbf{v}_m, \mathbf{A}\mathbf{u}_n \rangle_{\mathbf{B}} \begin{cases} = 0 & \text{if } m \neq n, \\ \neq 0 & \text{if } m = n. \end{cases} \tag{8.2}$$

This process can break down for various reasons, cf. [11]. From §4 we know already that $\{\mathbf{x}_n\}$ and $\{\mathbf{y}_n\}$ satisfying (4.6) and (4.7) may not exist and that a suitable modification of the process can be based on the theory of formal orthogonal polynomials. An argument analogous to the one given in §4 shows that if (8.1) and (8.2) can be fulfilled for $n = 0, 1, \ldots, \nu - 1$, then they are fulfilled by

$$\mathbf{u}_n = P'_n(\mathbf{A})\mathbf{x}_0\Gamma'_n, \quad \mathbf{v}_n = \overline{P'_n}(\mathbf{A}^H)\mathbf{y}_0\bar{\Gamma}'_n, \tag{8.3}$$

where the scale factors Γ'_n and $\bar{\Gamma}'_n$ are not necessarily the same as the factors Γ_n and $\bar{\Gamma}_n$ in (4.14), and where P'_n is the nth monic FOP1 with respect to the linear functional $\Phi' = \Phi_1$ defined by

$$\Phi'(z^k) := \phi_{k+1} := \langle \mathbf{y}_0, \mathbf{A}^{k+1}\mathbf{x}_0 \rangle_{\mathbf{B}}. \tag{8.4}$$

(Recall that if (8.2) holds for all $m < \nu$ and $n < \nu$, all the polynomials P'_n $(n < \nu)$ in (8.3) are regular FOP1s.)

In case of a breakdown, the formulas (8.3) point again to the correct generalization of the process: $\mathbf{u}_n$ and $\mathbf{v}_n$ must still have the same form, with P'_n being an nth FOP1 for Φ', even if it is not a regular FOP1; hence (8.2) does not hold for this n. The

recurrences of §5 allow us to find a recursive algorithm for computing $\mathbf{u}_n$ and $\mathbf{v}_n$ along with $\mathbf{x}_n$ and $\mathbf{y}_n$, and the relations of §§6 and 7 yield corresponding matrix results.

First we formulate the matrix algorithm that is based on a reinterpretation of Theorems 5.3 and 5.4. Since it yields both a pair of biorthogonal and a pair of biconjugate sequences, we call it, as in [11], the *BOBC algorithm.*

As an extension of (4.20a), for $i, n \in \mathbb{N}$ we let

$$\gamma_{n,i} := \Gamma_n/\Gamma_{n-i}, \quad \bar{\gamma}_{n,i} := \bar{\Gamma}_n/\bar{\Gamma}_{n-i}, \tag{8.5a}$$

$$\gamma'_{n,i} := \Gamma'_n/\Gamma'_{n-i}, \quad \bar{\gamma}'_{n,i} := \bar{\Gamma}'_n/\bar{\Gamma}'_{n-i}, \tag{8.5b}$$

and

$$\gamma^{\wedge}_{n,i} := \Gamma_n/\Gamma'_{n-i}, \quad \bar{\gamma}^{\wedge}_{n,i} := \bar{\Gamma}_n/\bar{\Gamma}'_{n-i}, \tag{8.6a}$$

$$\gamma^{\vee}_{n,i} := \Gamma'_n/\Gamma_{n-i}, \quad \bar{\gamma}^{\vee}_{n,i} := \bar{\Gamma}'_n/\bar{\Gamma}_{n-i}\,. \tag{8.6b}$$

Recall that the mixed three-term recurrence relations (5.25) allow us to generate the two sequences $\{P_{n_j^{\wedge}}\}_{j=0}^{J^{\wedge}}$ and $\{P'_{n_j^{\vee}}\}_{j=0}^{J^{\vee}}$ consisting of regular FOP1s for Φ and Φ', respectively, and that two full sequences of FOP1s for these two functionals are then defined by (5.27). If $h_j^{\wedge} := n_j^{\vee} - n_j^{\wedge} + 1 > 1$ or $h_j^{\vee} := n_{j+1}^{\wedge} - n_j^{\vee} > 1$, some polynomials on the two diagonals coincide or differ only by a factor of z, cf. (5.28); consequently,

$$\mathbf{x}_n = \mathbf{u}_n \gamma^{\wedge}_{n,0}, \quad \mathbf{y}_n = \mathbf{v}_n \bar{\gamma}^{\wedge}_{n,0}, \quad \text{if } n_j^{\wedge} \le n < n_j^{\vee}, \tag{8.7a}$$

$$\mathbf{x}_n = \mathbf{A}\mathbf{u}_{n-1}\gamma^{\wedge}_{n,1}, \quad \mathbf{y}_n = \mathbf{A}^H \mathbf{v}_{n-1}\bar{\gamma}^{\wedge}_{n,1}, \quad \text{if } n_j^{\vee} < n < n_{j+1}^{\wedge}\,. \tag{8.7b}$$

The polynomials W_m in (5.27) are for practicality again assumed to satisfy the three-term recurrence (2.10), so that the recurrences (5.29) hold, which translate into

$$\mathbf{x}_{n+1} = [\mathbf{A}\mathbf{x}_n - \mathbf{x}_n \alpha^W_{n-n_j^{\wedge}}]\gamma_{n+1,1} - \mathbf{x}_{n-1}\beta^W_{n-n_j^{\wedge}}\gamma_{n+1,2}, \quad n_j^{\wedge} \le n \le n_j^{\vee} - 1, \tag{8.8a}$$

$$\mathbf{x}_{n+1} = [\mathbf{A}\mathbf{x}_n - \mathbf{x}_n \alpha^W_{n-n_j^{\vee}+1}]\gamma_{n+1,1} - \mathbf{x}_{n-1}\beta^W_{n-n_j^{\vee}+1}\gamma_{n+1,2}, \quad n_j^{\vee} < n \le n_{j+1}^{\wedge} - 2, \tag{8.8b}$$

$$\mathbf{u}_{n+1} = [\mathbf{A}\mathbf{u}_n - \mathbf{u}_n \alpha^W_{n-n_j^{\wedge}}]\gamma'_{n+1,1} - \mathbf{u}_{n-1}\beta^W_{n-n_j^{\wedge}}\gamma'_{n+1,2}, \quad n_j^{\wedge} \le n \le n_j^{\vee} - 2, \tag{8.8c}$$

$$\mathbf{u}_{n+1} = [\mathbf{A}\mathbf{u}_n - \mathbf{u}_n \alpha^W_{n-n_j^{\vee}}]\gamma'_{n+1,1} - \mathbf{u}_{n-1}\beta^W_{n-n_j^{\vee}}\gamma'_{n+1,2}, \quad n_j^{\vee} \le n \le n_{j+1}^{\wedge} - 2\,. \tag{8.8d}$$

Of course, analogous formulas with the complex conjugate coefficients $\overline{\alpha^W_m}$ and $\overline{\beta^W_m}$, with the scale factors $\bar{\gamma}_{n,i}$ and $\bar{\gamma}'_{n,i}$ of (8.6) and with $\mathbf{A}$ replaced by $\mathbf{A}^H$ hold for $\{\mathbf{y}_n\}$ and $\{\mathbf{v}_n\}$, but from now on we only give those for $\{\mathbf{x}_n\}$ and $\{\mathbf{u}_n\}$. For simplicity we refer to these analogous formulas as the *conjugate* recurrences, although $\bar{\gamma}^{\wedge}_{n,i}$ and $\bar{\gamma}^{\vee}_{n,i}$ need not be complex conjugate to $\gamma^{\wedge}_{n,i}$ and $\gamma^{\vee}_{n,i}$.

If $\varepsilon^{\vee}_{i,j}$ and $\varepsilon^{\wedge}_{i,j}$ denote the coefficients of the polynomials $e_j^{\vee}$ and $e_j^{\wedge}$, respectively, (as is the case in (6.5)), the mixed three-term recurrence formulas (5.25) yield

$$\begin{aligned}\mathbf{u}_{n_j^{\vee}} = {} & \mathbf{x}_{n_j^{\vee}}\gamma^{\vee}_{n_j^{\vee},0} - \mathbf{x}_{n_j^{\vee}-1}\varepsilon^{\wedge}_{h_j^{\wedge}-2,j}\gamma^{\vee}_{n_j^{\vee},1} - \mathbf{x}_{n_j^{\vee}-2}\varepsilon^{\wedge}_{h_j^{\wedge}-3,j}\gamma^{\vee}_{n_j^{\vee},2} - \cdots \\ & - \mathbf{x}_{n_j^{\wedge}}\varepsilon^{\wedge}_{0,j}\gamma^{\vee}_{n_j^{\vee},h_j^{\wedge}-1} - \mathbf{u}_{n_{j-1}^{\vee}}\varphi^{\wedge}_j\gamma'_{n_j^{\vee},h_j^{\wedge}+h_{j-1}^{\vee}-1},\end{aligned} \tag{8.9a}$$

when $h_j^{\wedge} > 1$, cf. (5.27a) and (5.31a). In view of (5.27b), (5.29b), and (5.28b), when $h_j^{\vee} > 1$,

(8.9b)
$$\begin{aligned}\mathbf{x}_{n_{j+1}^{\wedge}} &= \mathbf{A}[\mathbf{u}_{n_{j+1}^{\wedge}-1}\gamma^{\wedge}_{n_{j+1}^{\wedge},1} - \mathbf{u}_{n_{j+1}^{\wedge}-2}\varepsilon^{\vee}_{h_j^{\vee}-2,j}\gamma^{\wedge}_{n_{j+1}^{\wedge},2} - \mathbf{u}_{n_{j+1}^{\wedge}-3}\varepsilon^{\vee}_{h_j^{\vee}-3,j}\gamma^{\wedge}_{n_{j+1}^{\wedge},3} - \cdots \\ &\quad - \mathbf{u}_{n_j^{\vee}}\varepsilon^{\vee}_{0,j}\gamma^{\wedge}_{n_{j+1}^{\wedge},h_j^{\vee}}] - \mathbf{x}_{n_j^{\wedge}}\varphi_j^{\vee}\gamma_{n_{j+1}^{\wedge},h_j^{\vee}+h_j^{\wedge}-1}\end{aligned}$$

(8.9c)
$$\begin{aligned}&= \mathbf{A}\mathbf{u}_{n_{j+1}^{\wedge}-1}\gamma^{\wedge}_{n_{j+1}^{\wedge},1} - \mathbf{x}_{n_{j+1}^{\wedge}-1}\varepsilon^{\vee}_{h_j^{\vee}-2,j}\gamma_{n_{j+1}^{\wedge},1} - \mathbf{x}_{n_{j+1}^{\wedge}-2}\varepsilon^{\vee}_{h_j^{\vee}-3,j}\gamma_{n_{j+1}^{\wedge},2} - \cdots \\ &\quad - \mathbf{x}_{n_j^{\vee}+1}\varepsilon^{\vee}_{0,j}\gamma_{n_{j+1}^{\wedge},h_j^{\vee}-1} - \mathbf{x}_{n_j^{\wedge}}\varphi_j^{\vee}\gamma_{n_{j+1}^{\wedge},h_j^{\vee}+h_j^{\wedge}-1}\end{aligned}$$

(8.9d)
$$\begin{aligned}&= [\mathbf{A}\mathbf{x}_{n_{j+1}^{\wedge}-1} - \mathbf{x}_{n_{j+1}^{\wedge}-1}(\varepsilon^{\vee}_{h_j^{\vee}-2,j} + \alpha^{W}_{h_j^{\vee}-3})]\gamma_{n_{j+1}^{\wedge},1} \\ &\quad - \mathbf{x}_{n_{j+1}^{\wedge}-2}(\varepsilon^{\vee}_{h_j^{\vee}-3,j} + \beta^{W}_{h_j^{\vee}-3})\gamma_{n_{j+1}^{\wedge},2} - \mathbf{x}_{n_{j+1}^{\wedge}-3}\varepsilon^{\vee}_{h_j^{\vee}-4,j}\gamma_{n_{j+1}^{\wedge},3} \\ &\quad - \cdots - \mathbf{x}_{n_j^{\vee}+1}\varepsilon^{\vee}_{0,j}\gamma_{n_{j+1}^{\wedge},h_j^{\vee}-1} - \mathbf{x}_{n_j^{\wedge}}\varphi_j^{\vee}\gamma_{n_{j+1}^{\wedge},h_j^{\vee}+h_j^{\wedge}-1}\end{aligned}$$

(cf. (5.31b)). However, when $h_j^{\wedge} = 1$, i.e., $n_j^{\vee} = n_j^{\wedge}$, then $W_{h_j^{\wedge}-1}(z) \equiv 1$ and $e_j^{\wedge}(z) \equiv 0$, so that instead of (8.9a) we simply obtain

(8.9e)
$$\mathbf{u}_{n_j^{\vee}} = \mathbf{x}_{n_j^{\wedge}}\gamma^{\vee}_{n_j^{\vee},0} - \mathbf{u}_{n_{j-1}^{\vee}}\varphi_j^{\wedge}\gamma'_{n_j^{\vee},h_{j-1}^{\vee}}.$$

Likewise, if $h_j^{\vee} = 1$, i.e., $n_{j+1}^{\wedge} = n_j^{\vee} + 1$, (8.8b)–(8.8d) are replaced by

(8.9f)
$$\mathbf{x}_{n_{j+1}^{\wedge}} = \mathbf{A}\mathbf{u}_{n_j^{\vee}}\gamma^{\wedge}_{n_{j+1}^{\wedge},1} - \mathbf{x}_{n_j^{\wedge}}\varphi_j^{\vee}\gamma_{n_{j+1}^{\wedge},h_j^{\wedge}}.$$

It remains to give formulas for the index sequences $\{n_j^{\wedge}\}$ and $\{n_j^{\vee}\}$ and for the coefficients $\varepsilon^{\vee}_{i,j}$, $\varepsilon^{\wedge}_{i,j}$, $\varphi_j^{\vee}$, and $\varphi_j^{\wedge}$ that appear in (8.9). First, according to (5.26) and (5.27), $h_j^{\wedge} := n_j^{\vee} - n_j^{\wedge} + 1$ and $h_j^{\vee} := n_{j+1}^{\wedge} - n_j^{\vee}$ are given by

(8.10a)
$$h_j^{\wedge} := \min\{k \in \mathbb{N}^+; \langle \mathbf{y}_{n_j^{\wedge}}, \mathbf{x}_{n_j^{\wedge}+k-1}\rangle_{\mathbf{B}} \neq 0\},$$

(8.10b)
$$h_j^{\vee} := \min\{k \in \mathbb{N}^+; \langle \mathbf{v}_{n_j^{\vee}}, \mathbf{A}\mathbf{u}_{n_j^{\vee}+k-1}\rangle_{\mathbf{B}} \neq 0\}.$$

Second, equations for the mentioned coefficients follow from Theorem 5.4; we choose relations (5.32a)–(5.32c) and (5.32f) in order to work without $\mathbf{v}_{n_j^{\vee}+1}, \ldots, \mathbf{v}_{n_{j+1}^{\wedge}-1}$:

(8.11a)
$$\varphi_j^{\wedge}\gamma^{\wedge}_{n_j^{\wedge},h_{j-1}^{\vee}}\overline{\bar{\gamma}_{n_j^{\vee},h_j^{\wedge}}}\langle \mathbf{y}_{n_j^{\wedge}-1}, \mathbf{A}\mathbf{u}_{n_{j-1}^{\vee}}\rangle_{\mathbf{B}} = \langle \mathbf{y}_{n_j^{\vee}}, \mathbf{x}_{n_j^{\wedge}}\rangle_{\mathbf{B}},$$

(8.11b)
$$\sum_{s=1}^{k} \varepsilon^{\wedge}_{h_j^{\wedge}-s-1,j}\gamma_{n_j^{\vee},s}\langle \mathbf{y}_{n_j^{\wedge}+k}, \mathbf{x}_{n_j^{\vee}-s}\rangle_{\mathbf{B}} = \langle \mathbf{y}_{n_j^{\wedge}+k}, \mathbf{x}_{n_j^{\vee}}\rangle_{\mathbf{B}}, \qquad k = 1, \ldots, h_j^{\wedge} - 1,$$

(8.11c)
$$\varphi_j^{\vee}\gamma^{\vee}_{n_j^{\vee},h_j^{\wedge}-1}\overline{\bar{\gamma}_{n_{j+1}^{\wedge}-1,h_j^{\vee}-1}}\langle \mathbf{y}_{n_j^{\vee}}, \mathbf{x}_{n_j^{\wedge}}\rangle_{\mathbf{B}} = \langle \mathbf{y}_{n_{j+1}^{\wedge}-1}, \mathbf{A}\mathbf{u}_{n_j^{\vee}}\rangle_{\mathbf{B}},$$

(8.11d)
$$\begin{aligned}&\sum_{s=1}^{k} \varepsilon^{\vee}_{h_j^{\vee}-s-1,j}\overline{\bar{\gamma}_{n_{j+1}^{\wedge}-1,s}}\langle \mathbf{y}_{n_{j+1}^{\wedge}-s-1}, \mathbf{u}_{n_j^{\vee}+k}\rangle_{\mathbf{B}} \\ &= \langle \mathbf{A}^H\mathbf{y}_{n_{j+1}^{\wedge}-1} - \mathbf{y}_{n_{j+1}^{\wedge}-1}\overline{\alpha^{W}_{h_j^{\vee}-2}} - \mathbf{y}_{n_{j+1}^{\wedge}-2}\overline{\beta^{W}_{h_j^{\vee}-2}}\bar{\gamma}_{n_{j+1}^{\wedge}-1,1}, \mathbf{u}_{n_j^{\vee}+k}\rangle_{\mathbf{B}}, \\ &\qquad\qquad k = 1, \ldots, h_j^{\vee} - 1.\end{aligned}$$

Again, (8.11a) and (8.11c) are single linear equations, hence explicit formulas, for $\varphi_j^{\wedge}$ and $\varphi_j^{\vee}$, respectively. Equations (8.11b) and (8.11d) are triangular systems for the

coefficients $\varepsilon^{\wedge}_{s,j}$ and $\varepsilon^{\vee}_{s,j}$ in the representation (6.5) of the polynomials $e^{\wedge}_j$ and $e^{\vee}_j$. If $h^{\wedge}_j = 1$, the system (8.11b) is void, and, if $h^{\vee}_j = 1$, the system (8.11d) is void.

We now obtain the following algorithm.

ALGORITHM 4 (NONGENERIC BOBC ALGORITHM). *Given a bounded linear operator* $\mathbf{A} : \mathcal{H} \to \mathcal{H}$ *and two initial vectors* $\mathbf{x}_0, \mathbf{y}_0 \in \mathcal{H}$ *satisfying* $\langle \mathbf{y}_0, \mathbf{x}_0 \rangle_{\mathbf{B}} \neq 0$, *set* $\mathbf{u}_0 := \mathbf{x}_0$, $\mathbf{v}_0 := \mathbf{y}_0$, $h^{\wedge}_0 := 1$. *Then construct sequences* $\{\mathbf{x}_n\}_{n=0}^{\infty}$, $\{\mathbf{y}_n\}_{n=0}^{\infty}$, $\{\mathbf{u}_n\}_{n=0}^{\infty}$, *and* $\{\mathbf{v}_n\}_{n=0}^{\infty}$ *according to the inductive process, which, for* $j = 0, 1, \ldots$, *consists of:*

(i) *If* $h^{\wedge}_j > 1$, *then* $\{\mathbf{x}_{n^{\wedge}_j+k}\}_{k=1}^{h^{\wedge}_j-1}$, $\{\mathbf{y}_{n^{\wedge}_j+k}\}_{k=0}^{h^{\wedge}_j-1}$, *and* $h^{\wedge}_j$ *are defined by executing concurrently* (8.8a), *the corresponding conjugate recurrence for* $\mathbf{y}_{n^{\wedge}_j+k}$, *and* (8.10a); *if* $h^{\wedge}_j = \infty$, *then* $J^{\wedge} := j$, $J^{\vee} := j - 1$; *in particular, if* $\mathbf{x}_{n^{\wedge}_j} = \mathbf{0}$ *or* $\mathbf{y}_{n^{\wedge}_j} = \mathbf{0}$, *then* $h^{\wedge}_j = \infty$ *and* $\mathbf{x}_{n^{\wedge}_j+k} = \mathbf{0}$ $(\forall k \geq 0)$ *or* $\mathbf{y}_{n^{\wedge}_j+k} = \mathbf{0}$ $(\forall k \geq 0)$, *respectively, and the algorithm terminates (in practice,* $\mathbf{x}_{n^{\wedge}_j+k}$ *and* $\mathbf{y}_{n^{\wedge}_j+k}$ *are then not needed);*

(ii) *once* $h^{\wedge}_j$ *has been determined, the nonzero constant* $\varphi^{\wedge}_j$ *is given by* (8.11a) *and, if* $h^{\wedge}_j > 1$, *the coefficients* $\{\varepsilon^{\wedge}_{s,j}\}_{s=1}^{h^{\wedge}_j-2}$ *are obtained by solving the triangular linear system* (8.11b);

(iii) *depending on whether or not* $h^{\wedge}_j > 1$, $\mathbf{u}_{n^{\vee}_j}$ *and* $\mathbf{v}_{n^{\vee}_j}$ *are then given by* (8.9a) *or* (8.9e) *and the conjugate recurrence; if* $\langle \mathbf{v}_{n^{\vee}_j}, \mathbf{A}\mathbf{u}_{n^{\vee}_j} \rangle_{\mathbf{B}} \neq 0$, *set* $h^{\vee}_j := 1$; *otherwise* $h^{\vee}_j > 1$;

(iv) *If* $h^{\vee}_j > 1$, *then* $\{\mathbf{u}_{n^{\vee}_j+k}\}_{k=1}^{h^{\vee}_j-1}$, $\mathbf{y}_{n^{\vee}_j+1}$, $\{\mathbf{y}_{n^{\vee}_j+k}\}_{k=2}^{h^{\vee}_j-1}$, *and* $h^{\vee}_j$ *are defined by* (8.8d), (8.7b), *the recurrence conjugate to* (8.8b), *and by* (8.10b); *if* $h^{\vee}_j = \infty$, *then* $J^{\vee} := J^{\wedge} := j$; *in particular, if* $\mathbf{u}_{n^{\vee}_j} = \mathbf{0}$ *or* $\mathbf{v}_{n^{\vee}_j} = \mathbf{0}$, *then* $h^{\vee}_j = \infty$ *and* $\mathbf{x}_{n^{\wedge}_j+k} = \mathbf{0}$ $(\forall k \geq 1)$ *or* $\mathbf{y}_{n^{\wedge}_j+k} = \mathbf{0}$ $(\forall k \geq 1)$, *respectively, and the algorithm terminates (in practice,* $\mathbf{x}_{n^{\wedge}_j+k}$ *and* $\mathbf{y}_{n^{\wedge}_j+k}$ *are then not needed*);

(v) *once* $h^{\vee}_j$ *has been determined, the nonzero constant* $\varphi^{\vee}_j$ *is given by* (8.11c) *and, if* $h^{\vee}_j > 1$, *the coefficients* $\{\varepsilon^{\vee}_{s,j}\}_{s=1}^{h^{\vee}_j-2}$ *are obtained by solving the triangular linear system* (8.11d);

(vi) *depending on whether* $h^{\vee}_j > 1$ *or not,* $\mathbf{x}_{n^{\wedge}_{j+1}}$ *and* $\mathbf{y}_{n^{\wedge}_{j+1}}$ *are either given by* (8.9b) *and the conjugate recurrence to* (8.9d) *or by* (8.9f) *and its conjugate recurrence; if* $\langle \mathbf{v}_{n^{\wedge}_{j+1}}, \mathbf{u}_{n^{\wedge}_{j+1}} \rangle_{\mathbf{B}} \neq 0$, *set* $h^{\wedge}_{j+1} := 1$, *otherwise* $h^{\wedge}_{j+1} > 1$.

The recurrence coefficients α^{W}_m *and* β^{W}_m *in* (8.8) *and the nonvanishing scale factors* Γ_n, Γ'_n, $\bar{\Gamma}_n$, *and* $\bar{\Gamma}'_n$ $(n \in \mathbb{N})$, *which determine* $\gamma^{\wedge}_{n,i}$, $\gamma^{\vee}_{n,i}$, $\gamma_{n,i}$, $\gamma'_{n,i}$, $\bar{\gamma}^{\wedge}_{n,i}$, $\bar{\gamma}^{\vee}_{n,i}$, $\bar{\gamma}_{n,i}$, *and* $\bar{\gamma}'_{n,i}$ *according to* (8.5) *and* (8.6), *can be chosen freely. (For the sake of simplicity, we assume that* $\Gamma_0 := \Gamma'_0 := \bar{\Gamma}_0 := \bar{\Gamma}'_0 := 1$.)

As in Algorithm 1 (§4) we could set $\Gamma_n := \Gamma'_n := \bar{\Gamma}_n := \bar{\Gamma}'_n := 1$ $(n \in \mathbb{N})$, which would imply that $\gamma^{\wedge}_{n,i} = \gamma^{\vee}_{n,i} = \gamma_{n,i} = \gamma'_{n,i} = \bar{\gamma}^{\wedge}_{n,i} = \bar{\gamma}^{\vee}_{n,i} = \bar{\gamma}_{n,i} = \bar{\gamma}'_{n,i} = 1$ $(\forall n, \forall i)$, but might lead to overflow or underflow.

Of course, Algorithm 4 also has a matrix interpretation, which is analogous to the one for Algorithm 1 that was formulated in Theorem 4.2 of Part I.

THEOREM 8.1. *Gather the vectors generated by Algorithm* 4 *into*

$$\mathbf{X} := [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots], \quad \mathbf{Y} := [\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \ldots], \tag{8.12a}$$

$$\mathbf{U} := [\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots], \quad \mathbf{V} := [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots], \tag{8.12b}$$

and the scale factors used into

$$\mathbf{\Gamma} := \operatorname{diag}[\Gamma_0, \Gamma_1, \Gamma_2, \ldots], \quad \bar{\mathbf{\Gamma}} := \operatorname{diag}[\bar{\Gamma}_0, \bar{\Gamma}_1, \bar{\Gamma}_2, \ldots], \tag{8.13a}$$

(8.13b) $$\mathbf{\Gamma}' := \operatorname{diag}[\Gamma'_0, \Gamma'_1, \Gamma'_2, \ldots], \quad \bar{\mathbf{\Gamma}}' := \operatorname{diag}[\bar{\Gamma}'_0, \bar{\Gamma}'_1, \bar{\Gamma}'_2, \ldots].$$

Let $\mathbf{D}^\wedge$ *and* $\mathbf{D}^\vee$ *be the block diagonal matrices*

(8.14) $$\mathbf{D}^\wedge := \Phi((\mathbf{p})^T\mathbf{p}), \quad \mathbf{D}^\vee := \Phi'((\mathbf{p}')^T\mathbf{p}')$$

expressing the formal orthogonality of the two sequences of FOP1*s, and let*

(8.15) $$\mathbf{D}^\wedge_\Gamma := \overline{\bar{\mathbf{\Gamma}}}\mathbf{D}^\wedge\mathbf{\Gamma}, \quad \mathbf{D}^\vee_\Gamma := \overline{\bar{\mathbf{\Gamma}}'}\mathbf{D}^\vee\mathbf{\Gamma}'$$

be corresponding diagonally scaled matrices.[4] *Furthermore, using the block matrices* $\mathbf{E}^\wedge$, $\mathbf{E}^\vee$, $\mathbf{F}^\wedge$, $\mathbf{F}^\vee$, $\mathbf{G}^\wedge$, *and* $\mathbf{G}^\vee$ *from* §6, *introduce the scaled matrices*

(8.16a) $$\mathbf{E}^\wedge_\Gamma := (\mathbf{\Gamma}')^{-1}\mathbf{E}^\wedge\mathbf{\Gamma}, \quad \overline{\mathbf{E}^\wedge_\Gamma} := (\bar{\mathbf{\Gamma}}')^{-1}\overline{\mathbf{E}^\wedge}\bar{\mathbf{\Gamma}},$$

(8.16b) $$\mathbf{E}^\vee_\Gamma := (\mathbf{\Gamma})^{-1}\mathbf{E}^\vee\mathbf{\Gamma}', \quad \overline{\mathbf{E}^\vee_\Gamma} := (\bar{\mathbf{\Gamma}})^{-1}\overline{\mathbf{E}^\vee}\bar{\mathbf{\Gamma}}',$$

(8.16c) $$\mathbf{F}^\wedge_\Gamma := (\mathbf{\Gamma})^{-1}\mathbf{F}^\wedge\mathbf{\Gamma}, \quad \overline{\mathbf{F}^\wedge_\Gamma} := (\bar{\mathbf{\Gamma}})^{-1}\overline{\mathbf{F}^\wedge}\bar{\mathbf{\Gamma}},$$

(8.16d) $$\mathbf{F}^\vee_\Gamma := (\mathbf{\Gamma}')^{-1}\mathbf{F}^\vee\mathbf{\Gamma}', \quad \overline{\mathbf{F}^\vee_\Gamma} := (\bar{\mathbf{\Gamma}}')^{-1}\overline{\mathbf{F}^\vee}\bar{\mathbf{\Gamma}}',$$

(8.16e) $$\mathbf{G}^\wedge_\Gamma := (\mathbf{\Gamma})^{-1}\mathbf{G}^\wedge\mathbf{\Gamma}', \quad \overline{\mathbf{G}^\wedge_\Gamma} := (\bar{\mathbf{\Gamma}})^{-1}\overline{\mathbf{G}^\wedge}\bar{\mathbf{\Gamma}}',$$

(8.16f) $$\mathbf{G}^\vee_\Gamma := (\mathbf{\Gamma}')^{-1}\mathbf{G}^\vee\mathbf{\Gamma}, \quad \overline{\mathbf{G}^\vee_\Gamma} := (\bar{\mathbf{\Gamma}}')^{-1}\overline{\mathbf{G}^\vee}\bar{\mathbf{\Gamma}}.$$

Then Algorithm 4 *induces the relations*

(8.17a) $$\mathbf{A}\mathbf{U}\mathbf{E}^\wedge_\Gamma = \mathbf{X}\mathbf{F}^\wedge_\Gamma, \quad \mathbf{A}^H\mathbf{V}\overline{\mathbf{E}^\wedge_\Gamma} = \mathbf{Y}\overline{\mathbf{F}^\wedge_\Gamma},$$

(8.17b) $$\mathbf{X}\mathbf{E}^\vee_\Gamma = \mathbf{U}\mathbf{F}^\vee_\Gamma, \quad \mathbf{Y}\overline{\mathbf{E}^\vee_\Gamma} = \mathbf{V}\overline{\mathbf{F}^\vee_\Gamma},$$

which imply

(8.18a) $$\mathbf{A}\mathbf{U} = \mathbf{X}\mathbf{G}^\wedge_\Gamma, \quad \mathbf{A}^H\mathbf{V} = \mathbf{Y}\overline{\mathbf{G}^\wedge_\Gamma},$$

(8.18b) $$\mathbf{X} = \mathbf{U}\mathbf{G}^\vee_\Gamma, \quad \mathbf{Y} = \mathbf{V}\overline{\mathbf{G}^\vee_\Gamma}.$$

Moreover, if we write the infinite matrix with (m,n)*-element* $\langle \mathbf{y}_m, \mathbf{x}_n\rangle_\mathbf{B}$ *formally as* $\mathbf{Y}^H\mathbf{X}$, *and the one with* (m,n)*-element* $\langle \mathbf{v}_m, \mathbf{A}\mathbf{u}_n\rangle_\mathbf{B}$ *as* $\mathbf{V}^H\mathbf{A}\mathbf{U}$, *then we have*

(8.19) $$\mathbf{Y}^H\mathbf{X} = \mathbf{D}^\wedge_\Gamma, \quad \mathbf{V}^H\mathbf{A}\mathbf{U} = \mathbf{D}^\vee_\Gamma.$$

Note that (8.9b) corresponds to (8.17a), while (8.9c) translates into the relation (8.18a), which is equivalent to (8.17a).

From Algorithm 4 it is a small step to a nongeneric version of the *BCG method*, which also goes under the names *Lanczos/ORTHOMIN* [16] and *BIOMIN* [11]. This *normalized nongeneric BIOMIN algorithm* is a nearly straightforward application of the above BOBC algorithm to the problem of solving a linear system of equations $\mathbf{A}\mathbf{z} = \mathbf{b}$. As in the generic case [11], the basic strategy is to define a sequence $\{\mathbf{z}_n\}$ of approximants in such a way that the vectors $\mathbf{x}_n$ generated by Algorithm 4 are the residuals, which means that $\Gamma_n P_n$ is the nth residual polynomial. Consequently, for the normalized algorithm we have to choose $\Gamma_n := 1/P_n(0)$, thus producing a breakdown whenever $P_n(0) = 0$. However, the latter equality holds whenever $h^\vee_j > 1$ and $n^\vee_j < n < n^\wedge_{j+1}$, cf. (5.28b). This mirrors the fact that the restriction of $\mathbf{A}$ that

[4] The solid overbar denotes complex conjugation.

is implicitly constructed at this stage is singular, and, hence, the projected system cannot be solved in general. Recall that the same difficulty occurred in the nongeneric normalized BIORES algorithm (Algorithm 2 of Part I). But here, the difficulty is easier to recognize since it corresponds exactly to the case $h_j^\vee > 1$. It is also easier to understand how to circumnavigate it.

By translating (8.9b) into a polynomial recurrence (i.e., by inserting (8.3) and (4.14)) we see that

$$P_{n_{j+1}^\wedge}(0)\,\Gamma_{n_{j+1}^\wedge} = -P_{n_j^\wedge}(0)\,\Gamma_{n_j^\wedge}\varphi_j^\vee\gamma_{n_{j+1}^\wedge,h_j^\vee+h_j^\wedge-1}, \tag{8.20}$$

so that normalization is inherited from $P_{n_j^\wedge}(0)\,\Gamma_{n_j^\wedge}$ to $P_{n_{j+1}^\wedge}(0)\,\Gamma_{n_{j+1}^\wedge}$, i.e., from $\mathbf{x}_{n_j^\wedge}$ to $\mathbf{x}_{n_{j+1}^\wedge}$, simply by choosing

$$\gamma_{n_{j+1}^\wedge,h_j^\vee+h_j^\wedge-1} := \frac{-1}{\varphi_j^\vee}. \tag{8.21}$$

In contrast to Algorithm 2, only the iterates $\mathbf{z}_{n_j^\wedge}$ are considered as approximants, and only the corresponding vectors $\mathbf{x}_{n_j^\wedge}$ are true residuals. (It is possible to modify Algorithm 2 accordingly, thus avoiding the breakdown due to normalization. The resulting version is, in fact, just the unnormalized Algorithm 3 with scale factors $\Gamma_{n_j^\wedge}$, which yield normalized iterates when $n = n_j^\wedge$.)

ALGORITHM 5 (NORMALIZED NONGENERIC BOBC ALGORITHM FOR LINEAR SYSTEMS: NORMALIZED NONGENERIC BIOMIN). *For solving* $\mathbf{Az} = \mathbf{b}$, *choose an initial approximation* $\mathbf{z}_0$, *set* $\mathbf{u}_0 := \mathbf{x}_0 := \mathbf{b} - \mathbf{Az}_0$, *choose* $\mathbf{v}_0 := \mathbf{y}_0$ *with* $\langle \mathbf{y}_0, \mathbf{x}_0\rangle_{\mathbf{B}} \neq 0$, *and apply Algorithm* 4 *with the special choice* (8.21) *for the relative scale factors* $\gamma_{n_{j+1}^\wedge,h_j^\vee+h_j^\wedge-1}$ (*which determine* $\Gamma_{n_{j+1}^\wedge}$, *while the other scale factors* Γ_n $(n \neq n_{j+1}^\wedge)$, Γ_n', $\bar{\Gamma}_n$, *and* $\bar{\Gamma}_n'$ *may be chosen arbitrarily nonzero*).

Additionally, compute for $j = 0, 1, \ldots$ *the approximant* $\mathbf{z}_{n_{j+1}^\wedge}$ *according to*

$$\begin{aligned}\mathbf{z}_{n_{j+1}^\wedge} = -[&\mathbf{u}_{n_{j+1}^\wedge-1}\gamma_{n_{j+1}^\wedge,1}^\wedge - \mathbf{u}_{n_{j+1}^\wedge-2}\varepsilon_{h_j^\vee-2,j}^\vee\gamma_{n_{j+1}^\wedge,2}^\wedge - \mathbf{u}_{n_{j+1}^\wedge-3}\varepsilon_{h_j^\vee-3,j}^\vee\gamma_{n_{j+1}^\wedge,3}^\wedge \\ &- \cdots - \mathbf{u}_{n_j^\vee}\varepsilon_{0,j}^\vee\gamma_{n_{j+1}^\wedge,h_j^\vee}^\wedge] + \mathbf{z}_{n_j^\wedge}\varphi_j^\vee\gamma_{n_{j+1}^\wedge,h_j^\vee+h_j^\wedge-1}.\end{aligned} \tag{8.22}$$

The algorithm terminates when $n = n_{j+1}^\wedge$ *and* $\mathbf{x}_n = \mathbf{0}$. *Then* $n_{j^\wedge}^\wedge = n$ *and* $\mathbf{z}_n$ *is the solution of* $\mathbf{Az} = \mathbf{b}$. *However, if* $n = n_{j+1}^\wedge = n_{j^\wedge}^\wedge$, *but* $\mathbf{x}_n \neq \mathbf{0}$, *the solution cannot be found using those initial vectors* (*a case of incurable breakdown*).

As an analogy to the generic case [11] and to Algorithm 3 of Part I, we also suggest an *unnormalized* version of the nongeneric BIOMIN algorithm. It not only avoids the danger of breakdown due to normalization (as our nongeneric normalized BIOMIN algorithm does too), but allows to monitor independently the damping effect of the Lanczos polynomials P_n and the often adverse effect of normalization at 0. In this unnormalized version of BIOMIN we can choose all the scale factors Γ_n arbitrarily. We keep track of them by evaluating a recurrence for $\rho_{n_j^\wedge} := \Gamma_{n_j^\wedge}P_n(0)$, which follows from (8.20). Note that $\rho_{n_j^\wedge} \neq 0$ $(\forall j)$ in view of $\varphi_j^\vee \neq 0$ $(\forall j)$. In contrast to our unnormalized nongeneric BIORES algorithm of Part I (Algorithm 3), we restrict ourselves here to this subsequence; thus there is now only a small difference between the normalized and the unnormalized version.

ALGORITHM 6 (UNNORMALIZED NONGENERIC BOBC ALGORITHM FOR LINEAR SYSTEMS: UNNORMALIZED NONGENERIC BIOMIN). *For solving* $\mathbf{Az} = \mathbf{b}$ *choose an*

initial approximation $\mathbf{z}_0$, *set* $\mathbf{u}_0 := \mathbf{x}_0 := \mathbf{b} - \mathbf{A}\mathbf{z}_0$, *choose* $\mathbf{v}_0 := \mathbf{y}_0$ *with* $\langle \mathbf{y}_0, \mathbf{x}_0 \rangle_{\mathbf{B}} \neq 0$, *and apply Algorithm* 4 (*with arbitrary nonzero scale factors* Γ_n, Γ'_n, $\bar{\Gamma}_n$, *and* $\bar{\Gamma}'_n$). *Additionally, compute recursively the vector sequence* $\{\mathbf{z}_{n_j^\wedge}\}$ *according to* (8.22) *and the scalar sequence* $\{\rho_{n_j^\wedge}\}$ *according to*

$$\rho_{n_{j+1}^\wedge} := -\varphi_j^\vee \gamma_{n_{j+1}^\wedge, h_j^\vee + h_j^\wedge - 1} \rho_{n_j^\wedge} \,. \tag{8.23}$$

The algorithm terminates when $n = n_{j+1}^\wedge$ *and* $\mathbf{x}_n = \mathbf{0}$. *Then,* $n_{J^\wedge}^\wedge = n$ *and* $\mathbf{z}_n/\rho_n$ *is the solution of* $\mathbf{A}\mathbf{z} = \mathbf{b}$. *However, if* $n = n_{j+1}^\wedge = n_{J^\wedge}^\wedge$, *but* $\mathbf{x}_n \neq \mathbf{0}$, *the solution cannot be found using those initial vectors* (*a case of incurable breakdown*).

By an induction argument we obtain the following theorem, which is analogous to Theorem 4.6.

THEOREM 8.2. (i) *In Algorithm* 5 (*normalized nongeneric* BIOMIN) *holds*

$$\mathbf{x}_{n_j^\wedge} = \mathbf{b} - \mathbf{A}\mathbf{z}_{n_j^\wedge}, \quad j = 0, 1, 2, \ldots \,. \tag{8.24}$$

(ii) *In Algorithm* 6 (*unnormalized nongeneric* BIOMIN) *holds*

$$\mathbf{x}_{n_j^\wedge} = \mathbf{b}\rho_{n_j^\wedge} - \mathbf{A}\mathbf{z}_{n_j^\wedge}, \quad j = 0, 1, 2, \ldots \,. \tag{8.25}$$

Proof. Assume that (8.25) holds up to a certain j. Using the formulas (8.23), (8.22), and (8.9b) of Algorithm 6 we get

$$\begin{aligned} \mathbf{b}\rho_{n_{j+1}^\wedge} - \mathbf{A}\mathbf{z}_{n_{j+1}^\wedge} = &- \mathbf{b}\varphi_j^\vee \gamma_{n_{j+1}^\wedge, h_j^\vee + h_j^\wedge - 1} \rho_{n_j^\wedge} + \mathbf{A}[\mathbf{u}_{n_{j+1}^\wedge - 1} \gamma^\wedge_{n_{j+1}^\wedge, 1} \\ &- \mathbf{u}_{n_{j+1}^\wedge - 2} \varepsilon^\vee_{h_j^\vee - 2, j} \gamma^\wedge_{n_{j+1}^\wedge, 2} - \mathbf{u}_{n_{j+1}^\wedge - 3} \varepsilon^\vee_{h_j^\vee - 3, j} \gamma^\wedge_{n_{j+1}^\wedge, 3} - \cdots \\ &- \mathbf{u}_{n_j^\vee} \varepsilon^\vee_{0,j} \gamma^\wedge_{n_{j+1}^\wedge, h_j^\vee}] - \mathbf{A}\mathbf{z}_{n_j^\wedge} \varphi_j^\vee \gamma_{n_{j+1}^\wedge, h_j^\vee + h_j^\wedge - 1} = \mathbf{x}_{n_{j+1}^\wedge} \,. \end{aligned} \tag{8.26}$$

Hence, (8.25) follows by induction. In Algorithm 5, (8.21) guarantees that $\rho_{n_j^\wedge}$, defined by (8.23), is 1 for all j, so that (8.24) holds. □

Finally, we want to sketch the nongeneric generalization of yet another important algorithm, namely, of *BIODIR* [11] (or *Lanczos/ORTHODIR* [16]). As in the generic case [11], we first define a "biconjugation algorithm," which is nothing more than the BO algorithm with the inner product $\langle ., . \rangle_{\mathbf{BA}}$ instead of $\langle ., . \rangle_{\mathbf{B}}$. This algorithm can be used to generate the sequences $\{\mathbf{u}_n\}$ and $\{\mathbf{v}_n\}$ alone, without concurrently building up $\{\mathbf{x}_n\}$ and $\{\mathbf{y}_n\}$.

ALGORITHM 7 (NONGENERIC "BICONJUGATION (BC) ALGORITHM"). *Given a bounded linear operator* $\mathbf{A} : \mathcal{H} \to \mathcal{H}$ *and two initial vectors* $\mathbf{u}_0, \mathbf{v}_0 \in \mathcal{H}$ *satisfying* $\langle \mathbf{u}_0, \mathbf{A}\mathbf{v}_0 \rangle_{\mathbf{B}} \neq 0$, *apply the nongeneric* BO *algorithm* (*Algorithm* 1) *with the inner product* $\langle ., . \rangle_{\mathbf{BA}}$ (*instead of* $\langle ., . \rangle_{\mathbf{B}}$) *to produce the two vector sequences* $\{\mathbf{u}_n\}$ *and* $\{\mathbf{v}_n\}$ *with scale factors* Γ'_n *and* $\bar{\Gamma}'_n$ *and the matrix* $\mathbf{H}'$ *containing the recurrence coefficients of the corresponding* FOP1*s. Denote the indices of the regular* FOP1*s by* n'_i, *and let* $h'_i := n'_{i+1} - n'_i$.

In view of their orthogonality properties, the resulting vector sequences are the same as the sequences $\{\mathbf{u}_n\}$, $\{\mathbf{v}_n\}$ generated by the BOBC algorithm (if the initial vectors and the scale factors are the same). By applying half a step of the nongeneric backward qd algorithm (specified by Theorem 7.3), we can find the factors $\mathbf{G}^\wedge$ and $\mathbf{G}^\vee$ of the relevant block UL decomposition of $\mathbf{H}'$. Finally, we can apply formulas

(8.9b) and (8.22) to compute the subsequences $\mathbf{x}_{n_j^\wedge}$ and $\mathbf{z}_{n_j^\wedge}$, and (8.23) to find the appropriate scale factors for the normalization given below.

ALGORITHM 8 (NORMALIZED NONGENERIC BIODIR). *For solving* $\mathbf{Az} = \mathbf{b}$, *choose an initial approximation* $\mathbf{z}_0$, *set* $\mathbf{u}_0 := \mathbf{x}_0 := \mathbf{b} - \mathbf{Az}_0$, *choose* $\mathbf{v}_0 := \mathbf{y}_0$ *with* $\langle \mathbf{y}_0, \mathbf{x}_0 \rangle_{\mathbf{B}} \neq 0$, *and apply Algorithm 7 (with arbitrary nonzero scale factors* Γ'_n *and* $\bar{\Gamma}'_n$*) in order to produce the two vector sequences* $\{\mathbf{u}_n\}$ *and* $\{\mathbf{v}_n\}$ *and the matrix* $\mathbf{H}'$ *of recurrence coefficients of the corresponding* FOP1*s. Concurrently, compute block by block the relevant block* UL *decomposition* $\mathbf{H}' = \mathbf{G}^\wedge \mathbf{G}^\vee$ *according to Theorem* 7.3. *The initial moments required for that are*

$$\phi_0 := \frac{\langle \mathbf{y}_0, \mathbf{x}_0 \rangle_{\mathbf{B}}}{\bar{\Gamma}_0 \Gamma_0}, \quad \phi_{n'_1} := \frac{\langle \mathbf{v}_0, \mathbf{A}\mathbf{u}_{n'_1 - 1} \rangle_{\mathbf{B}}}{\bar{\Gamma}'_0 \Gamma'_{n'_1 - 1}}. \tag{8.27}$$

Additionally, for $j = 0, 1, \ldots$ *compute the subsequences* $\{\mathbf{x}_{n_j}\}$ *and* $\{\mathbf{z}_{n_j}\}$ *according to* (8.9b) *and* (8.22), *the value of* $\gamma_{n_{j+1}^\wedge, h_j^\vee + h_j^\wedge - 1}$ *being given by* (8.21). (*This determines the scale factors* Γ_{n_j}, *while* Γ_n $(n \neq n_j)$ *can be chosen arbitrarily nonzero.*)

The algorithm terminates when $n = n_{j+1}^\wedge$ *and* $\mathbf{x}_n = \mathbf{0}$. *Then* $n_{j^\wedge}^\wedge = n$ *and* $\mathbf{z}_n$ *is the solution of* $\mathbf{Az} = \mathbf{b}$. *If* $n = n_{j+1}^\wedge = n_{j^\wedge}^\wedge$, *but* $\mathbf{x}_n \neq \mathbf{0}$, *the solution cannot be found using those initial vectors* (*a case of incurable breakdown*).

This algorithm is normalized in the same sense as Algorithm 5; the residual polynomials $\Gamma_{n_j} P_{n_j}$ of the approximants $\mathbf{z}_{n_j^\wedge}$ are normalized to 1 at 0. Of course, we could try to replace the backward qd step by the solution of an extra triangular system of equations similar to (8.11c) and (8.11d). But this would require computation of both extra vectors and inner products.

Note that the breakdown conditions for the nongeneric versions of unnormalized BIORES, normalized and unnormalized BIOMIN, and normalized BIODIR are all the same. This is in contrast to the generic versions of these algorithms [11].

9. The treatment of near-breakdown for diagonal sequences. So far we have assumed that we work with exact arithmetic and that, therefore, the regular formal orthogonal polynomials (FOP1s) are well defined, and the corresponding elements of the various vector sequences generated by Lanczos-type algorithms can be computed accurately. Index steps h_j of size greater than 1 between regular FOP1s occur as a consequence of serious, but curable, breakdown. However, in practice exact curable breakdown is very unlikely, but *near-breakdown* may occur as a consequence of either an exact breakdown contaminated by roundoff or a very small $|\delta_j|$. Any such near-breakdown means that the recurrence coefficients $\alpha_{j,i}$ and β_{j+1} are probably large and numerically not well determined; then the subsequent FOP1s and the corresponding Krylov space vectors must be expected to be inaccurate.

Therefore, one must find a way to treat near-breakdowns. The simplest approach would be to use exactly the same formulas as for the exact curable breakdown. This would mean that we proceed implicitly with slightly modified data, but process them in a stable way, instead of treating the original data in an instable way. However, in this section we show that we can do even better. It is possible to treat near-breakdown *exactly* and still fairly efficiently. If our previously defined algorithms are modified accordingly, then, in exact arithmetic, those regular FOP1s that are well conditioned are obtained independently of the threshold used to define near-breakdown. The same is true for the corresponding Krylov space vectors. (Of course, the number of "well-conditioned" regular FOP1s depends on the threshold.) Here we present only

the polynomial formulation of these algorithms. Details of implementation for the corresponding Lanzos-type algorithms are described in joint work with Freund and Nachtigal [8]

In this section we treat ordinary, diagonal sequences of FOP1s. Recall that $\{n_j\}_{j=0}^{J}$ denotes the index sequence of the regular FOP1s P_{n_j} for some functional Φ, and that this sequence is characterized by

$$\Phi(pP_{n_j}) = 0 \quad (\forall p \in \mathcal{P}_{n_{j+1}-2}), \tag{9.1a}$$

$$\Phi(z^{h_j-1}P_{n_j}^2) =: \delta_j \neq 0, \tag{9.1b}$$

where $h_j := n_{j+1} - n_j$, $j = 0, \ldots, J-1$ ($J \leq \infty$). This recursive definition of the sequence $\{n_j\}$ implies that for $j < J$ the diagonal blocks

$$\mathbf{D}_j = \begin{bmatrix} & & & & \delta_j \\ & & & \delta_j & \star \\ & & \iddots & \iddots & \vdots \\ & \delta_j & \iddots & & \star \\ \delta_j & \star & \cdots & \star & \star \end{bmatrix}$$

of the formal Gramian $\mathbf{D}$ in (3.23) and (8.14) are nonsingular. For the intermediate values of n (i.e., for those satisfying $n_j < n < n_{j+1}$ for some j) the FOP1s are not uniquely determined, and we made the particular choice (1.28) for these *inner* FOP1s.

Now, we want to extract an index subsequence $\{\tilde{n}_k\}_{k=0}^{K} \subset \{n_j\}_{j=0}^{J}$ that marks the well-conditioned regular FOP1s $P_{\tilde{n}_k}$. Before we come to its recursive definition we choose first, by analogy to (1.28), *tentative inner* FOP1*s* for this subsequence:

$$\check{p}_n(z) := W_{n-\tilde{n}_k}(z)P_{\tilde{n}_k}(z) \quad \text{if } \tilde{n}_k \leq n < \tilde{n}_{k+1}. \tag{9.2}$$

Here W_m is still a prescribed monic sequence, for example, one satisfying a three-term recurrence (2.10). In the latter case the tentative inner polynomials $\check{p}_n$ themselves are obtained by a three-term recurrence. For simplicity, we could choose $W_m(z) = z^m$, although in practice this is often a rather inappropriate basis. To simplify formulas we include in (9.2) $n = \tilde{n}_k$, where $\check{p}_n = P_{\tilde{n}_k}$ is regular and thus not inner. Actually, when computing the next well-conditioned regular FOP1 $P_{\tilde{n}_{k+1}}$, we also start from a polynomial of the form (9.2), with $n = \tilde{n}_{k+1}$, which is then orthogonalized with respect to the previous blocks. We assume here that for those FOP1s this orthogonalization process has already been carried out, so that (9.2) holds for $\tilde{n}_k \leq n < \tilde{n}_{k+1}$ instead of for $\tilde{n}_k < n \leq \tilde{n}_{k+1}$.

One is tempted to define the index steps $\tilde{h}_k := \tilde{n}_{k+1} - \tilde{n}_k$ by analogy to (9.1) by

$$\Phi(pP_{\tilde{n}_k}) = 0 \quad (\forall p \in \mathcal{P}_{\tilde{n}_k-1}), \tag{9.3a}$$

$$|\Phi(z^i P_{\tilde{n}_k}^2)| \leq \varepsilon \quad (0 \leq i \leq \tilde{h}_k - 2), \tag{9.3b}$$

$$\Phi(z^{\tilde{h}_k-1}P_{\tilde{n}_k}^2) =: \tilde{\delta}_k, \quad |\tilde{\delta}_k| > \varepsilon, \tag{9.3c}$$

where $\varepsilon > 0$ is some prescribed small constant. This would imply that

(9.4)

$$\Phi(\check{p}_m\check{p}_n) = \begin{cases} 0 & \text{if } \tilde{n}_k \leq n \leq \tilde{n}_{k+1} - 2 \text{ and } m+n < 2\tilde{n}_k, \\ O(\varepsilon) & \text{if } \tilde{n}_k \leq n \leq \tilde{n}_{k+1} - 2 \text{ and } 2\tilde{n}_k \leq m+n \\ & \qquad\qquad \leq \tilde{n}_k + \tilde{n}_{k+1} - 2, \\ \tilde{\delta}_k + O(\varepsilon) & \text{if } \tilde{n}_k \leq n \leq \tilde{n}_{k+1} - 1 \text{ and } m+n = \tilde{n}_k + \tilde{n}_{k+1} - 1. \end{cases}$$

For example, if $K = 5$ and $\tilde{n}_0 = 0$, $\tilde{n}_1 = 1$, $\tilde{n}_2 = 2$, $\tilde{n}_3 = 5$, $\tilde{n}_5 = 7$, and if $\tilde{\varepsilon}$, $\tilde{\delta}_k^{\varepsilon}$, and $\star$ are abbreviations for $O(\varepsilon)$, $\tilde{\delta}_k + O(\varepsilon)$, and $O(1)$, respectively, the formal Gramian of $\{\check{p}_n\}$ is then of the form

$$(9.5)\quad [\Phi(\check{p}_m \check{p}_n)]_{m,n=0}^{\infty} = \begin{bmatrix} \tilde{\delta}_0^{\varepsilon} & & & & & & & & & \\ & \tilde{\delta}_1^{\varepsilon} & & & & \tilde{\varepsilon} & & & & \\ & & \tilde{\delta}_2^{\varepsilon} & & \tilde{\varepsilon} & \tilde{\varepsilon} & & & & \\ & & & \tilde{\varepsilon} & \tilde{\varepsilon} & \tilde{\delta}_3^{\varepsilon} & & & & \\ & & \tilde{\varepsilon} & \tilde{\varepsilon} & \tilde{\delta}_3^{\varepsilon} & \star & & & & \\ & \tilde{\varepsilon} & \tilde{\varepsilon} & \tilde{\delta}_3^{\varepsilon} & \star & \star & & \tilde{\varepsilon} & & \\ & & & & & & \tilde{\varepsilon} & \tilde{\delta}_4^{\varepsilon} & & \ddots \\ & & & & & \tilde{\varepsilon} & \tilde{\delta}_4^{\varepsilon} & \star & & \tilde{\varepsilon} & \ddots \\ & & & & & & & \tilde{\varepsilon} & \tilde{\varepsilon} & \ddots \\ & & & & & & & \tilde{\varepsilon} & \tilde{\varepsilon} & \tilde{\varepsilon} & \\ & & & & & & \ddots & \ddots & \ddots & & \ddots \end{bmatrix}.$$

There are two serious problems with this approach: First, while the formal Gramian $\mathbf{D}$ of the FOP1s $\{P_n\}$ is block diagonal, this is obviously no longer true for the one of $\{\check{p}_n\}$. Here, small $O(\varepsilon)$ elements can penetrate into several off-diagonal blocks. As a consequence, it can be seen that the Hessenberg matrix of recurrence coefficients for $\{\check{p}_n\}$ is no longer block tridiagonal; the formula for the regular FOP1 $\check{p}_{\tilde{n}_{k+1}}$ may contain not only terms from the two previous blocks but even terms from older ones. Hence, the major advantage of the generic and the nongeneric Lanczos algorithms is lost here. Second, $\tilde{h}_k$ and thus the sequence $\{\tilde{n}_k\}$ is generally not appropriately defined. Since $|\tilde{\delta}_k|$ need not be much larger than ε, $|\tilde{\delta}_k + O(\varepsilon)|$ may be small or may even vanish, and the diagonal blocks of the formal Gramian may be singular.

The second difficulty can be overcome if we transform the sequence, $\{\check{p}_n\}$, by a block orthogonalization process[5] into a new formally block orthogonal sequence, $\{\tilde{p}_n\}$, whose formal Gramian is block diagonal. The index subsequence $\{\tilde{n}_k\}$ is then defined by requiring that the diagonal blocks

$$(9.6)\qquad \tilde{\mathbf{D}}_k = [\Phi(\tilde{p}_{\tilde{n}_k+i}\, \tilde{p}_{\tilde{n}_k+j})]_{i,j=0}^{\tilde{h}_k-1}$$

of this formal Gramian $\tilde{\mathbf{D}}$ are not only nonsingular, but neither near-singular nor ill conditioned:

$$(9.7)\qquad \sigma_{\min}(\tilde{\mathbf{D}}_k) > \varepsilon, \quad \kappa(\tilde{\mathbf{D}}_k) \leq \kappa'.$$

The appropriate order of magnitude of ε and of the upper bound κ' of the condition number κ depend on the functional Φ. If the latter or the inner polynomials are scaled such that $\sigma_{\max}(\tilde{\mathbf{D}}_k)$ is a priori bounded, it suffices—at least in theory—to make the first check. This is in accordance with the recommendation of Parlett [21]. Note that—also just in theory—the near-breakdown threshold ε need not be very small and the block size chosen need not be the minimal one satisfying (9.7). Unfortunately, in practice, roundoff spoils the orthogonality and the determination of the block size according to (9.7) may not work for large-scale systems. Therefore, a different strategy has to be chosen in the implementation (see [8] for details).

[5] Confer footnote 2 at the end of the introduction.

However, as is seen from (9.5), the block orthogonalization process suggested above also has the disadvantage that it may involve sums that contain elements from several previous blocks, i.e., the upper triangular matrix describing this process is not block bidiagonal, as one might have hoped. In fact, "long" recurrences may occur here not only for some $\tilde{p}_{\tilde{n}_k}$, but also for some inner $\tilde{p}_n$.

Things change if we build up the block orthogonal sequence $\{\tilde{p}_n\}$ directly by orthogonalizing $z\tilde{p}_n$ with respect to previous blocks. In fact, by a standard argument we are going to conclude that the Hessenberg matrix $\tilde{\mathbf{H}}$ of recurrence coefficients is then block tridiagonal. The freedom of choice present in (9.2) is gained from the possibility of adding any linear combination of polynomials that have already been computed in the block under construction. (The coefficients in this linear combination are the same as those in the recurrence (2.10).) Since this linear combination is already orthogonal to all previous blocks, it has no effect on the off-diagonal blocks of $\tilde{\mathbf{H}}$.

Let $\tilde{\mathbf{p}} := [\tilde{p}_0, \tilde{p}_1, \ldots]$ be the infinite row vector whose elements are the resulting monic block orthogonalized polynomials $\tilde{p}_n$. Then the block diagonal formal Gramian of these polynomials can be written as

$$\tilde{\mathbf{D}} := \Phi(\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}) := [\Phi(\tilde{p}_m\tilde{p}_n)]_{m,n=0}^{\infty}\,. \tag{9.8}$$

(Recall that its diagonal blocks satisfy (9.7) by definition of $\{\tilde{n}_k\}$.)

Since the polynomials are monic of ascending degree, they can certainly be generated by a recurrence that has the matrix form

$$z\tilde{\mathbf{p}}(z) = \tilde{\mathbf{p}}(z)\tilde{\mathbf{H}}, \tag{9.9}$$

with $\tilde{\mathbf{H}}$ being unit upper Hessenberg. Consequently,

$$\tilde{\mathbf{D}}\tilde{\mathbf{H}} = \Phi(\tilde{\mathbf{p}}^T\tilde{\mathbf{p}})\tilde{\mathbf{H}} = \Phi(\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}\tilde{\mathbf{H}}) = \Phi(\tilde{\mathbf{p}}^T z\tilde{\mathbf{p}}) \tag{9.10a}$$

$$= \Phi(\tilde{\mathbf{H}}^T\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}) = \tilde{\mathbf{H}}^T\Phi(\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}) = \tilde{\mathbf{H}}^T\tilde{\mathbf{D}}\,. \tag{9.10b}$$

Since $\tilde{\mathbf{H}}$ is upper Hessenberg and $\tilde{\mathbf{D}}$ is block diagonal, the left-hand side is upper block Hessenberg and the right-hand side is lower block Hessenberg. Consequently, both sides are block tridiagonal, and multiplication by $\tilde{\mathbf{D}}^{-1}$ shows that the same is true for $\tilde{\mathbf{H}}$ itself, i.e., by analogy to (3.6),

$$\tilde{\mathbf{H}} := \begin{bmatrix} \tilde{\mathbf{A}}_0 & \tilde{\mathbf{B}}_1 & & & \\ \tilde{\mathbf{C}}_0 & \tilde{\mathbf{A}}_1 & \tilde{\mathbf{B}}_2 & & \\ & \tilde{\mathbf{C}}_1 & \tilde{\mathbf{A}}_2 & \ddots & \\ & & \ddots & \ddots & (\tilde{\mathbf{B}}_K) \\ & & & (\tilde{\mathbf{C}}_{K-1}) & (\tilde{\mathbf{A}}_K) \end{bmatrix}. \tag{9.11}$$

In general the block structure is coarser here than in (3.6), but all block boundaries in this matrix are also present in the matrix $\mathbf{H}$ of (3.6). In view of $\tilde{\mathbf{H}}$ being unit upper Hessenberg, the blocks $\tilde{\mathbf{C}}_k$ have again just a 1 in the upper right corner, and the blocks $\tilde{\mathbf{A}}_k$ are either 1×1 or unit upper Hessenberg. It remains to investigate the upper triangular part of $\tilde{\mathbf{A}}_k$ and the superdiagonal blocks $\tilde{\mathbf{B}}_k$. Moreover, we have to discuss how to determine the elements of these blocks.

The only condition we are imposing is that $\{\tilde{p}_n\}$ be formally block orthogonal, i.e., $\Phi(\tilde{\mathbf{p}}^T\tilde{\mathbf{p}}) = \tilde{\mathbf{D}}$ is block diagonal. Splitting the row vector $\tilde{\mathbf{p}}$ into blocks of size $\tilde{h}_k$, we set $\tilde{\mathbf{p}} = [\tilde{\mathbf{p}}_0, \tilde{\mathbf{p}}_1, \ldots, (\tilde{\mathbf{p}}_K)]$ to get

$$\tilde{\mathbf{D}}_k = \Phi(\tilde{\mathbf{p}}_k^T\tilde{\mathbf{p}}_k)\,. \tag{9.12a}$$

From the kth block column of (9.10a) we extract the three conditions

$$\tilde{\mathbf{D}}_{k-1}\tilde{\mathbf{B}}_k = \Phi(\tilde{\mathbf{p}}_{k-1}^T z\tilde{\mathbf{p}}_k), \tag{9.12b}$$

$$\tilde{\mathbf{D}}_k\tilde{\mathbf{A}}_k = \Phi(\tilde{\mathbf{p}}_k^T z\tilde{\mathbf{p}}_k), \tag{9.12c}$$

$$\tilde{\mathbf{D}}_{k+1}\tilde{\mathbf{C}}_k = \Phi(\tilde{\mathbf{p}}_{k+1}^T z\tilde{\mathbf{p}}_k)\,. \tag{9.12d}$$

If $\tilde{\mathbf{p}}_{k-1}$, $\tilde{p}_{\tilde{n}_k}$, and $\tilde{\mathbf{D}}_{k-1}$ are known, (9.12a)–(9.12c) allow us, if they are used column by column and in parallel, to build up $\tilde{\mathbf{p}}_k$, $\tilde{\mathbf{B}}_k$, $\tilde{\mathbf{A}}_k$, and $\tilde{\mathbf{D}}_k$, and to determine $\tilde{h}_{k+1}$ and $\tilde{p}_{\tilde{n}_{k+1}}$. Actually, except for its last column, $\tilde{\mathbf{A}}_k$ can be chosen as an arbitrary unit upper Hessenberg matrix, hence we assume it given, except for the last column. For example, each of the other columns may be zero except for a 1 on the subdiagonal.

First, since $\Phi(p\tilde{\mathbf{p}}_k) = \mathbf{0}^T$ for all $p \in \mathcal{P}_{\tilde{n}_k-1}$, only the last line of (9.12b) is nonzero:

$$\tilde{\mathbf{D}}_{k-1}\tilde{\mathbf{B}}_k = \tilde{\mathbf{l}}_k\boldsymbol{\phi}_k^T, \tag{9.13a}$$

where

$$\boldsymbol{\phi}_k^T := \Phi(z\tilde{p}_{\tilde{n}_k-1}\tilde{\mathbf{p}}_k), \quad \tilde{\mathbf{l}}_k := [0,\ldots,0,1]^T \in \mathbb{C}^{\tilde{h}_k}\,. \tag{9.13b}$$

Hence, $\tilde{\mathbf{B}}_k$ has rank 1. Once $\tilde{p}_n$ is known for some n with $(\tilde{n}_k \le n \le \tilde{n}_{k+1} - 1)$, column n of $\tilde{\mathbf{B}}_k$ is obtained by solving a linear system with the coefficient matrix $\tilde{\mathbf{D}}_{k-1}$, which is no longer triangular, but has constant nonzero antidiagonal elements and a small upper left triangular part.[6] If $n < \tilde{n}_{k+1} - 1$, the corresponding column of $\tilde{\mathbf{A}}_k$ is prescribed, and thus $\tilde{p}_{n+1}$ can be computed according to (9.9). Moreover, the element $n+1$ in the first row (i.e., row $\tilde{n}_k + 1$) of $\tilde{\mathbf{D}}_k$ can be evaluated explicitly using the definition (9.12a) of $\tilde{\mathbf{D}}_k$. Once $n+1 = \tilde{n}_{k+1} - 1$, the whole first column of $\tilde{\mathbf{D}}_k$ is known, as is its first row, thanks to symmetry.

After splitting off the first row and column of $\tilde{\mathbf{D}}_k$ and the first row and the last column of $\tilde{\mathbf{A}}_k$, (9.12c) is seen to yield a set of $\tilde{h}_k - 1$ triangular systems for computing the yet unknown elements of $\tilde{\mathbf{D}}_k$. Then, by (9.12b), too, the last column $\tilde{\mathbf{a}}_k$ of $\tilde{\mathbf{A}}_k$ is also found by solving a linear system with coefficient matrix $\tilde{\mathbf{D}}_k$:

$$\tilde{\mathbf{D}}_k\tilde{\mathbf{a}}_k = \Phi(\tilde{\mathbf{p}}_k^T z\tilde{p}_{\tilde{n}_{k+1}-1})\,. \tag{9.14}$$

Finally, now that the last columns of $\tilde{\mathbf{A}}_k$ and $\tilde{\mathbf{B}}_k$ are known, the recurrence for the next well-conditioned regular FOP1, $\tilde{p}_{\tilde{n}_{k+1}}$ is ready.

Due to the special structure of $\tilde{\mathbf{C}}_k$, the only part of $\tilde{\mathbf{D}}_k$ that matters in (9.12d) is the first column. It is easy to see that for the elements in this first column (9.12d) provides formulas that are mathematically equivalent to those from the definition (9.12a) of $\tilde{\mathbf{D}}_k$.

Summarizing, we see that on the basis of the relations (9.12) we can build up $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{D}}$ column by column.

One may wonder what can be said about the order of magnitude of the elements of $\tilde{\mathbf{D}}_k$ and $\tilde{\mathbf{B}}_k$ if (9.3) holds for some $\varepsilon << 1$ and with $|\tilde{\delta}'_k| >> 1$ $(k' = k-1, k)$. In

[6] Column n of $\tilde{\mathbf{B}}_k$ refers to the part of column n of $\tilde{\mathbf{H}}$ that lies in $\tilde{\mathbf{B}}_k$; rows and elements are referred to in an analogous fashion.

the notation of (9.5) we clearly have, for $0 \le k < K$,

$$\tilde{\mathbf{D}}_k = \begin{bmatrix} \tilde{\varepsilon} & \tilde{\varepsilon} & \cdots & \tilde{\varepsilon} & \tilde{\delta}_k^{\varepsilon} \\ \tilde{\varepsilon} & & & \iddots & \star \\ \vdots & & \iddots & & \vdots \\ \tilde{\varepsilon} & \iddots & & & \star \\ \tilde{\delta}_k^{\varepsilon} & \star & \cdots & \star & \star \end{bmatrix}. \tag{9.15}$$

(When $K < \infty$, the last diagonal block $\tilde{\mathbf{D}}_K$ is infinite and consists of elements of order $O(\varepsilon)$.) From (9.15) it is easy to conclude that

$$\tilde{\mathbf{D}}_k^{-1} = \begin{bmatrix} \star & \star & \cdots & \star \\ \star & & & \tilde{\varepsilon} \\ \vdots & & \iddots & \vdots \\ \star & \tilde{\varepsilon} & \cdots & \tilde{\varepsilon} \end{bmatrix}, \tag{9.16}$$

and, therefore, that

$$\tilde{\mathbf{B}}_k = \tilde{\mathbf{D}}_{k-1}^{-1} \tilde{\mathbf{l}}_k \boldsymbol{\phi}_k^T = \begin{bmatrix} \tilde{\varepsilon} & \cdots & \tilde{\varepsilon} & \star \\ \tilde{\varepsilon}^2 & \cdots & \tilde{\varepsilon}^2 & \tilde{\varepsilon} \\ \vdots & & \vdots & \vdots \\ \tilde{\varepsilon}^2 & \cdots & \tilde{\varepsilon}^2 & \tilde{\varepsilon} \end{bmatrix}, \tag{9.17}$$

where $\tilde{\varepsilon}^2 := O(\varepsilon^2)$.

Actually, the order-of-magnitude statement in (9.17) can be seen to be a consequence of the block diagonality of $\tilde{\mathbf{D}}$, the block tridiagonality of $\tilde{\mathbf{H}}$, and the special structure of the blocks $\tilde{\mathbf{D}}_k$ and $\tilde{\mathbf{C}}_k$. Comparing superdiagonal blocks in the relation (9.10), we get $\tilde{\mathbf{D}}_{k-1}\tilde{\mathbf{B}}_k = \tilde{\mathbf{C}}_{k-1}^T \tilde{\mathbf{D}}_k$; hence

$$\tilde{\mathbf{B}}_k = \tilde{\mathbf{D}}_{k-1}^{-1} \tilde{\mathbf{C}}_{k-1}^T \tilde{\mathbf{D}}_k = \begin{bmatrix} \star & \star & \cdots & \star \\ \star & & & \tilde{\varepsilon} \\ \vdots & & \iddots & \vdots \\ \star & \tilde{\varepsilon} & \cdots & \tilde{\varepsilon} \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ \tilde{\varepsilon} & \cdots & \tilde{\varepsilon} & \star \end{bmatrix} = \begin{bmatrix} \tilde{\varepsilon} & \cdots & \tilde{\varepsilon} & \star \\ \tilde{\varepsilon}^2 & \cdots & \tilde{\varepsilon}^2 & \tilde{\varepsilon} \\ \vdots & & \vdots & \vdots \\ \tilde{\varepsilon}^2 & \cdots & \tilde{\varepsilon}^2 & \tilde{\varepsilon} \end{bmatrix}. \tag{9.18}$$

Comparing (9.13), (9.17), and (9.18), we get the additional relation

$$\boldsymbol{\phi}_k^T := \Phi(z\tilde{p}_{\tilde{n}_k-1}\tilde{\mathbf{p}}_k) = \tilde{\mathbf{l}}_k \boldsymbol{\phi}_k^T = \tilde{\mathbf{C}}_{k-1}^T \tilde{\mathbf{D}}_k = \Phi(\tilde{p}_{\tilde{n}_k}\tilde{\mathbf{p}}_k), \tag{9.19}$$

which is, indeed, a consequence of $\tilde{p}_n$ ($n \ge \tilde{n}_k$) being orthogonal to $\mathcal{P}_{\tilde{n}_k-1}$.

10. The treatment of near-breakdown for staircase sequences. In this section we modify the ideas of the previous section in order to compute a block staircase sequence of well-conditioned FOP1s instead of a block diagonal sequence.

The aim is to determine two full sequences $\tilde{\mathbf{p}} := \{\tilde{p}_n\}_{n=0}^{\infty}$, $\tilde{\mathbf{p}}' := \{\tilde{p}'_n\}_{n=0}^{\infty}$ of monic polynomials ($\tilde{p}_n$ and $\tilde{p}'_n$ being of degree n) that are block orthogonal, well conditioned, and block compatible in the following sense. There are two (finite or infinite) index sequences $\{\tilde{n}_k^{\wedge}\}_{k=0}^{K^{\vee}}$, $\{\tilde{n}_k^{\vee}\}_{k=0}^{K^{\wedge}}$ (with $K^{\vee} \le K^{\wedge} \le K^{\vee}+1 \le \infty$) satisfying

$$\tilde{n}_k^{\wedge} \le \tilde{n}_k^{\vee} \ (k = 0, \ldots, K^{\vee}), \quad \tilde{n}_k^{\vee} < \tilde{n}_{k+1}^{\wedge} \ (k = 0, \ldots, K^{\wedge} - 1) \tag{10.1}$$

such that the following two conditions hold: First

$$\Phi(\tilde{p}_m \tilde{p}_n) = 0 \text{ if } m < \tilde{n}_k^{\wedge} \le n \text{ for some } k \quad \text{or } m \le \tilde{n}_k^{\vee} < n \text{ for some } k, \tag{10.2a}$$

$$\Phi'(\tilde{p}'_m \tilde{p}'_n) = 0 \text{ if } m < \tilde{n}_k^{\vee} \le n \text{ for some } k \quad \text{or } m < \tilde{n}_{k+1}^{\wedge} \le n \text{ for some } k, \tag{10.2b}$$

which means that the two formal Gramians

$$\tilde{\mathbf{D}}^{\wedge} := \Phi(\tilde{\mathbf{p}}^T \tilde{\mathbf{p}}), \quad \tilde{\mathbf{D}}^{\vee} := \Phi'((\tilde{\mathbf{p}}')^T \tilde{\mathbf{p}}') \tag{10.3}$$

are block diagonal, with blocks starting at the columns $\tilde{n}_k^{\wedge}$, $\tilde{n}_k^{\vee} + 1$ ($\le \tilde{n}_{k+1}^{\wedge}$) and $\tilde{n}_k^{\vee}$, $\tilde{n}_{k+1}^{\wedge}$ ($\le \tilde{n}_{k+1}^{\vee}$), respectively. Second, for their diagonal blocks

$$\tilde{\mathbf{D}}_k^{\wedge} := [\Phi(\tilde{p}_m \tilde{p}_n)]_{m,n=\tilde{n}_k^{\wedge}}^{\tilde{n}_k^{\vee}}, \quad \tilde{\mathbf{D}}_{k+\frac{1}{2}}^{\wedge} := [\Phi(\tilde{p}_m \tilde{p}_n)]_{m,n=\tilde{n}_k^{\vee}+1}^{\tilde{n}_{k+1}^{\wedge}-1}, \tag{10.4a}$$

$$\tilde{\mathbf{D}}_k^{\vee} := [\Phi'(\tilde{p}'_m \tilde{p}'_n)]_{m,n=\tilde{n}_k^{\vee}}^{\tilde{n}_{k+1}^{\wedge}-1}, \quad \tilde{\mathbf{D}}_{k-\frac{1}{2}}^{\vee} := [\Phi'(\tilde{p}'_m \tilde{p}'_n)]_{m,n=\tilde{n}_k^{\wedge}}^{\tilde{n}_k^{\vee}-1}, \tag{10.4b}$$

holds, by analogy to (9.7), (for some $\varepsilon > 0$, $\kappa' > 1$):

$$\sigma_{\min}(\tilde{\mathbf{D}}_k^{\wedge}) > \varepsilon, \quad \kappa(\tilde{\mathbf{D}}_k^{\wedge}) \le \kappa', \tag{10.5a}$$

$$\sigma_{\min}(\tilde{\mathbf{D}}_{k+\frac{1}{2}}^{\wedge}) > \varepsilon, \quad \kappa(\tilde{\mathbf{D}}_{k+\frac{1}{2}}^{\wedge}) \le \kappa', \tag{10.5b}$$

$$\sigma_{\min}(\tilde{\mathbf{D}}_k^{\vee}) > \varepsilon, \quad \kappa(\tilde{\mathbf{D}}_k^{\vee}) \le \kappa', \tag{10.5c}$$

$$\sigma_{\min}(\tilde{\mathbf{D}}_{k-\frac{1}{2}}^{\vee}) > \varepsilon, \quad \kappa(\tilde{\mathbf{D}}_{k-\frac{1}{2}}^{\vee}) \le \kappa'. \tag{10.5d}$$

Note that some of the blocks $\tilde{\mathbf{D}}_{k+(1/2)}^{\wedge}$ and $\tilde{\mathbf{D}}_{k-(1/2)}^{\vee}$ may be void. They exist only if the respective index step

$$\tilde{h}_k^{\vee} := \tilde{n}_{k+1}^{\wedge} - \tilde{n}_k^{\vee}, \quad \tilde{h}_k^{\wedge} := \tilde{n}_k^{\vee} - \tilde{n}_k^{\wedge} + 1 \tag{10.6}$$

is larger than 1. Then $\tilde{\mathbf{D}}_{k+(1/2)}^{\wedge}$ lies in $\tilde{\mathbf{D}}^{\wedge}$ between $\tilde{\mathbf{D}}_k^{\wedge}$ and $\tilde{\mathbf{D}}_{k+1}^{\wedge}$, and $\tilde{\mathbf{D}}_{k-(1/2)}^{\vee}$ lies in $\tilde{\mathbf{D}}^{\vee}$ between $\tilde{\mathbf{D}}_{k-1}^{\vee}$ and $\tilde{\mathbf{D}}_k^{\vee}$. The blocks in (10.4a) have order $\tilde{h}_k^{\wedge}$ and $\tilde{h}_k^{\vee} - 1$; those in (10.4b) have order $\tilde{h}_k^{\vee}$ and $\tilde{h}_k^{\wedge} - 1$, respectively.

The recursive process for constructing these two sequences, like the one in §5, alternates between the two sequences. The block structure generated in this way in the corresponding analog of the Padé table (which is a "near-FOP1 table") still resembles the one of Fig. 3, but at this point it is restricted to the two diagonals specified by l and $l + 1$. A conflict arises because we require the four conditions (10.5a)–(10.5d), although (10.5a) and (10.5c) would be enough to determine the two index sequences $\{\tilde{n}_k^{\wedge}\}$ and $\{\tilde{n}_k^{\vee}\}$. But it is important that $\tilde{p}'_{\tilde{n}_k^{\vee}}$ lies in the first row of a block, and $\tilde{p}_{\tilde{n}_{k+1}^{\wedge}}$ in the first column, as happens automatically in §5, cf. Fig. 3. This must now be enforced by a condition guaranteeing that the blocks $\tilde{\mathbf{D}}_{k+(1/2)}^{\wedge}$ and $\tilde{\mathbf{D}}_{k-(1/2)}^{\vee}$ are nonsingular. In order that $\tilde{p}'_{\tilde{n}_k^{\vee}}$ and $\tilde{p}_{\tilde{n}_{k+1}^{\wedge}}$ be well-conditioned regular FOP1s, these blocks can be neither near-singular nor ill conditioned. Consequently, on each diagonal the sizes of some blocks may turn out to be larger than when we apply the algorithm of the previous section to the respective diagonal. But this is the price we have to pay for a "compatible" block structure on the two diagonals, i.e., index sequences $\{\tilde{n}_k^{\wedge}\}, \{\tilde{n}_k^{\vee}\}$, which together define the blocks on both diagonals.

Now we come to the details of the algorithm. Its basic pattern is as follows:

(i) When $n = \tilde{n}_k^\wedge$, Φ-orthogonalize $z\tilde{p}'_{n-1}$ with respect to $\mathcal{P}_{\tilde{n}_k^\wedge - 1}$ to get $\tilde{p}_n$; check (10.5a) to determine whether $\tilde{\mathbf{D}}_k^\wedge$ is just 1×1. If yes, set $\tilde{n}_k^\vee := n$ and proceed with (iii).

(ii) If not, i.e., if $\tilde{n}_k^\vee > \tilde{n}_k^\wedge$, then, for $n = \tilde{n}_k^\wedge + 1, \tilde{n}_k^\wedge + 2, \ldots$, Φ'-orthogonalize $\tilde{p}_{n-1}$ with respect to $\mathcal{P}_{\tilde{n}_k^\wedge - 1}$ to get $\tilde{p}'_{n-1}$, and set $\tilde{p}_n := z\tilde{p}'_{n-1}$; check (10.5a) and (10.5d) to determine whether $\tilde{\mathbf{D}}_k^\wedge$ and $\tilde{\mathbf{D}}_{k-(1/2)}^\vee$ are completed, in which case $\tilde{n}_k^\vee := n$.

(iii) When $n = \tilde{n}_k^\vee$, Φ'-orthogonalize $\tilde{p}_n$ with respect to $\mathcal{P}_{\tilde{n}_k^\vee - 1}$ to get $\tilde{p}'_n$; check (10.5c) to determine whether $\tilde{\mathbf{D}}_k^\vee$ is just 1×1. If yes, set $\tilde{n}_{k+1}^\wedge := n$ and proceed with (i).

(iv) If not, i.e., if $\tilde{n}_{k+1}^\wedge > \tilde{n}_k^\vee + 1$, then, for $n = \tilde{n}_k^\vee + 1, \tilde{n}_k^\vee + 2, \ldots$, Φ-orthogonalize $z\tilde{p}'_{n-1}$ with respect to $\mathcal{P}_{\tilde{n}_k^\vee}$ to get $\tilde{p}_n$ and set $\tilde{p}'_n := \tilde{p}_n$; check (10.5c) and (10.5b) to determine whether $\tilde{\mathbf{D}}_k^\vee$ and $\tilde{\mathbf{D}}_{k-(1/2)}^\wedge$ are completed, in which case $\tilde{n}_{n+1}^\wedge := n + 1$.

Note that the choice $\tilde{p}_n := z\tilde{p}'_{n-1}$ in (ii) implies that $\tilde{p}_n$ is Φ-orthogonal to $\mathcal{P}_{\tilde{n}_k^\wedge - 1}$; analogously, the choice $\tilde{p}'_n := \tilde{p}_n$ in (iv) implies that $\tilde{p}'_n$ is Φ'-orthogonal to $\mathcal{P}_{\tilde{n}_k^\vee - 1}$.

By analogy to the generality introduced in §9, one can modify $\tilde{p}_{n-1}$ and $z\tilde{p}'_{n-1}$ (in (ii)), and $z\tilde{p}'_{n-1}$ and $\tilde{p}_n$ (in (iv)), by adding a linear combination of polynomials (of the other type) that have already been found in the respective substep.

Since the polynomials $\tilde{p}_n$ (and, likewise, $\tilde{p}'_n$) are not orthogonal to each other within the blocks, the orthogonalization procedures called in the algorithm have to make use of the inverses $(\tilde{\mathbf{D}}_k^\wedge)^{-1}$, $(\tilde{\mathbf{D}}_{k+(1/2)}^\wedge)^{-1}$, (and $(\tilde{\mathbf{D}}_{k-(1/2)}^\vee)^{-1}$, $(\tilde{\mathbf{D}}_k^\vee)^{-1}$, respectively).

As is seen from this recipe, another complication that arises is that the analog of (5.28) does not hold. We still have

$$\Phi(p\tilde{p}_n) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\wedge - 1}) \Rightarrow \Phi'(p\tilde{p}_n) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\wedge - 2}), \tag{10.7a}$$

$$\Phi'(p\tilde{p}'_{n-1}) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\vee - 1}) \Rightarrow \Phi(pz\tilde{p}'_{n-1}) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\vee - 1}), \tag{10.7b}$$

$$\Phi(p\tilde{p}_n) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\vee}) \Rightarrow \Phi'(p\tilde{p}_n) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\vee - 1}), \tag{10.7c}$$

$$\Phi'(p\tilde{p}'_{n-1}) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\wedge - 1}) \Rightarrow \Phi(pz\tilde{p}'_{n-1}) = 0 \ (\forall p \in \mathcal{P}_{\tilde{n}_k^\wedge - 1}), \tag{10.7d}$$

and we know that the left-hand sides are true when $\tilde{n}_k^\wedge \le n$ in (a), $\tilde{n}_k^\vee < n$ in (b) and (c), and $\tilde{n}_k^\wedge < n$ in (d), respectively; but, in general,

$$\Phi'(z^{\tilde{n}_k^\wedge - 1}\tilde{p}_n) \neq 0 \quad \text{and} \quad \Phi(z^{\tilde{n}_k^\vee} z\tilde{p}'_{n-1}) \neq 0\,. \tag{10.8}$$

Therefore, $\tilde{p}_n$ needs to be Φ'-orthogonalized with respect to $z^{\tilde{n}_k^\wedge - 1}$ or $\tilde{p}'_{\tilde{n}_k^\wedge - 1}$ to get $\tilde{p}'_n$; likewise $z\tilde{p}'_{n-1}$ needs to be Φ-orthogonalized with respect to $z^{\tilde{n}_k^\vee}$ or $\tilde{p}_{\tilde{n}_k^\vee}$ to get $\tilde{p}_n$. In the recurrence formulas for $\tilde{p}'_n$ (or $\tilde{p}_n$) the polynomials $\tilde{p}'_m$ (or $\tilde{p}_m$, respectively) of the previous block appear, but not those from older blocks.

There are other, theoretically equivalent formulations for the algorithm. Our version, which is truly sequential, suggests an implementation by recurrence formulas that have the matrix form

$$\tilde{\mathbf{p}}(z) = \tilde{\mathbf{p}}'(z)\tilde{\mathbf{G}}^\wedge, \quad z\tilde{\mathbf{p}}'(z) = \tilde{\mathbf{p}}(z)\tilde{\mathbf{G}}^\vee, \tag{10.9}$$

as in (6.15), with $\tilde{\mathbf{G}}^\wedge$ unit upper triangular and $\tilde{\mathbf{G}}^\vee$ unit upper Hessenberg. As in (7.1), elimination of $\tilde{\mathbf{p}}'$ or $\tilde{\mathbf{p}}$, respectively, leads to

$$z\tilde{\mathbf{p}}(z) = \tilde{\mathbf{p}}(z)\tilde{\mathbf{H}}^\wedge, \ \text{where} \ \tilde{\mathbf{H}}^\wedge := \tilde{\mathbf{G}}^\vee\tilde{\mathbf{G}}^\wedge, \tag{10.10a}$$

and

$$z\tilde{\mathbf{p}}'(z) = \tilde{\mathbf{p}}'(z)\tilde{\mathbf{H}}^{\vee}, \quad \text{where } \tilde{\mathbf{H}}^{\vee} := \tilde{\mathbf{G}}^{\wedge}\tilde{\mathbf{G}}^{\vee}. \tag{10.10b}$$

Each of these equalities is of the same type as (9.9). But as we mentioned above, if we apply the algorithm of §9 to Φ, the block structure of $\tilde{\mathbf{H}}$ in (9.9) may turn out to be finer than the one of $\tilde{\mathbf{H}}^{\wedge}$ in (10.10a), because here we have the four conditions (10.5) instead of the condition (9.7) determining the block sizes in (9.9). But we have also pointed out that there is no need to choose in §9 the minimum block sizes satisfying (9.7), and since each of the pairs (10.5a)–(10.5b) and (10.5c)–(10.5d) is equivalent to (9.7) (applied to the two different functionals Φ and Φ'), each of (10.10a) and (10.10b) is, indeed, identical to a case of (9.9). From §9 it follows in particular that $\tilde{\mathbf{H}}^{\wedge}$ and $\tilde{\mathbf{H}}^{\vee}$ are block tridiagonal unit Hessenberg matrices. Therefore, $\tilde{\mathbf{G}}^{\wedge}$ and $\tilde{\mathbf{G}}^{\vee}$ are block upper bidiagonal and block lower bidiagonal, respectively:

(10.11)

$$\tilde{\mathbf{G}}^{\vee} := \begin{bmatrix} \tilde{\mathbf{G}}_0^{\vee} & & & & \\ \tilde{\mathbf{L}}_0^{\vee} & \tilde{\mathbf{G}}_1^{\vee} & & & \\ & \tilde{\mathbf{L}}_1^{\vee} & \tilde{\mathbf{G}}_2^{\vee} & & \\ & & \ddots & \ddots & \\ & & & \tilde{\mathbf{L}}_{J^{\wedge}-1}^{\vee} & \tilde{\mathbf{G}}_{J^{\wedge}}^{\vee} \end{bmatrix}, \quad \tilde{\mathbf{G}}^{\wedge} := \begin{bmatrix} \tilde{\mathbf{G}}_0^{\wedge} & \tilde{\mathbf{F}}_1^{\wedge} & & & \\ & \tilde{\mathbf{G}}_1^{\wedge} & \tilde{\mathbf{F}}_2^{\wedge} & & \\ & & \tilde{\mathbf{G}}_2^{\wedge} & \ddots & \\ & & & \ddots & \tilde{\mathbf{F}}_{J^{\wedge}}^{\wedge} \\ & & & & \tilde{\mathbf{G}}_{J^{\wedge}}^{\wedge} \end{bmatrix},$$

analogous to (6.16). From (10.9) and our description of the algorithm, it is clear that the blocks $\tilde{\mathbf{L}}_k^{\vee}$ have again just a 1 in their upper right corner, that the blocks $\tilde{\mathbf{G}}_k^{\vee}$ are itself unit upper Hessenberg, and that the blocks $\tilde{\mathbf{G}}_k^{\wedge}$ are unit upper triangular. Further results on the structure of the blocks can be obtained by mixing the approach of §6 with the one of §9, but the details become somewhat tedious.

Conclusions. In the two parts of this paper we have reviewed the theory of formal orthogonal polynomials (FOPs) of the first and second kind (FOP1s and FOP2s, respectively), and we have derived old and new recurrences for recursively constructing certain sequences of such FOPs. From the beginning we dealt with the so-called nonnormal or nongeneric case, where some of the FOPs are not regular and the associated Padé table has singular blocks. This led to "nongeneric" algorithms for constructing such FOPs. By translating these nongeneric algorithms into algorithms for sequences of vectors in Krylov space, we found nongeneric versions of the Lanczos-type algorithms BIORES, BIOMIN, and BIODIR. They can overcome most exact breakdowns of the previously known standard versions of these algorithms. (The exception is the so-called incurable breakdown.)

Finally, in §§9 and 10, we addressed the near-breakdown and presented algorithms for generating sequences of FOP1s in a stable way. By translating these "look-ahead" algorithms for constructing FOP1s into algorithms for sequences of vectors in Krylov space, we readily find stable look-ahead versions of BIORES, BIOMIN, and BIODIR.

Since the FOP1s are denominators of Padé approximants, and since the numerators of Padé approximants satisfy the same recurrences, with different initial conditions, the algorithms given in this work can also be considered as algorithms for generating sequences of Padé approximants on either a diagonal or a generalized staircase of the Padé table. The look-ahead versions can be expected to be stable, in contrast to other algorithms [2], [4]–[6], [10] that have been proposed for nonnormal Padé tables and the related partial realization problem of systems theory. Applications to the fast solution of Hankel systems are also foreseeable.

Acknowledgment. The author would like to thank Susan Göldi for typing the first version of this manuscript and to Noël Nachtigal for making a number of linguistic corrections.

REFERENCES

[1] D. L. Boley, S. Elhay, G. H. Golub, and M. H. Gutknecht, *Nonsymmetric Lanczos and finding orthogonal polynomials associated with indefinite weights*, Numer. Algorithms, 1 (1991), pp. 21–43.

[2] A. Bultheel, *Recursive algorithms for nonnormal Padé tables*, SIAM J. Appl. Math., 39 (1980), pp. 106–118.

[3] ———, *Laurent Series and Their Padé Approximations*, Birkhäuser, Basel/Boston, 1987.

[4] A. Bultheel and M. V. Barel, *Padé techniques for model reduction in linear system theory: A survey*, J. Comput. Appl. Math., 14 (1986), pp. 401–438.

[5] G. Claessens and L. Wuytack, *On the computation of non-normal Padé approximants*, J. Comput. Appl. Math., 5 (1979), pp. 283–289.

[6] A. Draux, *Polynômes orthogonaux formels—applications*, Lecture Notes in Mathematics, Vol. 974, Springer-Verlag, Berlin, 1983.

[7] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, G. A. Watson, ed., Lecture Notes in Mathematics, Vol. 506, Springer-Verlag, Berlin, 1976, pp. 73–89.

[8] R. Freund, M. Gutknecht, and N. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.

[9] G. Golub and M. Gutknecht, *Modified moments for indefinite weight functions*, Numer. Math., 57 (1990), pp. 607–624.

[10] W. B. Gragg and A. Lindquist, *On the partial realization problem*, Linear Algebra Appl., 50 (1983), pp. 277–319.

[11] M. H. Gutknecht, *The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions, and the qd algorithm*, Proceedings of the Copper Mountain Conference on Iterative Methods (preliminary version).

[12] ———, *Continued fractions associated with the Newton–Padé table*, Numer. Math., 56 (1989), pp. 547–589.

[13] ———, *A completed theory of the unsymmetric Lanczos process and related algorithms, Part* I, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 594–639.

[14] C. J. Hegedüs, *Generating Conjugate Directions for Arbitrary Matrices by Matrix Equations* I.–II., Tech. Report KFKI-1990-36/M, Central Research Institute for Physics, Hungarian Academy of Sciences, Budapest, Hungary, 1990.

[15] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, Dover, New York, 1964.

[16] K. C. Jea and D. M. Young, *On the simplification of generalized conjugate-gradient methods for nonsymmetrizable linear systems*, Linear Algebra Appl., 52 (1983), pp. 399–417.

[17] W. D. Joubert, *Lanczos methods for the solution of nonsymmetric systems of linear equations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 926–943.

[18] ———, *Generalized Conjugate Gradient and Lanczos Methods for the Solution of Nonsymmetric Systems of Linear Equations*, Ph.D. thesis, Center for Numerical Analysis, University of Texas, Austin, TX, 1990; Tech. Report CNA-238.

[19] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bureau Standards, 45 (1950), pp. 255–281.

[20] ———, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bureau Standards, 49 (1952), pp. 33–53.

[21] B. N. Parlett, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 567–593.

[22] B. N. Parlett, D. R. Taylor, and Z. A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp., 44 (1985), pp. 105–124.

[23] H. Rutishauser, *Der Quotienten-Differenzen-Algorithmus*, Mitt. Inst. Angew. Math. ETH, Nr.7, Birkhäuser, Basel, Switzerland, 1957.

[24] D. R. Taylor, *Analysis of the Look Ahead Lanczos Algorithm*, Ph.D. thesis, Dept. of Mathematics, University of California, Berkeley, CA, 1982.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 59–61, January 1994

003

A BOTTOM-UP INDUCTIVE PROOF OF THE SINGULAR VALUE DECOMPOSITION*

C.-T. PAN† AND KERMIT SIGMON‡

Abstract. The singular value decomposition (SVD) has a long history. The first proofs of the SVD for real square matrices came out of the study of bilinear forms, first by Beltrami in 1873 and, independently, by Jordan in 1874. Beltrami recognized and used the relationship of the SVD to the eigenvalue decomposition of the matrices A^TA and AA^T, while Jordan used an inductive argument that constructs the SVD from the largest singular value and its associated singular vectors. Many proofs of the SVD in modern references are still based on one of these methods. The purpose of this note is to give a new simple "bottom-up" inductive proof of the SVD, starting from the smallest singular value, which is essentially different from either of these methods.

Key words. singular value decomposition

AMS subject classifications. 15A18, 15A23

The singular value decomposition (SVD) has a long history, a detailed survey of which is given in [6, pp. 134–144]. The first proofs of the SVD for real square matrices came out of the study of bilinear forms, first by Beltrami in 1873 [2] and, independently, by Jordan in 1874 [7]. Beltrami recognized and used the relationship of the SVD to the matrices A^TA and AA^T. Jordan used an inductive argument that constructs the SVD from the largest singular value and its associated singular vectors. The first proof of the SVD for square complex matrices seems to be by Autonne in 1915 [1] and, later in 1939, Eckart and Young [3] who dealt with the rectangular complex case. Many proofs of the SVD in modern references either rely on the eigenvalue decomposition of the positive semidefinite Hermitian matrices A^*A and AA^* [5], [8]–[10] or use a "top-down" inductive argument similar to Jordan's [4], [5, p. 427].

The purpose of this note is to give a new simple "bottom-up" inductive proof of the SVD, starting from the smallest singular value. One should note that this proof is essentially different from the "top-down" one; there does not appear to be a direct dual to the "top-down" proof. The QR decomposition is needed in our proof. The proof is motivated by ideas from a paper by Stewart [11]. Our proof also shows that, if the estimates of the smallest singular value and its associated right singular vector were exact in each step, Stewart's URV decomposition [11] renders the exact SVD.

In this paper, bold lowercase letters denote the column vectors, and $\|\cdot\|_2$ is either the Euclidean norm of a vector or the spectral norm of a matrix. The ith column of an identity matrix is denoted by $\mathbf{e}_i$.

LEMMA 1. *For a square nonsingular complex matrix A one has*

$$\min_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2 = \frac{1}{\|A^{-1}\|_2}.$$

* Received by the editors March 23, 1992; accepted for publication (in revised form) April 21, 1992. This research was supported in part by the Institute for Mathematics and Its Applications with funds provided by the National Science Foundation.

† Department of Mathematical Sciences, Northern Illinois University, DeKalb, Illinois 60115 (pan@math.niu.edu).

‡ Department of Mathematics, University of Florida, Gainesville, Florida 32611 (sigmon@math.ufl.edu).

Proof. Since A is nonsingular, it is easy to see that

$$\|A^{-1}\|_2 = \max_{\mathbf{x}\neq 0} \frac{\|A^{-1}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \max_{\mathbf{y}\neq 0} \frac{\|\mathbf{y}\|_2}{\|A\mathbf{y}\|_2} = \frac{1}{\min\limits_{\mathbf{y}\neq 0} \dfrac{\|A\mathbf{y}\|_2}{\|\mathbf{y}\|_2}} = \frac{1}{\min\limits_{\|\mathbf{y}\|_2=1} \|A\mathbf{y}\|_2}. \qquad \square$$

LEMMA 2. *If R is an $m \times n$ upper triangular matrix with $m \geq n$, then*

$$|r_{ii}| \geq \min_{\|\mathbf{x}\|_2=1} \|R\mathbf{x}\|_2$$

for each diagonal entry r_{ii} of R.

Proof. If R is rank deficient, the stated bound is immediate since $R\mathbf{x} = 0$ for some $\|\mathbf{x}\|_2 = 1$, so we may assume that R has full column rank.

First note that for any entry a_{ij} of a complex matrix A,

$$|a_{ij}| = |\mathbf{e}_i^* A \mathbf{e}_j| \leq \|\mathbf{e}_i\|_2 \|A\mathbf{e}_j\|_2 \leq \|\mathbf{e}_i\|_2 \|A\|_2 \|\mathbf{e}_j\|_2 = \|A\|_2.$$

The matrix R has the form $[R_1 \quad 0]$ with R_1 invertible and $(R_1^{-1})_{ii} = 1/r_{ii}$. Thus $\|R_1^{-1}\|_2 \geq |(R_1^{-1})_{ii}| = |1/r_{ii}|$, from which, by Lemma 1, the result follows since

$$\min_{\|\mathbf{x}\|_2=1} \|R\mathbf{x}\|_2 = \min_{\|\mathbf{x}\|_2=1} \|R_1\mathbf{x}\|_2 = \frac{1}{\|R_1^{-1}\|_2} \leq |r_{ii}|. \qquad \square$$

THEOREM 1. *Each matrix $A \in \mathbf{C}^{m\times n}$ has an* SVD. *That is, there exist unitary matrices*

$$U \in \mathbf{C}^{m\times m} \quad \textit{and} \quad V \in \mathbf{C}^{n\times n}$$

such that

$$U^* A V = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbf{C}^{m\times n}, \qquad p = \min\{m, n\},$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

If $A \in \mathbf{R}^{m\times n}$, then U and V may be taken to be real orthogonal matrices.

Proof. Assume that $m \geq n$ (otherwise, consider A^*). Let $\mathbf{x}_0$ be a unit vector such that $\|A\mathbf{x}_0\|_2 = \min_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$, and set $\sigma := \|A\mathbf{x}_0\|_2$. Let $V_1 \in \mathbf{C}^{n\times n}$ be a unitary matrix whose last column is $\mathbf{x}_0$ and let $U_1 \in \mathbf{C}^{m\times m}$ be a unitary matrix such that $U_1^*(AV_1) = R =: [r_{ij}]$ is upper triangular (here we use the QR decomposition).

Since

$$\|R\mathbf{e}_n\|_2 = \|U_1^* A V_1 \mathbf{e}_n\|_2 = \|U_1^* A \mathbf{x}_0\|_2 = \|A\mathbf{x}_0\|_2 = \sigma,$$

from Lemma 2 we have

$$|r_{1n}|^2 + \cdots + |r_{nn}|^2 = \sigma^2 \leq |r_{nn}|^2,$$

and hence $|r_{1n}|^2 + \cdots + |r_{n-1,n}|^2 = 0$. It follows that $|r_{nn}| = \sigma$ so that $r_{nn} = \sigma e^{it}$ for some t. Now define the diagonal unitary matrix $W_1 := \operatorname{diag}(1, \ldots, 1, e^{it}) \in \mathbf{C}^{n\times n}$ and observe that

$$A = U_1 R (V_1 W_1)^* \quad \text{with } R = \begin{bmatrix} R_1 & 0 \\ 0 & \sigma \\ 0 & 0 \end{bmatrix} \in \mathbf{C}^{m\times n}.$$

Since $R_1 \in \mathbf{C}^{(n-1)\times(n-1)}$ and $V_1 W_1$ is unitary, a straightforward inductive argument proves that there exist unitary matrices $U \in \mathbf{C}^{m\times m}$ and $V \in \mathbf{C}^{n\times n}$ such that

$$A = U \begin{bmatrix} \sigma_1 & & & 0 \\ & \ddots & & \\ & & & \sigma_n \\ - & - & - & - \\ & 0 & & \end{bmatrix} V^*. \qquad \square$$

REFERENCES

[1] L. AUTONNE, *Sur les matrices hypohermitiennes et sur les matrices unitaires*, Ann. Univ. Lyon Nouvelle Série I, Fasc., 38 (1915), pp. 1–77.

[2] E. BELTRAMI, *Sulle funzioni bilineari*, Giornale de Matematiche, 11 (1873), pp. 98–106.

[3] C. ECKART AND G. YOUNG, *A principal axis transformation for non–hermitian matrices*, Bull. Amer. Math. Soc., 45 (1939), pp. 118–121.

[4] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[5] R. A. HORN AND C. A. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[6] ———, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[7] C. JORDAN, *Mémoire sur les formes bilinéaires*, J. de Math. Pures Appl., Deuxième Série, 19 (1874), pp. 35–54.

[8] P. LANCASTER AND M. TISMENETSKI, *The Theory of Matrices*, Academic Press, New York, 1985.

[9] B. NOBLE AND J. W. DANIEL, *Applied Linear Algebra*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1988.

[10] G. W. STEWART, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[11] ———, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Processing, 40 (1992), pp. 1535–1541.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 62–79, January 1994

PREDICTING STRUCTURE IN SPARSE MATRIX COMPUTATIONS*

JOHN R. GILBERT†

Abstract. Many sparse matrix algorithms—for example, solving a sparse system of linear equations—begin by predicting the nonzero structure of the output of a matrix computation from the nonzero structure of its input. This paper is a catalog of ways to predict nonzero structure. It contains known results for some problems, including various matrix factorizations, and new results for other problems, including some eigenvector computations.

Key words. sparse matrix algorithms, graph theory, matrix factorization, systems of linear equations, eigenvectors

AMS subject classifications. 15A18, 15A23, 65F50, 68R10

1. Introduction.

A sparse matrix algorithm is an algorithm that performs a matrix computation in such a way as to take advantage of the zero/nonzero structure of the matrices involved. Usually this means not explicitly storing or manipulating some or all of the zero elements; sometimes sparsity can also be exploited to work on different parts of a matrix problem in parallel. Large sparse matrix computations arise in structural design, geodetics, fluid dynamics, heat transport, semiconductor modeling, circuit analysis, molecular dynamics, geophysical reservoir analysis, and many other areas. It is common for problems to be so large that they could not be solved at all without sparse techniques.

Many sparse matrix algorithms [6], [15]–[17], [21] have a phase that predicts the nonzero structure of the solution from the nonzero structure of the problem, followed by a phase that does the numerical computation in a static data structure. This saves space, because the space used by the pointers in a dynamic data structure during the first phase can be reused by the numeric values in the second phase. Also, in many applications a sequence of problems with the same nonzero structure must be solved, and the structural phase can be done just once. The structural phase may also be used to schedule the numerical phase efficiently on a parallel machine [20], [29].

Structure prediction can be used to save time as well as space in sparse Gaussian elimination. The asymptotically fastest algorithms used to compute the Cholesky factorization of a symmetric positive definite matrix are those of the Yale Sparse Matrix Package [12] and Sparspak [15], which predict the structure of the triangular factor by a version of Theorem 4.3. Gilbert and Peierls [24] have used prediction of the structure of the solution of a triangular system of equations to develop the first algorithm that performs sparse LU factorization with partial pivoting in worst-case time proportional to the number of real arithmetic operations. (This method of prediction is a special case of Theorem 5.1.)

Graph theory is a useful language in which to state and prove the results of structure prediction. One reason for this is that the structural effect of a matrix computation often depends on path structure, which is easier to describe in terms of graphs than in terms of matrices. Parter [33] was among the first to use graph theory

* Received by the editors July 30, 1987; accepted for publication (in revised form) April 16, 1992.
† Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304 (gilbert@parc.xerox.com). This work was done while the author was visiting the University of Iceland and the Institute for Mathematics and Its Applications at the University of Minnesota. This research was partially supported by the National Science Foundation grant DCR-8451385.

as a tool to investigate sparse matrix computation; Fiedler [13] also pioneered many of these ideas.

This paper is a catalog of the effects of several common matrix computations on nonzero structure. It includes arithmetic, linear systems, various factorizations, and some eigenvector problems. Where appropriate, it cites algorithms to compute nonzero structure as well as theorems that describe it. Some of these results are not new. The known results are scattered among papers on various topics in linear algebra and algorithms that have been published in journals on numerical analysis, theoretical computer science, operations research, and engineering. Here they are presented in a common framework, together with a few new results.

Three or four different graph models are used for structure prediction. Undirected graphs model symmetric matrices; directed graphs model unsymmetric matrices under symmetric permutations; bipartite graphs model arbitrary rectangular matrices; and column intersection graphs can sometimes be used to apply undirected graph results to rectangular matrices. This paper describes results using all the models, but concentrates most heavily on those that use directed graphs to model unsymmetric matrices with nonzero diagonal elements.

Sections 3–6 contain the results of the paper: Roughly speaking, the results in §3 are immediate; those in §4 are known; most of those in §5 are consequences of known results; and those in §6 are new. This paper is based on an earlier technical report [19].

2. Definitions. We assume that the reader is familiar with such basic graph-theoretic terms as directed graph, undirected graph, and path. Harary [26] is a good general reference.

2.1. Directed graphs and matrix structures. Let A be an $n \times n$ matrix. The *structure* of A is its directed graph

$$\text{struct}(A) = G(A),$$

whose vertices are the integers $1, \ldots, n$ and whose edges are

$$\{ (i,j) : i \neq j \text{ and } A_{ij} \neq 0 \}.$$

When no ambiguity can arise, we shall sometimes not distinguish between a matrix, its graph, and the set of edges of its graph.

The graph $G(A)$ does not specify whether the diagonal elements of A are zero or not. In this paper we will use $G(A)$ only for matrices with nonzero diagonal elements; in the context of structure prediction, matrices with zero diagonal elements are more usefully studied by means of their bipartite graphs, as described below.

Applying the same permutation to the columns and rows of A corresponds to renumbering the vertices of the directed graph of A. In other words, if P is a permutation matrix, then the directed graph of PAP^T is isomorphic to the directed graph of A. In general, if P and Q are different permutation matrices, then the directed graph of PAQ^T is not isomorphic to that of A.

The *structure* of a vector x is

$$\text{struct}(x) = \{ i : x_i \neq 0 \},$$

which can be interpreted as a set of vertices of the directed graph of any $n \times n$ matrix.

We will use the notation $G_1 \subseteq G_2$ to mean that graph G_1 is a subgraph of graph G_2; that is, that both the edge and vertex sets of G_1 are subsets of those of G_2.

2.2. Predicting structure in a computation. To say more precisely what we mean by the structural effect of a computation, we make some remarks based on those of Brayton, Gustavson, and Willoughby [3] and Edenbrandt [10]. Let f be a function from one or more matrices or vectors to a matrix or vector. The structure of A may not determine the structure of $f(A)$; for example, in general the sum of two full vectors is full, but $(1,1)^T + (1,-1)^T$ is not full. We wish to ignore zeros created by coincidence in the numerical values of A and determine the smallest structure that is "big enough" for the result of f with any input of the given structure. That is, given f and $\mathrm{struct}(A)$, we want to determine

$$\bigcup_B \{\mathrm{struct}(f(B)) : \mathrm{struct}(B) \subseteq \mathrm{struct}(A)\}.$$

Brayton, Gustavson, and Willoughby called an algorithm "s-minimal" if it computes this structure from $\mathrm{struct}(A)$. We sometimes call this a "one-at-a-time" structure prediction, since each position in the predicted structure can be made nonzero, but there is no guarantee that all can be made nonzero at the same time.

Most of the functions we consider in this paper have the property that for each input structure S, there is a worst-case value A with $\mathrm{struct}(A) = S$ such that $\mathrm{struct}(B) \subseteq S$ implies $\mathrm{struct}(f(B)) \subseteq \mathrm{struct}(f(A))$. In other words, each input structure corresponds to a unique maximal output structure. We sometimes call such an analysis an "all-at-once" structure prediction.

A function for which there is no "all-at-once" structure prediction is $f(A) = U$, the upper triangular factor of A in Gaussian elimination with partial pivoting. Suppose the structure of A is

$$\begin{pmatrix} \times & & \\ \times & \times & \\ \times & & \times \end{pmatrix}.$$

Depending on the relative magnitudes of the elements in the first column, the structure of $f(A)$ may be

$$\begin{pmatrix} \times & & \\ & \times & \\ & & \times \end{pmatrix}, \quad \begin{pmatrix} \times & \times & \\ & \times & \times \\ & & \times \end{pmatrix}, \quad \begin{pmatrix} \times & \times & \\ & \times & \\ & & \times \end{pmatrix}, \quad \text{or} \quad \begin{pmatrix} \times & & \times \\ & \times & \times \\ & & \times \end{pmatrix}.$$

The smallest structure big enough for $f(A)$ is a full upper triangular matrix, even though $f(A)$ cannot be full.

2.3. Graph terminology. Representing a matrix structure as a graph has the advantage that it is easy to describe properties that depend on paths in the graph. Here we define several graph notions that depend on path structure.

The notation $i \xrightarrow{A} j$ means that there is an edge from i to j in $G(A)$; that is, that $A_{ij} \neq 0$. The notation $i \overset{A}{\Longrightarrow} j$ means that there is a directed path from i to j in $G(A)$. Such a path may have length zero; that is, $i \overset{A}{\Longrightarrow} i$ always holds. The matrix A may be omitted if it is clear from context.

Now let $G(A)$ be a directed graph, and let x be a subset of the vertices of G. We say x is *closed* (with respect to A) if there is no edge of G from a vertex not in x to a vertex in x; that is, if $x_j \neq 0$ and $A_{ij} \neq 0$ imply $x_i \neq 0$. The *closure* of x (with respect to A) is the smallest closed set containing x,

$$\mathrm{closure}(x) = \bigcap \{ y : x \subseteq y \text{ and } y \text{ is closed}\},$$

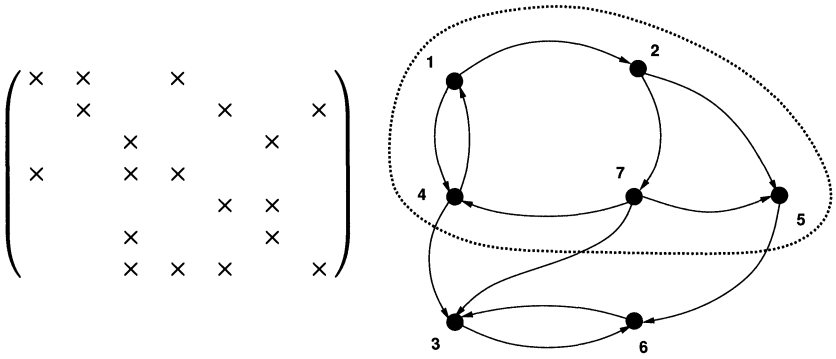


FIG. 1. *A matrix and its directed graph.* closure($\{5\}$) *is outlined.*

which is the set of vertices of A from which there are paths to vertices of x. Figure 1 shows a matrix and its graph, with closure($\{5\}$) outlined.

The *transitive closure* of A is the graph $G^*(A)$ whose edges correspond to paths in A. Thus

$$i \overset{G^*(A)}{\longrightarrow} j \quad \text{if and only if} \quad i \neq j \quad \text{and} \quad i \overset{A}{\Longrightarrow} j.$$

The structural effect of Gaussian elimination can be described in terms of a subgraph of $G^*(A)$ called the *filled graph* of A, which we write as $G^+(A)$. This graph has edges corresponding to paths whose highest-numbered vertices are their endpoints:

$$i \overset{G^+(A)}{\longrightarrow} j \quad \text{if and only if} \quad i \neq j \quad \text{and} \quad i \overset{A}{\Longrightarrow} j \quad \text{through vertices less than } \min(i,j).$$

Figure 2 is an example. Notice that the filled graph depends on the numbering of the vertices of A, whereas the transitive closure and the closure of a vertex are preserved under renumbering (that is, under graph isomorphism).

Remark 2.1. Rose [35] introduced the notation $G^*(A)$ for the filled graph of A, but that notation is also widely used for transitive closure. Since we want to refer to both transitive closures and filled graphs, we use $G^+(A)$ for the "smaller" of the two.

A graph G is *strongly connected* if there is a path from every vertex to every other vertex, or, equivalently, if G^* is a complete directed graph. A square matrix A is called *irreducible* if struct(A) is strongly connected. Clearly, for any permutation matrix P, PAP^T is irreducible if and only if A is. The *strongly connected components* (or just *strong components*) of a graph G are its maximal strongly connected subgraphs. Every vertex of a graph is in exactly one strong component, and every edge is in at most one strong component. If a square matrix A is permuted into block triangular form with as many diagonal blocks as possible, the diagonal blocks partition the rows and columns of A into sets corresponding to the strong components of struct(A). Figure 3 shows the strong components of the graph in Fig. 1, with the vertices renumbered so that the matrix has a diagonal block for each strong component.

A square matrix A is called *fully indecomposable* if struct(PAQ^T) is strongly connected for all permutation matrices P and Q. Clearly, this is equivalent to struct(PA)

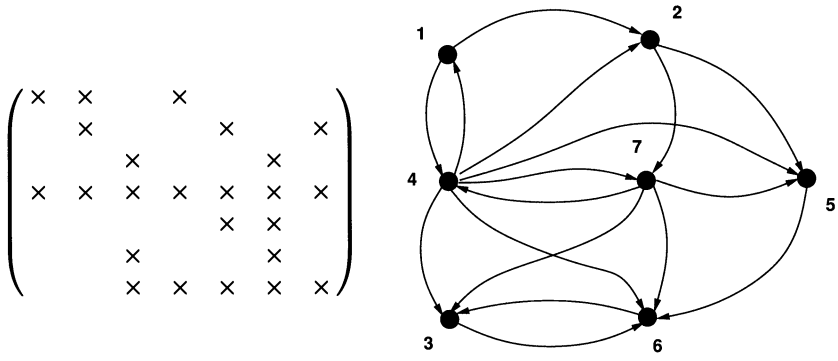


FIG. 2. *Filled graph of example from Fig.* 1.

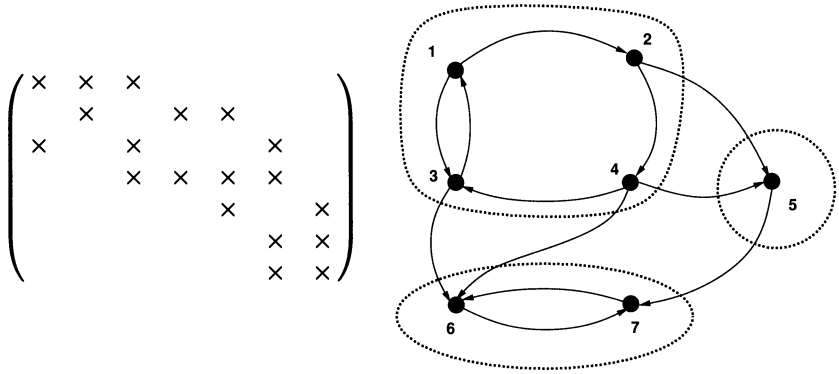


FIG. 3. *Matrix from Fig.* 1 *permuted to block triangular form. Strong components are outlined.*

being strongly connected for all permutations P. As we will remark in the next section, fully indecomposable matrices are the same as square strong Hall matrices.

If matrix A is symmetric, its directed graph contains edge (i, j) if and only if it contains edge (j, i). Informally, we shall not distinguish between this graph and the *undirected graph* of A, which has an undirected edge $\{i, j\}$ if $A_{ij} \neq 0$. Chordal graphs are undirected graphs that are useful for describing symmetric Gaussian elimination. An undirected graph is *chordal* if every cycle of length at least 4 has a *chord*; that is, if for every cycle $v_1, v_2, \ldots, v_k, v_1$ with $k \geq 4$ there is some edge $\{v_i, v_j\}$ for which $i \neq j \pm 1 \pmod{k}$.

2.4. Nonsquare matrices. This paper concentrates on structure prediction results that use directed graphs and that (for the most part) apply to square matrices with nonzero diagonals. For completeness, however, we include some results on matrices that need not be square, or that may have zero diagonal elements. Two other graph models are appropriate in that case.

If A has m rows and n columns, then $A^T A$ is an $n \times n$ symmetric matrix, whose

diagonal is nonzero if A has no zero columns. Its structure is related to the *column intersection graph* of A, which is the undirected graph $G_\cap(A)$ whose vertices are the integers $1, \ldots, n$ (corresponding to the columns of A), and whose edges are

$$\{\{i,j\} : i \neq j \text{ and } \exists k \text{ with } A_{ki} \neq 0 \text{ and } A_{kj} \neq 0\}.$$

Thus $G_\cap(A)$ has an edge between any pair of columns that share a nonzero row. This implies that

$$G(A^TA) \subseteq G_\cap(A),$$

with equality unless there is numerical cancellation in A^TA. Permuting the rows of A does not change the column intersection graph: if P is a row permutation matrix, then $G_\cap(PA)$ is the same as $G_\cap(A)$.

If A has m rows and n columns, the *bipartite graph* of A is the undirected graph $H(A)$ whose vertices are $1', 2', \ldots, m'$ and $1, 2, \ldots, n$, and whose edges are $\{\{i', j\} : A_{ij} \neq 0\}$. The superscript prime notation is intended to indicate that the row and column vertices of $H(A)$ are chosen from two different copies of the positive integers. Permuting the rows and columns of A only relabels the vertices of the bipartite graph: if P and Q are row and column permutation matrices, then $H(PAQ^T)$ is isomorphic to $H(A)$.

Several structure prediction problems use matchings and alternating paths in the bipartite graph of a matrix [4], [6], [7], [20], [23], [21], [32]. This paper does not consider such problems in detail, but we include enough definitions here to state some of these results in later sections.

Let A be an $m \times n$ matrix with $m \geq n$. We say that A has the *Hall property* if, for every k with $0 \leq k \leq n$, every set of k columns of A contains nonzeros in at least k rows. (That is, every set of k column vertices of $H(A)$ is adjacent to at least k row vertices.) We say that A has the *strong Hall property* if, for every k with $0 < k < n$, every set of k columns of A contains nonzeros in at least $k+1$ rows. The graph $H(A)$ has a matching that covers all of its columns if and only if A has the Hall property. A square matrix A with nonzero diagonal is irreducible if and only if $H(A)$ has the strong Hall property; an arbitrary square matrix A is fully indecomposable if and only if $H(A)$ has the strong Hall property. See Lovasz and Plummer [30] for background on bipartite matching. Our terminology is from Coleman, Edenbrandt, and Gilbert [4].

Incidentally, although permuting the rows and the columns of A independently can change the directed graph of A, it does not change the row partition and the column partition induced by the strong components of the graph of A. Another way to say this is that, given a matrix A, if we first permute rows and columns (asymmetrically) to make the diagonal elements nonzero, and then permute rows and columns symmetrically to block triangular form with as many diagonal blocks as possible, then regardless of the initial choice of nonzero diagonal we always get the same block triangular form, up to possible permutation of the diagonal blocks and reordering of the rows and columns within each block.

2.5. Other definitions. We will call a finite set $\{x_1, \ldots, x_n\}$ of complex numbers *algebraically independent* if the point $(x_1, \ldots, x_n)$ is not a zero of any nonzero n-variable polynomial with integer coefficients. Then x_i is transcendental over the field $\mathbf{Q}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ of the rationals extended by all the x's except x_i. There exist arbitrarily large algebraically independent sets, even of real numbers, by a simple countability argument.

3. Products. The following simple result is used in the proof of Theorem 6.1. The structure is the bipartite graph of the matrix in question.

THEOREM 3.1. *Let the structures of an $m \times n$ matrix A and an n-vector x be given.*

(i) *Whatever values A and x have,* struct(Ax) *is a subset of the row vertices of $H(A)$ adjacent to column vertices whose indices are in* struct(x).

(ii) *There exits values for the nonzeros of A and x such that* struct(Ax) *is equal to the set of row vertices described above.* □

The generalization to products of matrices is immediate, since each column of AB is A times a column of B. Theorem 3.1 also implies that $G(A^TA) \subseteq G_\cap(A)$.

4. Factorizations. In this section we describe the structural effect of several matrix factorizations. The necessary definitions are in §2.

4.1. LU factorization. For the factorization $A = LU$, where L is lower triangular with unit diagonal and U is upper triangular, we consider square matrices A with nonzero diagonal, and the graph in question is the directed graph of A. The square matrix $L + U - I$ represents the entire factorization. (Not all nonsingular matrices have LU factorizations without pivoting. In a later subsection we consider factorization with partial pivoting.)

THEOREM 4.1 (see [36]). *Let a structure $G(A)$ be given, with nonzero diagonal elements.*

(i) *If values are chosen for which A has an LU factorization as above, then $G(L+U-I) \subseteq G^+(A)$.*

(ii) *Values for the nonzeros of A exist with $G(L+U-I) = G^+(A)$.* □

Rose and Tarjan [36] gave an algorithm for computing $G^+(A)$ from A in $O(nm)$ time, where A is $n \times n$ with m nonzeros. They also showed that $G^*(A)$ can be computed in time asymptotically the same as that to compute $G^+(A)$, so a faster algorithm to compute $G^+(A)$ would give a faster algorithm to compute transitive closures than the best currently known. By using various transitively reduced graphs, Eisenstat, Gilbert, and Liu [11], [22] give algorithms to compute $G^+(A)$ that are more efficient in practice than transitive closure.

Remark 4.2. A nonsingular square matrix may have an LU factorization even though it has zeros on the diagonal. In this case, Theorem 4.1(i) still holds; but the converse, part (ii), is false. Brayton, Gustavson, and Willoughby [3] gave a counterexample. Let

$$\text{struct}(A) = \begin{pmatrix} \times & \times & \times & \\ \times & & & \\ \times & \times & \times & \\ \times & & & \times \end{pmatrix}.$$

The $(4,3)$ entry in $G^+(A)$ is nonzero, but $L_{4,3} = 0$ regardless of the nonzero values of A.

4.2. Cholesky factorization. Here we consider the factorization $A = LL^T$, where A is a symmetric, positive definite matrix, and L is lower triangular with positive diagonal. Then A has a nonzero diagonal because it is positive definite, and the directed graph of A corresponds to an undirected graph because A is symmetric.

THEOREM 4.3 (see [37]). *Let a symmetric structure $G(A)$ be given, with nonzero diagonal elements.*

(i) *No matter what values A has, if A has a Cholesky factorization $A = LL^T$ then $G(L) \subseteq G^+(A)$.*

(ii) *There exist symmetric values for the nonzeros of A such that $G(L + L^T) = G^+(A)$.* □

Rose, Tarjan, and Lueker [37] gave an $O(n + f)$ algorithm to compute $G^+(A)$ for a symmetric $n \times n$ matrix A with fill of size f. Another such algorithm is implemented in several standard sparse matrix computation packages [12], [15].

Rose showed that the graphs of Cholesky factors of symmetric matrices are exactly the chordal graphs; or, equivalently, that a structure can be reordered to have no fill if and only if it is chordal.

THEOREM 4.4 (see [37]). *Let a symmetric structure $G(A)$ be given, with nonzero diagonal elements.*

(i) *$G^+(A)$ is a chordal graph.*

(ii) *Conversely, if $G(A)$ is a chordal graph, then there is a permutation matrix P such that $G^+(PAP^T) = G(PAP^T)$.* □

Rose, Tarjan, and Lueker [37] and Tarjan and Yannakakis [39] gave linear-time algorithms to determine whether a $G(A)$ is chordal and, if so, to reorder its vertices so that $G^+(PAP^T) = G(PAP^T)$. Such a reordering is called a "perfect elimination order."

4.3. Partial pivoting. The example in §2 showed that a result of the form of Theorem 4.3 is not possible for LU factorization with partial pivoting, because no single choice of nonzero values for A is guaranteed to produce nonzeros in all possible positions of U. George and Ng [16] gave an upper bound. A few remarks are necessary to understand the bound.

There are two ways to write the LU factorization, with partial pivoting, of a square matrix A. One is as $A = PLU$, where L is unit lower triangular, U is upper triangular, and P is a permutation matrix. The other is as $A = P_1L_1P_2L_2 \ldots P_{n-1}L_{n-1}U$, where P_i is a permutation that just transposes row i and a higher-numbered row, and L_i is a Gauss transform (a unit lower triangular matrix with nonzeros only in column i). To get the first factorization, use the standard outer product form of Gaussian elimination to replace A by its triangular factors, pivoting by interchanging two rows of the matrix at the beginning of each major step; at major step k, each row thus interchanged contains entries of L in the first $k - 1$ positions and entries of the partially factored A in the remaining positions. To get the second factorization, pivot by interchanging, at the beginning of major step k, only columns k through n of the two rows in question. In this case an entry of the lower triangle is never moved once it is computed, and only the rows of the partially factored matrix are interchanged.

The factorizations are equivalent in the sense that the same arithmetic is performed in each case, the two U's are the same, and the values of the nonzeros in $L - I$ and $\hat{L} = \sum_{1 \le i < n}(L_i - I)$ are the same; only the positions of the nonzeros in the lower triangular factors are different. The George–Ng theorem describes the structures of $\hat{L}$ and U, saying that both of them are subsets of the structure of the symbolic Cholesky factor of A^TA, that is, of the filled graph of the column intersection graph of A.

THEOREM 4.5 (see [16]). *Let a structure $G(A)$ be given. Whatever values A has, if Gaussian elimination with partial pivoting gives the factors $\hat{L}$ and U as above, then $G(\hat{L} + U) \subseteq G_\cap^+(A)$.* □

The example in §2 showed that the converse of this theorem is not true. Various partial converses hold, however. If we restrict ourselves to structures that are strong Hall (fully indecomposable), which includes irreducible structures with nonzero diago-

nals, then there is a "one-at-a-time" converse to Theorem 4.5 for the upper triangular factor U:

THEOREM 4.6 (see [20]). *Let a square strong Hall structure $H(A)$ be given. For any choice of $i < j$ with $i \overset{G_\cap^+(A)}{\longrightarrow} j$, there is a choice of values for A for which the factorization with partial pivoting $PA = LU$ makes $U_{ij} \neq 0$.* □

The "all-at-once" version of this statement is not true even for irreducible matrices. For example, take $G(A)$ to be a tridiagonal matrix plus a full first column. Then $G(A)$ is irreducible (and strong Hall), and it is easy to see that $G_\cap^+(A)$ is full. As Theorem 4.6 states, it is possible to choose values for A to make any single position in U nonzero; however, the first row of U will always be some row of A, so the entire first row of U cannot be nonzero at the same time.

For the lower triangular factor L, Theorem 4.5 is not as tight as possible, even in the "one-at-a-time" sense for strong Hall structures. For example, let $G(A)$ be tridiagonal (and hence strong Hall). Then $G_\cap^+(A)$ is five-diagonal, predicting that $\hat{L}$ could be lower tridiagonal; but, in fact, $\hat{L}$ must be lower bidiagonal. George and Ng [17] suggest a way of predicting the structures of $\hat{L}$ and U by efficiently simulating all possible pivoting steps. Gilbert and Ng [23] have recently shown that this method (which we do not describe here in detail) gives a tight "one-at-a-time" prediction of the structures of both L and U in the strong Hall case.

4.4. QR factorization. Suppose A is an $m \times n$ matrix with $m \leq n$. Here we consider the factorization $A = QR$, where Q is an orthogonal matrix and R is upper triangular with nonnegative diagonal. George and Heath observed that, since this R is the same as the Cholesky factor of A^TA, the structure of R can be predicted by forming $G_\cap(A)$ and doing structural Cholesky factorization.

THEOREM 4.7 (see [14]). *Let the structure $H(A)$ be given for a rectangular matrix A with at least as many rows as columns. Whatever values A has, if A has full column rank, then its orthogonal factorization $A = QR$ satisfies $G(R) \subseteq G_\cap^+(A)$.* □

The converse of this theorem is false; for example, if A is upper triangular with a nonzero diagonal and a full first row, then $G_\cap(A) = G_\cap^+(A)$ is full, but the orthogonal factor R is equal to A. Coleman, Edenbrandt, and Gilbert supplied a converse in the strong Hall case.

THEOREM 4.8 (see [4]). *Let a structure $H(A)$ be given with at least as many rows as columns. If $H(A)$ has the strong Hall property, then there exist values for the nonzeros of A such that $G(R) = G_\cap^+(A)$, where R is the orthogonal factor of A as above.* □

Hare, Johnson, Olesky, and van den Driessche [27] gave a more complicated prediction of the structure of both R and the orthogonal factor Q. They showed that their prediction was tight in the "one-at-a-time" sense for all Hall structures, that is, for all structures with full symbolic column rank. Pothen [34] then proved that the prediction of Hare et al. was in fact tight in the "all-at-once" sense, thus finishing off the problem for both Q and R in the Hall case.

George and Ng [18] studied another representation of the structure of the orthogonal factor Q in the case that A is square and has nonzero diagonal elements. They showed that a suitable representation of Q has a structure that also satisfies Theorem 4.7. Suppose A is reduced to upper triangular form by a sequence of Householder transformations that zero the subdiagonal elements of the first column, then the second column, and so on. The Householder transformation used to zero column j (for $1 \leq j < n$) has the form $Q_j = I - w_j w_j^T$, where w_j is a column vector whose first $j-1$

entries are zero. The orthogonal factor $Q = Q_1^T Q_2^T \dots Q_{n-1}^T$ is conveniently represented by the lower triangular matrix W whose columns are the w_j. The George–Ng result describes the structure of this triangular matrix.

THEOREM 4.9 (see [18]). *Let a structure $G(A)$ be given for a square matrix A with nonzero diagonal elements. Whatever values A has, if its orthogonal factor is represented by the matrix W described above, then $G(W) \subseteq G_\cap^+(A)$.* □

This result is useful because in practice it often suffices to represent the orthogonal factor by the sequence of Householder vectors, and that representation is often sparser than an explicit representation of Q. For example, consider a symmetric structure whose graph $G(A)$ is a square grid with n vertices, corresponding to the standard five-point finite difference stencil on a $k \times k$ mesh with $n = k^2$. The number of nonzeros in A is $\Theta(n)$. Take the numeric values of A to be algebraically independent. It is straightforward to use the results cited above [18], [27] to show that a nested dissection ordering of $G_\cap(A)$ asymptotically minimizes the number of nonzeros in all of Q, R, and W, and that for such an ordering the number of nonzeros in both R and W is $\Theta(n \log n)$ while the number of nonzeros in Q is $\Theta(n^{3/2})$. Thus in this model problem W is a much more efficient way to store the orthogonal factor than Q.

5. Solutions of linear systems. In this section we determine the structure of the solution x to the square system of linear equations $Ax = b$. We solve the related problem of determining the structure of A^{-1}. These results have not appeared before in this form, but the upper bounds in Theorem 5.1 and Corollary 5.4 are straightforward consequences of Tarjan's work on elimination methods for solving path problems in graphs [40] and are closely related to work done by Fiedler [13]. The proofs here are somewhat different.

The extremes of path structure in directed graphs are a strongly connected graph (which corresponds to an irreducible matrix) and an acyclic graph (which corresponds to a permutation of a triangular matrix). Some of these results become almost trivial at the strongly connected extreme—for example, the inverse of an irreducible matrix is full in the absence of coincidental cancellation. In solving general nonsingular linear systems it is often advantageous to begin by partitioning the matrix into strong components and then to factor only the irreducible blocks of the partition. This approach is taken, for example, in the Duff and Reid MA28 code [9].

Curiously enough, the most important applications of the results in this section are at the opposite extreme, for triangular systems. Structure prediction for sparse triangular systems is used in efficient algorithms for LU factorization with partial pivoting [24] and in parallel triangular solution [1].

Throughout this section A is an $n \times n$ matrix with nonzero diagonal, and the graph in question is the directed graph $G(A)$.

THEOREM 5.1. *Let the structures of A and b be given.*

(i) *Whatever the values of the nonzeros in A and b, if A is nonsingular then*

$$\text{struct}(A^{-1}b) \subseteq \text{closure}(b).$$

(ii) *There exist nonzero values for which* $\text{struct}(A^{-1}b) = \text{closure}(b)$. *(In fact, all the nonzeros in b can have the value* 1.)

Proof. Part (i). Let values be given for which A is nonsingular. Renumber the vertices of A so that $\text{closure}(b) = \{1, 2, \dots, k\}$ for some $k \leq n$. Then $Ax = b$ can be partitioned as

$$\begin{pmatrix} B & D \\ C & E \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix},$$

where B is $k \times k$. By the definition of closure(b), there is no edge (i,j) with $i \notin$ closure(b) and $j \in$ closure(b). Therefore $C = 0$. Then $Ez = 0$. Since A is nonsingular and $C = 0$, matrix E is nonsingular. Therefore $z = 0$. Thus struct(x) $\subseteq \{1, \ldots, k\} =$ closure(b).

Part (ii). Choose algebraically independent values for the nonzeros of A, and let $b_i = 1$ if $i \in$ struct(b). Then A is nonsingular because $\det A$ is a nonzero polynomial in the nonzeros of A. Let $x = A^{-1}b$. Renumber the vertices of A so that struct(x) $= \{1, 2, \ldots, k\}$ for some $k \leq n$. Then $Ax = b$ can be partitioned as

$$\begin{pmatrix} B & D \\ C & E \end{pmatrix} \begin{pmatrix} y \\ 0 \end{pmatrix} = \begin{pmatrix} d \\ e \end{pmatrix},$$

where B is $k \times k$ and all entries of y are nonzero. Consider row i of C. We have

$$\sum_{1 \leq j \leq k} c_{ij} y_j = e_i. \tag{1}$$

Now B is nonsingular, since $\det B$ is a nonzero polynomial. By Cramer's rule, $By = d$ implies $y_j = \det(B|_j^d)/\det B$, where $B|_j^d$ is B with column j replaced by d. Then (1) implies

$$\sum_{i \leq j \leq k} c_{ij} \det(B|_j^d) - e_i \det B = 0. \tag{2}$$

This is a polynomial with rational coefficients in the entries of A, so it is the zero polynomial. Now $y_j \neq 0$ implies that $\det(B|_j^d)$ is not the zero polynomial, so c_{ij} must be zero. Thus $C = 0$. This implies that $x = \binom{y}{0}$ is closed. Furthermore, $\det B \neq 0$, so (2) implies $e_i = 0$. Thus $e = 0$, so $b = \binom{d}{0}$ and struct(b) $\subseteq$ struct(x) $=$ closure(x). Therefore closure(b) $\subseteq$ struct(x). With part (i), this gives closure(b) $=$ struct(x). □

Remark 5.2. The proof of part (i) never assumes that the "nonzero values" of A are in fact different from zero. Thus we have the slightly stronger result that if $G(\hat{A}) \subseteq G(A)$ and struct($\hat{b}$) $\subseteq$ struct(b) and $\hat{A}$ is nonsingular, then struct($\hat{A}^{-1}\hat{b}$) $\subseteq$ closure(b).

Remark 5.3. It seems natural to conjecture in part (i) that if A is singular and $Ax = b$ has a solution, then it has some solution with struct(x) $\subseteq$ closure(b). Oddly enough, this is false. Consider

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

All solutions to $Ax = b$ are of the form $(\alpha, -\alpha, 1, 1)^T$, none of which is a subset of closure(b) $= (\times, \times, 0, 0)^T$.

COROLLARY 5.4. *Let the structure $G(A)$ be given.*

(i) *Whatever values A has, if A is nonsingular then $G(A^{-1}) \subseteq G^*(A)$.*

(ii) *Values exist for the nonzeros of A such that $G(A^{-1}) = G^*(A)$.*

Proof. Note that column j of $G^*(A)$ is closure($e^{(j)}$), where $e^{(j)}$ is the jth unit vector. The corollary is immediate from Theorem 5.1, noting that part (ii) of the theorem holds even if the right-hand-side entries are all zeros and ones. □

Corollary 5.4 implies that if A is irreducible with nonzero diagonal, then A^{-1} is full unless numerical coincidence occurs. Duff et al. [8] gave another proof of this special case.

The case where A is allowed to have zeros on the diagonal is a straightforward extension. First, for A^{-1} to exist, $H(A)$ must be Hall. That implies that a permutation P exists for which PA has nonzero diagonal. Then the structure of the solution to $Ax = b$ is the structure of $A^{-1}b = (PA)^{-1}(Pb)$, which is the closure of Pb with respect to the graph $G(PA)$ by Theorem 5.1. The structure of A^{-1} can be predicted similarly by permuting to nonzero diagonal, forming the transitive closure, and permuting back. This also implies a slightly stronger version of the result of Duff et al.: if nonsingular A is fully indecomposable, or, equivalently, strong Hall (with no restriction on the diagonal), then A^{-1} is full unless numerical coincidence occurs.

The case where A is symmetric is simpler and less interesting, but the puzzling examples such as the one in Remark 5.3 do not arise. If A is symmetric and its graph is not connected, then A is block diagonal, and a linear system splits into a separate problem for each block. If A is connected, then it is strongly connected and the closure of every nonempty set is the whole graph. Then the upper bound in part (i) of Theorem 5.1 is trivial, and values exist to achieve it.

THEOREM 5.5. *Let a symmetric structure for A be given along with a nonzero structure for b. If the structure for A is connected (i.e., irreducible, or not block diagonal), then there exist symmetric values for A such that* $\text{struct}(A^{-1}b) = \{1, 2, \ldots, n\}$; *that is, x is full. Also, in this case, A^{-1} is full.*

Proof. The proof is almost identical to that of Theorem 5.1 (ii), so this is just a sketch: Choose algebraically independent values for the lower triangle of A and make the upper triangle symmetric. Then A is nonsingular. The polynomial in (2) does not contain c_{ji}, so we can still conclude $c_{ij} = 0$ from the fact that it occurs multiplied by a nonzero polynomial. Therefore $A^{-1}b$ is closed. But if symmetric A is connected, then it is strongly connected, so the only nonempty closed set is $\{1, 2, \ldots, n\}$. □

6. Eigenvectors. In this section we determine the structure of the eigenvectors of a square matrix A. The results in this section are new. We deal only with the case of distinct eigenvalues. As described at the end of the section, the reason we cannot handle multiple eigenvalues is related to Remark 5.3 above.

Throughout this section A is an $n \times n$ matrix with nonzero diagonal, and the graph in question is the directed graph of A. Recall that $e^{(i)}$ is the ith unit vector and $\text{closure}(e^{(i)})$ is the structure of column i of the transitive closure of A.

THEOREM 6.1. *Let the structure $G(A)$ be given.*

(i) *Whatever the values of the nonzeros in A, if A has n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$, then the eigenvectors of A can be numbered $u^{(1)}, \ldots, u^{(n)}$ such that* $\text{struct}(u^{(i)}) \subseteq \text{closure}(e^{(i)})$.

(ii) *There exist nonzero values for which A has n distinct eigenvalues, and the eigenvectors satisfy* $\text{struct}(u^{(i)}) = \text{closure}(e^{(i)})$.

Proof. Part (i). Let values be given for A. Renumber the vertices of A to put A in block upper triangular form—that is, to put the strongly connected components of A in topological order. Then A is partitioned as

$$A = \begin{pmatrix} B_1 & C_{1,2} & \cdots & C_{1,s} \\ & B_2 & \cdots & C_{2,s} \\ & & \ddots & \vdots \\ 0 & & & B_s \end{pmatrix},$$

where each B_j is square and strongly connected. Renumber the eigenvalues and eigenvectors in nondecreasing order of the highest-numbered nonzero in the eigenvector.

That is, if $u_k^{(i)} = u_{k+1}^{(i)} = \cdots = u_n^{(i)} = 0$, then $u_k^{(i-1)} = u_{k+1}^{(i-1)} = \cdots = u_n^{(i-1)} = 0$, for $1 < i \leq n$.

Consider some eigenvector $u^{(i)}$. Suppose its highest-numbered nonzero is in a row that runs through block B_j. Then $Au^{(i)} = \lambda_i u^{(i)}$ is partitioned as

$$\begin{pmatrix} D & E & F \\ 0 & B_j & G \\ 0 & 0 & H \end{pmatrix} \begin{pmatrix} v \\ w \\ 0 \end{pmatrix} = \lambda_i \begin{pmatrix} v \\ w \\ 0 \end{pmatrix}, \qquad \text{where } D = \begin{pmatrix} B_1 & \cdots & C_{1,j-1} \\ & \ddots & \vdots \\ 0 & & B_{j-1} \end{pmatrix}, \quad \text{etc.}$$

Then $B_j w = \lambda_i w$ with $w \neq 0$, so λ_i is an eigenvalue of B_j. In fact, each λ_i is an eigenvalue of one B_j, with j increasing as i increases. Since no B_j has more eigenvalues than its dimension, we conclude by counting rows that row i and column i of A run through B_j. Now B_j is strongly connected, so closure$(e^{(i)})$ = closure(B_j) (where closure(B_j) denotes the closure of the set of vertices of B_j with respect to A).

We have $Dv + Ew = \lambda_i v$, so

$$(D - \lambda_i I)v = Ew.$$

Since the eigenvalues of A are simple, λ_i is not an eigenvalue of D and $D - \lambda_i I$ is nonsingular. Thus, by Theorem 5.1,

$$\text{struct}(v) \subseteq \text{closure}(Ew).$$

Now if D is $m \times m$ and B_j is $t \times t$, struct$(w) \subseteq \{m+1, m+2, \ldots, m+t\}$ and struct$(Ew) \subseteq \{1 \leq k \leq m : a_{kl} \neq 0 \text{ for some } l \in \text{struct}(w)\}$ by Theorem 3.1, so struct$(Ew) \subseteq$ closure(w) (closure still with respect to A) and struct$(v) \subseteq$ closure(w). Therefore, struct$(u^{(i)})$ = struct$(v) \cup$ struct$(w) \subseteq$ closure(w). Since $w \subseteq B_j$, this implies that struct$(u^{(i)}) \subseteq$ closure(B_j) = closure$(e^{(i)})$.

Part (ii). Choose algebraically independent values for A, choosing the diagonal elements so far apart that no two are closer than $2 \max_j \sum_{i \neq j} |a_{ij}|$. By Gerschgorin's theorem [25], this guarantees that there are n distinct, simple eigenvalues. (It would be more elegant to conclude that the eigenvalues are simple from the algebraic independence of the elements, but I do not know how to prove it.)

First we will show that each eigenvector is closed. Let u be an eigenvector with $Au = \lambda u$. Renumber the vertices of A so that struct$(u) = \{1, 2, \ldots, t\}$ for some $t \leq n$. Then $Au = \lambda u$ can be partitioned as

$$\begin{pmatrix} B & D \\ C & E \end{pmatrix} \begin{pmatrix} v \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} v \\ 0 \end{pmatrix}, \tag{3}$$

where B is $t \times t$ and $v_k \neq 0$ for $1 \leq k \leq t$. We will show $C = 0$. Intuitively, it seems clear that if $C \neq 0$, then v cannot be both an eigenvector of B and a null vector of C. A field-theoretic argument makes this intuition precise.

Since $Bv = \lambda v$ and the diagonal elements of B are far enough apart that their Gerschgorin discs do not overlap, λ is in the Gerschgorin disc of exactly one b_{kk}. Renumber vertices 1 through t so that b_{kk} is b_{11}. Choose v such that $v_1 = 1$. Then $Bv = \lambda v$ partitions into

$$\begin{pmatrix} b_{11} & f^T \\ g & B' \end{pmatrix} \begin{pmatrix} 1 \\ v_2 \\ \vdots \\ v_t \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ v_2 \\ \vdots \\ v_t \end{pmatrix},$$

where f and g are $t-1$-vectors. Now we have

$$(B' - \lambda I)\begin{pmatrix} v_2 \\ \vdots \\ v_t \end{pmatrix} = -g.$$

By Gerschgorin's theorem, λ is not an eigenvalue of B', so $B' - \lambda I$ is nonsingular and

$$v_k = \frac{\det(B' - \lambda I)|_k^{-g}}{\det(B' - \lambda I)} \qquad \text{for } 2 \le k \le t. \tag{4}$$

Now we fix i and j and show that $c_{ij} = 0$ (for $1 \le i \le n-t$ and $1 \le j \le t$).

Let F be the field obtained by adjoining to $\mathbf{Q}$ (the rationals) all the nonzeros of B and all the nonzeros of row i of C except c_{ij}. Now $F[x]$ is the ring of one-variable polynomials with coefficients in F, and $F(\lambda)$ is the field obtained by adjoining λ to F. We know λ is a zero of a nonzero polynomial in $F[x]$, namely, $\det(B - xI)$. Therefore, λ is algebraic over F, so every element of $F(\lambda)$ is a zero of some nonzero polynomial in $F[x]$.

Since $Cv = 0$, we have

$$\sum_{1 \le k \le t} c_{ik} v_k = 0.$$

All the v_k are nonzero, so

$$c_{ij} = -\frac{1}{v_j} \sum_{k \ne j} c_{ik} v_k. \tag{5}$$

By (4), each v_k is a rational function of λ and elements of F, so $v_k \in F(\lambda)$. Each c_{ik} with $k \ne j$ is in F. Therefore, the whole right-hand side of (5) is in $F(\lambda)$, so $c_{ij} \in F(\lambda)$. This means that c_{ij} is a zero of a nonzero polynomial in $F[x]$. But if c_{ij} is nonzero, then c_{ij} was chosen to be transcendental over F. Thus $c_{ij} = 0$; and, since i and j were arbitrary, $C = 0$.

Recalling the partition of A in (3), $C = 0$ implies that the eigenvector $u = \binom{v}{0}$ is closed.

Now all the eigenvectors of A are closed. Renumber the eigenvectors so that λ_i is in the Gerschgorin disc of a_{ii}. The argument following (3) shows that λ_i is in a Gerschgorin disc whose index j corresponds to a nonzero $u_j^{(i)}$ of $u^{(i)}$; since λ_i is in only one disc, this means $u_i^{(i)} \ne 0$. Therefore, $\text{struct}(e^{(i)}) \subseteq \text{struct}(u^{(i)})$. Since $u^{(i)}$ is closed, $\text{closure}(e^{(i)}) \subseteq \text{struct}(u^{(i)})$. Part (i) gives the opposite containment, so $\text{struct}(u^{(i)}) = \text{closure}(e^{(i)})$. □

COROLLARY 6.2. *Let the structure of A be given.*

(i) *No matter what nonzero values A has, if A has only simple eigenvalues, then its eigenvectors can be ordered so that the matrix U whose columns are the eigenvectors has $G(U) \subseteq G^*(A)$.*

(ii) *There exist values for the nonzeros of A such that the eigenvectors can be ordered so that $G(U) = G^*(A)$.*

Proof. Similar to Corollary 5.4. □

Remark 6.3. It is natural to conjecture that if A has multiple eigenvalues, then there is some choice of a maximal set of eigenvectors whose structure is a subset of the

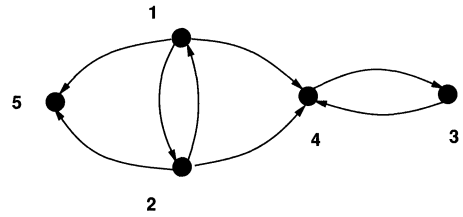


FIG. 4. *Graph of counterexample in Remark* 6.3.

transitive closure of A. Again, oddly, this is false. From the example in Remark 5.3 we can construct

$$A = \begin{pmatrix} 3 & 1 & 0 & 1 & -1 \\ 2 & 4 & 0 & 1 & -1 \\ 0 & 0 & 3 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}, \quad \text{so } G^*(A) = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & & \times & \times & \\ & & \times & \times & \\ & & & & \times \end{pmatrix}.$$

The graph of A is shown in Fig. 4. The characteristic equation of A is $\det(xI - A) = (x-2)^4(x-5)$, so the eigenvalues are 2 and 5. The eigenspace of 5 is one-dimensional and consists of multiples of $(1,2,0,0,0)^T$, which comprise a column of the transitive closure. However, the eigenspace of 2 is also one-dimensional and consists of multiples of $(0,0,1,1,1)^T$, which do not form a subset of any column of the transitive closure.

Mascarenhas [31] has recently extended Theorem 6.1 to the case where A has multiple eigenvalues, provided that no two diagonal blocks of the block upper triangular form of A share an eigenvalue. He has also proved a similar result for the more general case where there are n linearly independent eigenvectors, that is, where each eigenvalue has equal geometric and algebraic multiplicity.

For symmetric A, the situation for eigenvectors is the same as for symmetric linear systems: If A is block diagonal, then each block is a separate problem; if A is not block diagonal (i.e., A is irreducible or connected), then the upper bounds are both trivial and tight.

THEOREM 6.4. *Let a symmetric structure for A be given. If the structure is connected, then there exist symmetric values for A such that A has n distinct eigenvalues, and all its eigenvectors are full.*

Proof. Just as in Theorem 5.5, the proof of Theorem 6.1 part (ii) goes through, even if A is required to be symmetric. □

7. Remarks, applications, and open problems. We have described several matrix computations in which the nonzero structure of the result of the computation can be inferred, partly or completely, from the nonzero structure of the input to the computation. The language of graph theory seems most appropriate to state these results, primarily because path structure is most easily described in graph-theoretic terms.

Matchings in bipartite graphs are important in several of the results of §4. Bipartite matching theory plays a central role in two other structural problems that we have not described here: finding the sparsest basis for the range space (McCormick [32]) and for the null space (Coleman and Pothen [5], [6], Gilbert and Heath [21]) of a rectangular matrix. It turns out that the structural range space problem can be solved in polynomial time, but the null space problem is NP-complete.

Several applications of structure prediction to solving systems of linear equations were cited in §1. Some of the present work was motivated by Gilbert and Peierls's use of structure prediction for triangular linear systems as part of an efficient algorithm for sparse LU factorization with partial pivoting [24]. Another application of structure prediction for triangular systems is in a practical problem in reservoir analysis. Here a finite-element model of an underground reservoir of hot water (to be tapped for power and heating for the city of Reykjavík) requires the solution of hundreds of positive definite linear systems with the same coefficient matrix. All the systems have very sparse right-hand sides, and, in addition, only a few of the unknown values are required for each system. Sigurðsson [38] has used structure prediction with a simpler version of Theorem 5.1 to speed up the Sparspak triangular solver for this problem.

We have been concerned exclusively with predicting nonzero structure in this paper. A related question is: Given a matrix and a matrix function, which entries of the matrix are unchanged in value by application of the function? Barrett, Johnson, Olesky, and van den Driessche [2], [28] have given such characterizations for functions including LU factorization and Schur complement.

A few open problems in structure prediction, some of which have already been mentioned, are as follows. Is it possible to give a tight bound on the nonzero structures of the factors in Gaussian elimination with partial pivoting (§4)? What can be said about solutions to singular linear systems in light of the counterexample in §5? What can be said about eigenvector structures for matrices with multiple eigenvalues ([31, §6])? What can be said about the structure of the singular value decomposition (SVD) of a rectangular matrix [25]? The relationship between the singular values of A and the eigenvalues of A^TA, together with Theorem 6.4 on eigenvectors of symmetric matrices, suggests that the SVD of a connected matrix is always full (ignoring numerical cancellation). This would certainly confirm the conventional wisdom that there is no such thing as an SVD with sparse singular vectors.

Acknowledgments. My thanks to Tom Coleman, Anders Edenbrandt, Mike Heath, Joseph Liu, Esmond Ng, Ragnar Sigurðsson, and Sven Sigurðsson for interesting and useful discussions of these problems. Earl Zmijewski gave this paper a careful and helpful critical reading. Gene Golub nudged me to finish revising it for publication.

REFERENCES

[1] F. Alvarado and R. Schreiber, *Optimal parallel solution of sparse triangular systems*, SIAM J. Sci. Comput., 14 (1993) , pp. 446–460.

[2] W. W. Barrett, C. R. Johnson, D. D. Olesky, and P. van den Driessche, *Inherited matrix entries: Principal submatrices of the inverse*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 313–322.

[3] R. K. Brayton, F. G. Gustavson, and R. A. Willoughby, *Some results on sparse matrices*, Math. Comput., 24 (1970), pp. 937–954.

[4] T. F. Coleman, A. Edenbrandt, and J. R. Gilbert, *Predicting fill for sparse orthogonal factorization*, J. Assoc. Comput. Mach., 33 (1986), pp. 517–532.

[5] T. F. Coleman and A. Pothen, *The null space problem* I: *Complexity*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 527–537.

[6] ———, *The null space problem* II: *Algorithms*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 544–563.

[7] I. S. Duff, *On algorithms for obtaining a maximum transversal*, ACM Trans. Math. Software, 7 (1981), pp. 315–330.

[8] I. S. Duff, A. M. Erisman, C. W. Gear, and J. K. Reid, *Some remarks on the inverses of sparse matrices*, Mathematics and Computer Science Division Report 51, Argonne National Laboratory, Argonne, IL, 1985.

[9] I. S. DUFF AND J. K. REID, *Some design features of a sparse matrix code*, ACM Trans. Math. Software, 5 (1979), pp. 18–35.
[10] A. G. EDENBRANDT, *Combinatorial Problems in Matrix Computation*, Ph.D. thesis, Cornell University, Ithaca, NY, 1985.
[11] 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.
[12] S. C. EISENSTAT, M. H. SCHULTZ, AND A. H. SHERMAN, *Algorithms and data structures for sparse symmetric Gaussian elimination*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 225–237.
[13] M. FIEDLER, *Inversion of bigraphs and connection with the Gauss elimination*, in Graphs, Hypergraphs, and Block Systems, Zielona Gora, Czechoslovakia, 1976, pp. 57–68.
[14] A. GEORGE AND M. T. HEATH, *Solution of sparse linear least squares problems using Givens rotations*, Linear Algebra Appl., 34 (1980), pp. 69–83.
[15] A. GEORGE AND J. W. H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Engelwood Cliffs, NJ, 1981.
[16] A. GEORGE AND E. NG, *An implementation of Gaussian elimination with partial pivoting for sparse systems*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 390–409.
[17] ———, *Symbolic factorization for sparse Gaussian elimination with partial pivoting*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 877–898.
[18] ———, *On the complexity of sparse QR and LU factorization of finite-element matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 849–861.
[19] J. R. GILBERT, *Predicting structure in sparse matrix computations*, Tech. Report 86–750, Cornell University, Ithaca, NY, 1986.
[20] ———, *An efficient parallel sparse partial pivoting algorithm*, Tech. Report 88/45052-1, Christian Michelsen Institute, Bergen, Norway, 1988.
[21] J. R. GILBERT AND M. T. HEATH, *Computing a sparse basis for the null space*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 446–459.
[22] 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.
[23] J. R. GILBERT AND E. NG, *Predicting Structure in Nonsymmetric Sparse Matrix Factorizations*, Tech. Report CSL-92-8, Xerox Palo Alto Research Center, 1992; Graph Theory and Sparse Matrix Computation, A. George, J. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, 1993.
[24] 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.
[25] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.
[26] F. HARARY, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.
[27] D. R. HARE, C. R. JOHNSON, D. D. OLESKY, AND P. VAN DEN DRIESSCHE, *Sparsity analysis of the QR factorization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 655–669.
[28] ———, *Inherited matrix entries: LU factorization*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 94–104.
[29] J. W. H. LIU, *Computational models and task scheduling for parallel sparse Cholesky factorization*, Parallel Comput., 3 (1986), pp. 327–342.
[30] L. LOVASZ AND M. D. PLUMMER, *Matching Theory*, North-Holland, Amsterdam, 1986.
[31] W. MASCARENHAS, *Predicting structure in eigenvector computation*, Linear Algebra Appl., to appear.
[32] S. T. MCCORMICK, *A Combinatorial Approach to Some Sparse Matrix Problems*, Ph.D. thesis, Stanford University, Stanford, CA, 1983.
[33] S. PARTER, *The use of linear graphs in Gauss elimination*, SIAM Rev., 3 (1961), pp. 119–130.
[34] A. POTHEN, *Predicting the Structure of Sparse Orthogonal Factors*, manuscript, 1991.
[35] D. J. ROSE, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, Graph Theory and Computing, Ronald C. Read, ed., Academic Press, New York, 1972, pp. 183–217.
[36] D. J. ROSE AND R. E. TARJAN, *Algorithmic aspects of vertex elimination on directed graphs*, SIAM J. Appl. Math., 34 (1978), pp. 176–197.
[37] D. J. ROSE, R. E. TARJAN, AND G. S. LUEKER, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
[38] S. SIGURÐSSON, *Sparse matrix techniques in geothermal reservoir modelling*, manuscript, 1991.

[39] R. E. TARJAN AND M. YANNAKAKIS, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 13 (1984), pp. 566–579.

[40] R. E. TARJAN, *A unified approach to path problems*, J. Assoc. Comput. Mach., 28 (1981), pp. 577–593.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 80–97, January 1994

005

CIRCULANT PRECONDITIONED TOEPLITZ LEAST SQUARES ITERATIONS*

RAYMOND H. CHAN†, JAMES G. NAGY‡, AND ROBERT J. PLEMMONS§

Abstract. The authors consider the solution of least squares problems $\min \|b - Tx\|_2$ by the preconditioned conjugate gradient method, for m-by-n complex Toeplitz matrices T of rank n. A circulant preconditioner C is derived using the T. Chan optimal preconditioner on n-by-n Toeplitz row blocks of T. For Toeplitz T that are generated by 2π-periodic continuous complex-valued functions without any zeros, the authors prove that the singular values of the preconditioned matrix TC^{-1} are clustered around 1, for sufficiently large n. The paper shows that if the condition number of T is of $O(n^\alpha)$, $\alpha > 0$, then the least squares conjugate gradient method converges in at most $O(cd\alpha \log n + 1)$ steps. Since each iteration requires only $O(m \log n)$ operations using the Fast Fourier Transform, it follows that the total complexity of the algorithm is then only $O(\alpha m \log^2 n + m \log n)$. Conditions for *superlinear convergence* are given and regularization techniques leading to superlinear convergence for least squares computations with ill-conditioned Toeplitz matrices arising from inverse problems are derived. Numerical examples are provided illustrating the effectiveness of the authors' methods.

Key words. least squares, Toeplitz matrix, circulant matrix, preconditioned conjugate gradients, regularization

AMS subject classifications. 65F10, 65F15

1. Introduction.

The conjugate gradient (CG) method is an iterative method for solving Hermitian positive definite systems $Ax = b$ (see, for instance, Golub and Van Loan [21]). When A is a rectangular m-by-n matrix of rank n, one can still use the CG algorithm to find the solution to the least squares problem

$$\min \|b - Ax\|_2. \tag{1}$$

This can be done by applying the algorithm to the normal equations in factored form,

$$A^*(b - Ax) = 0, \tag{2}$$

which can be solved by conjugate gradients without explicitly forming the matrix A^*A (see Bjorck [7]).

The convergence of the CG algorithm and its variations depends on the singular values of the data matrix A (see Axelsson [5]). If the singular values cluster around a fixed point, convergence will be rapid. Thus, to make the algorithm a useful iterative method, one usually *preconditions* the system. The preconditioned conjugate gradient (PCG) algorithm then solves (1) by transforming the problem with a preconditioner M, applying the CG method to the transformed problem, and then transforming back. More precisely, one can use the CG method to solve

$$\min \|b - AM^{-1}y\|_2,$$

* Received by the editors December 2, 1991; accepted for publication May 27, 1992.

† Department of Mathematics, University of Hong Kong, Hong Kong (rchan@uxmail.ust.hk).

‡ Institute for Mathematics and Its Applications, University of Minnesota, Minneapolis, Minnesota 55455 (nagy@cygnus.math.smu.edu).

§ Department of Mathematics and Computer Science, Wake Forest University, P. O. Box 7388, Winston-Salem, North Carolina 27109 (plemmons@deacon.mthcsc.wfu.edu). The paper was completed while this author was visiting the Institute for Mathematics and Its Applications, University of Minnesota, Minneapolis, Minnesota 55455. The research of this author was supported by United States Air Force grant AFOSR-91-0163.

and then set $x = M^{-1}y$.

In this paper we consider the least squares problem (1), with the data matrix $A = T$, where T is a rectangular m-by-n Toeplitz matrix of rank n. The matrix $T = (t_{jk})$ is said to be *Toeplitz* if $t_{jk} = t_{j-k}$, i.e., T is constant along its diagonals. An n-by-n matrix C is said to be *circulant* if it is Toeplitz and its diagonals c_j satisfy $c_{n-j} = c_{-j}$ for $0 < j \leq n-1$. Toeplitz least squares problems occur in a variety of applications, especially in signal and image processing. (See Andrews and Hunt [3], Jain [24], and Oppenheim and Schafer [28].)

Recall that the solution to the least squares problem

$$\min \|b - Tx\|_2 \tag{3}$$

can be found by the PCG method by applying the method to the normal equations (2) in factored form, that is, using T and T^* without forming T^*T. The preconditioner M considered in this paper is given by an n-by-n circulant matrix $M = C$, where C^*C is then a circulant matrix that approximates T^*T.

The version of the PCG algorithm we use is given in [7] and can be stated as follows.

Algorithm PCG for Least Squares. *Let $x^{(0)}$ be an initial approximation to $Tx = b$, and let C be a given preconditioner. This algorithm computes the least squares solution, x, to $Tx = b$.*

$r^{(0)} = b - Tx^{(0)}$
$p^{(0)} = s^{(0)} = C^{-*}T^*r^{(0)}$
$\gamma_0 = \|s^{(0)}\|_2^2$
for $k = 0, 1, 2, \ldots$
  $q^{(k)} = TC^{-1}p^{(k)}$
  $\alpha_k = \gamma_k / \|q^{(k)}\|_2^2$
  $x^{(k+1)} = x^{(k)} + \alpha_k C^{-1}p^{(k)}$
  $r^{(k+1)} = r^{(k)} - \alpha_k q^{(k)}$
  $s^{(k+1)} = C^{-*}T^*r^{(k+1)}$
  $\gamma_{k+1} = \|s^{(k+1)}\|_2^2$
  $\beta_k = \gamma_{k+1}/\gamma_k$
  $p^{(k+1)} = s^{(k+1)} + \beta_k p^{(k)}$

The idea of using the PCG method with circulant preconditioners for solving square positive definite Toeplitz systems was first proposed by Strang [30], although the application of circulant approximations to Toeplitz matrices has been used for some time in image processing, e.g., in [6]. The convergence rate of the method was analyzed by R. Chan and Strang [9] for Toeplitz matrices that are generated by positive Wiener class functions. Since then, considerable research has been done in finding other good circulant preconditioners or extending the class of generating functions for which the method is effective. (See T. Chan [17], R. Chan [10], Tyrtyshnikov [32], Tismenetsky [31], Huckle [23], Ku and Kuo [25], R. Chan and Yeung [13], T. Chan and Olkin [18], R. Chan and Jin [12], and R. Chan and Yeung [14].)

Recently, the idea of using circulant preconditioners has been extended to non-Hermitian square Toeplitz systems by R. Chan and Yeung [15] and to Toeplitz least squares problems by Nagy [26] and by Nagy and Plemmons [27]. The main aims of this paper are to formalize and establish convergence results and to provide applications in the case where T is a rectangular Toeplitz (block) matrix.

For the purpose of constructing the preconditioner, we will see that by extending the Toeplitz structure of the matrix T and, if necessary, padding zeros to the bottom left-hand side, we may assume without loss of generality that $m = kn$ for some positive integer k. This padding is only for convenience in constructing the preconditioner and does not alter the original least squares problem. In the material to follow, we consider the case where k is a constant independent of n. More precisely, we consider in this paper, kn-by-n matrices T of the form

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_k \end{bmatrix}, \tag{4}$$

where each square block T_j is a Toeplitz matrix. Notice that if T itself is a rectangular Toeplitz matrix, then each block T_j is necessarily Toeplitz.

Following [26], [27], for each block T_j, we construct a circulant approximation C_j. Then our preconditioner is defined as a square circulant matrix C, such that

$$C^*C = \sum_{j=1}^{k} C_j^* C_j.$$

Notice that each C_j is an n-by-n circulant matrix. Hence they can all be diagonalized by the Fourier matrix F, i.e.,

$$C_j = F \Lambda_j F^*,$$

where Λ_j is diagonal (see Davis [19]). Therefore, the spectrum of C_j, $j = 1, \ldots, k$, can be computed in $O(n \log n)$ operations by using the Fast Fourier Transform (FFT). Since

$$C^*C = F \sum_{j=1}^{k} (\Lambda_j^* \Lambda_j) F^*,$$

C^*C is also circulant and its spectrum can be computed in $O(kn \log n)$ operations. Here we choose, as in [26], [27],

$$C = F \left(\sum_{j=1}^{k} \Lambda_j^* \Lambda_j \right)^{1/2} F^*. \tag{5}$$

The number of operations per iteration in Algorithm PCG for Least Squares depends mainly on the work of computing the matrix-vector multiplications. In our case, this amounts to computing products:

$$Ty, \quad T^*z, \quad C^{-1}y, \quad C^{-*}y$$

for some n-vectors y and m-vectors z. Since

$$C^{-1}y = F \left(\sum_{j=1}^{k} \Lambda_j^* \Lambda_j \right)^{-1/2} F^* y,$$

the products $C^{-1}y$ and $C^{-*}y$ can be found efficiently by using the FFT in $O(n \log n)$ operations. For the products Ty and T^*z, with T in block form with k n-by-n blocks T_j, we have to compute n products of the form $T_j w$, where T_j is an n-by-n Toeplitz matrix and w is an n-vector. However, the product $T_j w$ can be computed using the FFT by first embedding T_j into a $2n$-by-$2n$ circulant matrix. The multiplication thus requires $O(2n \log(2n))$ operations. It follows that the operations for computing Ty and T^*z are of the order $O(m \log n)$, where $m = nk$. Thus we conclude that the cost per iteration in the PCG method is of the order $O(m \log n)$.

As already mentioned in the beginning, the convergence rate of the method depends on the distribution of the singular values of the matrix TC^{-1}, which are the same as the square roots of the eigenvalues of the matrix $(C^*C)^{-1}(T^*T)$. We will show, then, that if the generating functions of the blocks T_j are 2π-periodic continuous functions and if one of these functions has no zeros, then the spectrum of $(C^*C)^{-1}(T^*T)$ will be clustered around 1, for sufficiently large n. We remark that the class of 2π-periodic continuous functions contains the Wiener class of functions, which in turn contains the class of rational functions considered in Ku and Kuo [25].

By using a standard error analysis of the CG method, we then show that if the condition number $\kappa(T)$ of T is of $O(n^\alpha)$, then the number of iterations required for convergence, for sufficiently large n, is at most $O(\alpha \log n + 1)$, where $\alpha > 0$. Since the number of operations per iteration in the CG method is of $O(m \log n)$, the total complexity of the algorithm is therefore of $O(\alpha m \log^2 n + m \log n)$. In the case in which $\alpha = 0$, i.e., T is well conditioned, the method converges in $O(1)$ steps. Hence the complexity is reduced to just $O(m \log n)$ operations, for sufficiently large n. On the other hand, the superfast direct algorithms by Ammar and Gragg [2] require $O(n \log^2 n)$ operations for n-by-n Toeplitz linear systems. The stability of fast direct methods has been studied by Bunch [8].

The outline of the paper is as follows. In §2, we construct the circulant preconditioners C for the Toeplitz least squares problem and study some of the spectral properties of these preconditioners. In §3, we show that the iteration matrix TC^{-1} has singular values clustered around 1. In §4, we then establish the convergence rate of the PCG method when applied to the preconditioned system, and indicate when it is superlinear. In §5, we discuss the technique of regularization when the given Toeplitz matrix T is ill conditioned. Numerical results and concluding remarks are given in §6.

2. Properties of the circulant preconditioner. In this section, we consider circulant preconditioners for least square problems and study their spectral properties. We begin by recalling some results for square Toeplitz systems.

For simplicity, we denote by $C_{2\pi}$ the Banach space of all 2π-periodic continuous complex-valued functions equipped with the supremum norm $\|\cdot\|_\infty$. As already mentioned in §1, this class of functions contains the Wiener class of functions. For all $f \in C_{2\pi}$, let

$$a_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta, \quad k = 0, \pm 1, \pm 2, \ldots,$$

be the Fourier coefficients of f. Let A be the n-by-n complex Toeplitz matrix with the (j,k)th entry given by a_{j-k}. The function f is called the generating function of the matrix A.

For a given n-by-n matrix A, we let C be the n-by-n circulant approximation of A as defined in T. Chan [17], i.e., C is the minimizer of $F(X) = \|A - X\|_F$ over all

circulant matrices X. For the special case where A is Toeplitz, the (j,ℓ)th entry of C is given by the diagonal $c_{j-\ell}$ where

$$c_k = \begin{cases} \dfrac{(n-k)a_k + ka_{k-n}}{n} & 0 \le k < n, \\ c_{n+k} & 0 < -k < n. \end{cases} \tag{6}$$

The following three lemmas are proved in R. Chan and Yeung [15]. The first two give the bounds of $||A||_2$ and $||C||_2$ and the last one shows that $A - C$ has a clustered spectrum for certain Toeplitz matrices A.

LEMMA 1. *Let $f \in C_{2\pi}$. Then we have*

$$||A||_2 \le 2||f||_\infty < \infty, \quad n = 1, 2, \ldots. \tag{7}$$

If, moreover, f has no zeros, i.e.,

$$\min_{\theta \in [-\pi,\pi]} |f(\theta)| > 0,$$

then there exists a constant $c > 0$ such that for all n sufficiently large, we have

$$||A||_2 > c. \tag{8}$$

LEMMA 2. *Let $f \in C_{2\pi}$. Then we have*

$$||C||_2 \le 2||f||_\infty < \infty, \quad n = 1, 2, \ldots. \tag{9}$$

If, moreover, f has no zeros, then for all sufficiently large n, we also have

$$||C^{-1}||_2 \le 2 \left\| \frac{1}{f} \right\|_\infty < \infty. \tag{10}$$

LEMMA 3. *Let $f \in C_{2\pi}$. Then for all $\epsilon > 0$, there exist N and $M > 0$, such that for all $n > N$,*

$$A - C = U + V,$$

where

$$\text{rank } U \le M$$

and

$$||V||_2 \le \epsilon.$$

Now let us consider the general least squares problem (3) where T is an m-by-n matrix with $m \ge n$. For the purpose of constructing the preconditioner, we assume that $m = kn$, without loss of generality, since otherwise the final block T_k can be extended to an $n \times n$ Toeplitz matrix by extending the diagonals and padding the lower left part with zeros. (This modification is only for constructing the preconditioner. The original least squares problem (3) is not changed.) Thus we can partition T as (4), without loss of generality. We note that the solution to the least square problem (3) can be obtained by solving the normal equations

$$T^*Tx = T^*b,$$

in factored form, where

$$T^*T = \sum_{j=1}^{k} T_j^* T_j.$$

Of course, one can avoid actually forming T^*T for implementing the CG method for the normal equations [7].

We will assume in the following that k is a constant independent of n and that each square block T_j, $j = 1, \ldots, k$ is generated by a generating function f_j in $\mathcal{C}_{2\pi}$. Following Nagy [26] and Nagy and Plemmons [27], we define a preconditioner for T based upon preconditioners for the blocks T_j.

For each block T_j, let C_j be the corresponding T. Chan circulant preconditioner as defined in (6). Then it is natural to consider the square circulant matrix

$$C^*C = \sum_{j=1}^{k} C_j^* C_j \tag{11}$$

as a circulant approximation to T^*T [27]. Note, however, that C is computed (or applied) using (5). Clearly C is invertible if one of the C_j is. In fact, using Lemma 2, we have the following lemma.

LEMMA 4. *Let $f_j \in \mathcal{C}_{2\pi}$ for $j = 1, 2, \ldots, k$. Then we have*

$$\|C\|_2^2 \leq 4 \sum_{j=1}^{k} \|f_j\|_\infty^2 < \infty, \quad n = 1, 2, \ldots. \tag{12}$$

If, moreover, one of the f_j (say, f_ℓ) has no zeros, then for all sufficiently large n, we also have

$$\|(C^*C)^{-1}\|_2 \leq 4 \left\| \frac{1}{f_\ell} \right\|_\infty^2 < \infty. \tag{13}$$

Proof. Equation (12) clearly follows from (11) and (9). To prove (13), we just note that $C_j^* C_j$ are positive semidefinite matrices for all $j = 1, \ldots, k$; hence

$$\lambda_{\min}(C^*C) \geq \lambda_{\min}(C_\ell^* C_\ell),$$

where $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue. Thus by (10), we then have

$$\|(C^*C)^{-1}\|_2 \leq \|(C_\ell^* C_\ell)^{-1}\|_2 = \|C_\ell^{-1}\|_2^2 \leq 4 \left\| \frac{1}{f_\ell} \right\|_\infty^2. \qquad \square$$

3. Spectrum of TC^{-1}. In this section, we show that the spectrum of the matrix

$$(C^*C)^{-1}(T^*T)$$

is clustered around 1. It will follow then, that the singular values of TC^{-1} are also clustered around 1, since $(C^*C)^{-1}(T^*T)$ is similar to $(TC^{-1})^*(TC^{-1})$. We begin by analyzing the spectrum of each block.

LEMMA 5. *For $1 \leq j \leq k$, if $f_j \in \mathcal{C}_{2\pi}$, then for all $\epsilon > 0$, there exist N_j and $M_j > 0$, such that for all $n > N_j$,*

$$T_j^* T_j - C_j^* C_j = U_j + V_j,$$

where U_j and V_j are Hermitian matrices with

$$\text{rank } U_j \leq M_j$$

and

$$||V_j||_2 \leq \epsilon.$$

Proof. We first note that by Lemma 3, we have for all $\epsilon > 0$, there exist positive integers N_j and M_j such that for all $n > N_j$,

$$T_j - C_j = \tilde{U}_j + \tilde{V}_j,$$

where rank $\tilde{U}_j \leq M_j$ and $||\tilde{V}_j||_2 \leq \epsilon$. Therefore,

$$\begin{aligned} & T_j^* T_j - C_j^* C_j \\ &= T_j^*(T_j - C_j) + (T_j - C_j)^* C_j \\ &= T_j^*(T_j - C_j) - (T_j - C_j)^*(T_j - C_j) + (T_j - C_j)^* T_j \\ &= T_j^*(\tilde{U}_j + \tilde{V}_j) - (\tilde{U}_j + \tilde{V}_j)^*(\tilde{U}_j + \tilde{V}_j) + (\tilde{U}_j + \tilde{V}_j)^* T_j \\ &\equiv U_j + V_j. \end{aligned}$$

Here

$$\begin{aligned} U_j &= T_j^* \tilde{U}_j + \tilde{U}_j^* T_j - \tilde{U}_j^* \tilde{U}_j - \tilde{U}_j^* \tilde{V}_j - \tilde{V}_j^* \tilde{U}_j \\ &= \tilde{U}_j^*(T_j - \tilde{U}_j - \tilde{V}_j) + (T_j - \tilde{V}_j)^* \tilde{U}_j \end{aligned}$$

and

$$V_j = \tilde{V}_j^* T_j + T_j^* \tilde{V}_j - \tilde{V}_j^* \tilde{V}_j.$$

It is clear that both U_j and V_j are Hermitian matrices. Moreover, we have rank $U_j \leq 2M_j$ and

$$||V_j||_2 \leq 2\epsilon ||T_j||_2 + \epsilon^2.$$

By (7), we then have

$$||V_j||_2 \leq 4\epsilon ||f_j||_\infty + 2\epsilon^2. \qquad \square$$

Using the facts that

$$T^*T - C^*C = \sum_{j=1}^{k} (T_j^* T_j - C_j^* C_j)$$

and that k is independent of n, we immediately have the following lemma.

LEMMA 6. *Let $f_j \in \mathcal{C}_{2\pi}$ for $j = 1, \ldots, k$. Then for all $\epsilon > 0$, there exist N and $M > 0$, such that for all $n > N$,*

$$T^*T - C^*C = \tilde{U} + \tilde{V},$$

where $\tilde{U}$ and $\tilde{V}$ are Hermitian matrices with

$$\text{rank } \tilde{U} \leq M \tag{14}$$

and

$$\|\tilde{V}\|_2 \leq \epsilon. \tag{15}$$

We now show that the spectrum of the preconditioned matrix

$$(C^*C)^{-1}(T^*T)$$

is clustered around 1. We note that this is equivalent to showing that the spectrum of $(C^*C)^{-1}(T^*T) - I$, where I is the n-by-n identity matrix, is clustered around zero.

THEOREM 1. *Let $f_j \in \mathcal{C}_{2\pi}$ for all $j = 1, \ldots, k$. If one of the f_j (say, f_ℓ) has no zeros, then for all $\epsilon > 0$, there exist N and $M > 0$, such that for all $n > N$, at most M eigenvalues of the matrix*

$$(C^*C)^{-1}(T^*T) - I$$

have absolute values larger than ϵ.

Proof. By Lemma 6, we have

$$(C^*C)^{-1}(T^*T) - I = (C^*C)^{-1}(T^*T - C^*C) = (C^*C)^{-1}(\tilde{U} + \tilde{V}).$$

Therefore, the spectra of the matrices

$$(C^*C)^{-1}(T^*T) - I \quad \text{and} \quad (C^*C)^{-1/2}(\tilde{U} + \tilde{V})(C^*C)^{-1/2}$$

are the same. However, by (14), we have

$$\operatorname{rank}\left\{(C^*C)^{-1/2}\tilde{U}(C^*C)^{-1/2}\right\} \leq M$$

and by (15) and (13), we have

$$\|(C^*C)^{-1/2}\tilde{V}(C^*C)^{-1/2}\|_2 \leq \|\tilde{V}\|_2\|(C^*C)^{-1}\|_2 \leq 4\hat{\epsilon}\left\|\frac{1}{f_\ell}\right\|_\infty^2,$$

where $\hat{\epsilon}$ replaces the ϵ specified in (15). Thus by applying the Cauchy interlace theorem (see Wilkinson [33]) to the Hermitian matrix

$$(C^*C)^{-1/2}\tilde{U}(C^*C)^{-1/2} + (C^*C)^{-1/2}\tilde{V}(C^*C)^{-1/2},$$

we see that its spectrum is clustered around zero. Hence the spectrum of the matrix $(C^*C)^{-1}(T^*T)$ is clustered around 1. □

From Theorem 1, we have the desired clustering result; namely, if $f_j \in \mathcal{C}_{2\pi}$ for all $j = 1, \ldots, k$ and if one of the f_j has no zeros, then the *singular values of the preconditioned matrix TC^{-1} are clustered around* 1.

4. Convergence rate of the method. In this section, we analyze the convergence rate of Algorithm PCG for Least Squares, for our circulant preconditioned Toeplitz matrix TC^{-1}. We show first that the method converges, for sufficiently large n, in at most $O(\alpha \log n + 1)$ steps where $O(n^\alpha)$ is the condition number of T. We begin by noting the following error estimate of the CG method.

LEMMA 7. *Let G be a positive definite matrix and x be the solution to $Gx = b$. Let x_j be the jth iterant of the ordinary CG method applied to the equation $Gx = b$. If the eigenvalues $\{\delta_k\}$ of G are such that*

$$0 < \delta_1 \leq \cdots \leq \delta_p \leq \gamma_1 \leq \delta_{p+1} \leq \cdots \leq \delta_{n-q} \leq \gamma_2 \leq \delta_{n-q+1} \leq \cdots \leq \delta_n,$$

then

$$\frac{||x - x_j||_G}{||x - x_0||_G} \leq 2\left(\frac{\gamma - 1}{\gamma + 1}\right)^{j-p-q} \cdot \max_{\delta \in [\gamma_1, \gamma_2]} \left\{ \prod_{k=1}^{p} \left(\frac{\delta - \delta_k}{\delta_k} \right) \right\}. \tag{16}$$

Here

$$\gamma \equiv \left(\frac{\gamma_2}{\gamma_1}\right)^{1/2} \geq 1$$

and $||v||_G \equiv v^* G v$.

Proof. It is well known that an error estimate of the CG method is given by the following minimax inequality:

$$\frac{||x - x_j||_G}{||x - x_0||_G} \leq \min_{\mathcal{P}_j} \max_{k=1,\ldots,n} |\mathcal{P}_j(\delta_k)|,$$

where $\mathcal{P}_j$ is any jth degree polynomial with constant term 1 (see Axelsson and Barker [4]). To obtain an upper bound, we first use linear polynomials of the form $(\delta - \delta_k)/\delta_k$ that pass through (i.e., have as roots) the outlying eigenvalues δ_k, $1 \leq k \leq p$ and $n - q + 1 \leq k \leq n$, in order to minimize the maximum absolute value of $\mathcal{P}_j$ at these eigenvalues. These polynomials are thus used as factors of the polynomials being constructed. Next we use a $(j - p - q)$th degree Chebyshev polynomial $\mathcal{T}_{j-p-q}$ to minimize the maximum absolute value of $\mathcal{P}_j$ in the interval $[\delta_{p+1}, \delta_{n-q}]$. Then we get

$$\frac{||x - x_j||_G}{||x - x_0||_G} \leq \mathcal{T}_{j-p-q}\left[\frac{\gamma_2 + \gamma_1}{\gamma_2 - \gamma_1}\right]^{-1} \max_{\delta \in [\gamma_1, \gamma_2]} \left\{ \prod_{k=1}^{p} \left(\frac{\delta - \delta_k}{\delta_k} \right) \prod_{k=n-q+1}^{n} \left(\frac{\delta_k - \delta}{\delta_k} \right) \right\}.$$

Equation (16) now follows by noting that for $\delta \in [\gamma_1, \gamma_2]$, we always have

$$0 \leq \frac{\delta_k - \delta}{\delta_k} \leq 1, \quad n - q + 1 \leq k \leq n$$

and that

$$\mathcal{T}_{j-p-q}\left[\frac{\gamma_2 + \gamma_1}{\gamma_2 - \gamma_1}\right]^{-1} \leq 2\left(\frac{\gamma - 1}{\gamma + 1}\right)^{j-p-q}$$

(see Axelsson and Barker [4]). □

For the system

$$(C^*C)^{-1}(T^*T)x = (C^*C)^{-1}T^*b, \tag{17}$$

the iteration matrix G is given by

$$G = (C^*C)^{-1/2}(T^*T)(C^*C)^{-1/2}.$$

By Theorem 1, we can choose $\gamma_1 = 1 - \epsilon$ and $\gamma_2 = 1 + \epsilon$. Then p and q are constants that depend only on ϵ but not on n. By choosing $\epsilon < 1$, we have

$$\frac{\gamma - 1}{\gamma + 1} = \frac{1 - \sqrt{1 - \epsilon^2}}{\epsilon} < \epsilon.$$

In order to use (16), we need a lower bound for δ_k, $1 \le k \le p$. We first note that

$$||G^{-1}||_2 = ||(T^*T)^{-1}(C^*C)||_2 \le \frac{||C||_2^2}{||T||_2^2}\kappa(T^*T).$$

If one of the f_ℓ has no zeros, then by (8), we have for n sufficiently large

$$||T||_2^2 \ge ||T_\ell||_2^2 \ge c$$

for some $c > 0$ independent of n. Combining this with (12), we then see that for all n sufficiently large,

$$||G^{-1}||_2 \le \tilde{c} \cdot \kappa(T^*T) \le \tilde{c}n^\alpha,$$

for some constant $\tilde{c}$ that does not depend on n. Therefore,

$$\delta_k \ge \min_\ell \delta_\ell = \frac{1}{||G^{-1}||_2} \ge cn^{-\alpha}, \quad 1 \le k \le n.$$

Thus for $1 \le k \le p$ and $\delta \in [1-\epsilon, 1+\epsilon]$, we have

$$0 \le \frac{\delta - \delta_k}{\delta_k} \le cn^\alpha.$$

Hence (16) becomes

$$\frac{||x - x_j||_G}{||x - x_0||_G} < c^p n^{p\alpha} \epsilon^{j-p-q}.$$

Given arbitrary tolerance $\tau > 0$, an upper bound for the number of iterations required to make

$$\frac{||x - x_j||_G}{||x - x_0||_G} < \tau$$

is therefore given by

$$j_0 \equiv p + q - \frac{p \log c + \alpha p \log n - \log \tau}{\log \epsilon} = O(\alpha \log n + 1).$$

Since by using FFTs, the matrix-vector products in Algorithm PCG for Least Squares can be done in $O(m \log n)$ operations for any n-vector v, the cost per iteration of the CG method is of $O(m \log n)$. Thus we conclude that the work of solving (17) to a given accuracy τ is $O(\alpha m \log^2 n + m \log n)$ when $\alpha > 0$ and for sufficiently large n.

The convergence analysis given above can be further strengthened. For T an m-by-n matrix of the form (4) with $m = kn$, let $\lambda_{\min}(T_j^* T_j) = O(n^{-\alpha_j})$ for $j = 1, \ldots, k$. By Lemma 1, we already know that

$$\lambda_{\min}(T_j^* T_j) \le \lambda_{\max}(T_j^* T_j) \le 2||f||_\infty^2,$$

therefore $\alpha_j \ge 0$. By the Cauchy interlace theorem, we see that

$$\lambda_{\min}(T^*T) \ge \sum_{j=1}^{k} \lambda_{\min}(T_j^* T_j) \ge O(n^{-\alpha}),$$

where

$$\alpha = \min_j \alpha_j \geq 0.$$

Therefore,

$$\kappa(T^*T) \leq \frac{\lambda_{\max}(T^*T)}{\lambda_{\min}(T^*T)} \leq O(n^\alpha).$$

In the case when one of the $\alpha_j = 0$, i.e., the block T_j is well conditioned independent of n, we see that the least squares problem is also well conditioned, so that $\kappa(T) = O(1)$.

When at least one $\alpha_j = 0$, i.e., $\kappa(T) = O(1)$, the number of iterations required for convergence is of $O(1)$. Hence the complexity of the algorithm reduces to $O(m \log n)$, for sufficiently large n. We remark that in this case, one can show further that the *method converges superlinearly* for the preconditioned least squares problem due to the clustering of the singular values for sufficiently large n (see R. Chan and Strang [9] or R. Chan [11] for details). In contrast, the method converges just linearly for the nonpreconditioned case. This contrast is illustrated very well in the section on numerical tests.

5. Preconditioned regularized least squares. In this section we consider solving least squares problems (3), where the rectangular matrix T is ill conditioned. Such systems arise in many applications, such as signal and image restoration; see [3], [24], [28]. Often, the ill-conditioned nature of T results from discretization of ill-posed problems in partial differential and integral equations. Here, for example, the problem of estimating an original image from a blurred and noisy observed image is an important case of an *inverse problem* and was first studied by Hadamard [22] in the inversion of certain integral equations. Because of the ill conditioning of T, naively solving $Tx = b$ will lead to extreme instability with respect to perturbations in b. The method of *regularization* can be used to achieve stability for these problems [7]. Stability is attained by introducing a stabilizing operator (called a regularization operator), which restricts the set of admissible solutions. Since this causes the regularized solution to be biased, a scalar (called a regularization parameter) is introduced to control the degree of bias. More specifically, the regularized solution is computed as

$$\min \left\| \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} T \\ \mu L \end{bmatrix} x(\mu) \right\|_2, \tag{18}$$

where μ is the regularization parameter and the $p \times n$ matrix L is the regularization operator.

The standard least squares solution to (3), given by $x = T^\dagger b$, is useless for these problems because it is dominated by rapid oscillations due to the errors. Hence in (18), one adds a term $\min \|Lx\|^2$ to (3) in order to *smooth* the solution x. Choosing L as a kth difference operator matrix forces the solution to have a small kth derivative. The regularization parameter μ controls the degree of smoothness (i.e., degree of bias) of the solution, and is usually small. Choosing μ is not a trivial problem. In some cases a priori information about the signal and the degree of perturbations in b can be used to choose μ [1], or generalized cross-validation techniques may be used, e.g., [7]. If no a priori information is known, then it may be necessary to solve (18) for several

values of μ [20]. Recent analytical methods for choosing an optimal parameter μ are discussed by Reaves and Mersereau [29].

Based on the discussion above, the regularization operator L is usually chosen to be the identity matrix or some discretization of a differentiation operator [6], [20]. Thus L is typically a Toeplitz matrix. Hence, if T has the Toeplitz block form (4), then the matrix

$$\tilde{T} = \begin{bmatrix} T \\ \mu L \end{bmatrix}$$

retains this structure, with the addition of one block (or two blocks if L is a difference operator with more rows than columns). Since $\tilde{T}$ has the block structure (4), we can form the circulant preconditioner C for $\tilde{T}$ and use the PCG algorithm for least squares problems to solve (18).

Notice that if L is chosen to be the identity matrix, then the circulant preconditioner for $\tilde{T}$ can be constructed by simply adding μ to each of the eigenvalues of the circulant preconditioner for T. In addition, the last block in $\tilde{T}$ (i.e., μI) has singular values μ. Thus, due to the remarks at the end of §4, if each block in T is generated by a function in $\mathcal{C}_{2\pi}$, and if $\mu \neq 0$, then $\kappa(\tilde{T}) \leq O(\mu^{-1})$ for all n. It follows then, for these problems, that (18) can be solved in $O(m \log n)$ operations, for sufficiently large n.

6. Numerical tests. In this section we report on some numerical experiments that use the preconditioner C given by (5) in §1 for the CG algorithm PCG for solving Toeplitz and block Toeplitz least squares problems. Here the preconditioner C is based on the T. Chan optimal preconditioner C_i, for each block T_i of T, as in §2. The experiments are designed to illustrate the performance of the preconditioner on a variety of problems, including some in which one or more Toeplitz blocks are very ill conditioned.

We use the stopping criteria $\|s^{(j)}\|_2/\|s^{(0)}\|_2 < 10^{-7}$ for all numerical tests given in this section. (Note that $s^{(j)}$ is the (normal equations) residual after j iterations, and the zero vector is our initial guess. Observe that the value $\|s^{(j)}\|_2$ is computed as part of the CG algorithm.) All experiments were performed using the Pro-Matlab software on our workstations. The machine epsilon for Pro-Matlab on this system is approximately 2.2×10^{-16}.

To describe most of the Toeplitz matrices used in the examples below, we use the following notation. Let the m-vector c be the first column of T and the n-vector r^T be the first row of T. Then

$$T = \text{Toep}(c, r).$$

The right-hand-side vector b is generally chosen to be the vector of all ones.

Example 1. In this example we construct $m \times n$ Toeplitz matrices generated by a positive function in the Wiener class, varying the number of rows and columns and fixing the number of blocks in the block form (4) to $k = 3$. This example is a rectangular generalization of test data used by Strang [30] and is defined as follows. Let

$$c(i) = \frac{1}{2^{i-1}}, \quad i = 1, \ldots, m, \quad \text{and} \quad r(j) = \frac{1}{2^{j-1}}, \quad j = 1, \ldots, n.$$

The convergence results for this example are shown in Table 1, which shows the number of iterations required for Algorithm PCG to converge using T (i.e., no

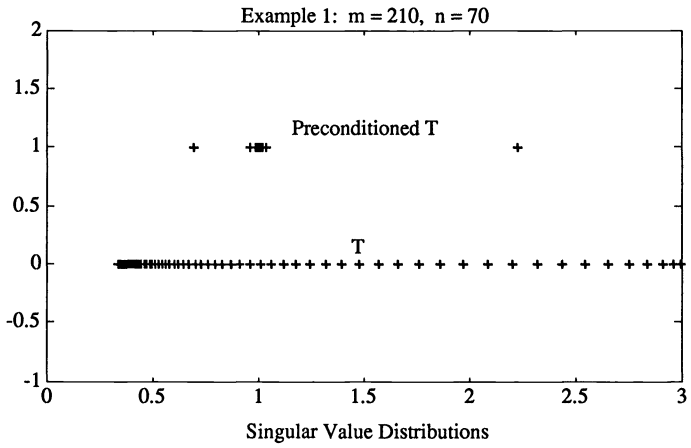


FIG. 1. *Singular values for T and TC^{-1} in Example* 1.

preconditioner) and our C. We can see from Table 1 that the use of our preconditioner does accelerate the convergence rate of the CG algorithm for this problem. Moreover, for this example the number of iterations remains essentially constant as m and n increase.

In Fig. 1 we plot the singular values of T and TC^{-1}. The plot of the singular value distributions shows that the preconditioner clusters the singular values very well for this example.

Example 2. In this example we use the following three generating functions in the Wiener class to construct a $3n \times n$ block Toeplitz matrix.

(i) Example (a) from R. Chan and Yeung [15]:
$c_1(j) = r_1(j) = (|j-1|+1)^{-1.1} + \sqrt{-1}(|j-1|+1)^{-1.1}, \quad j = 1, 2, \ldots, n.$

(ii) Example (b) from R. Chan and Yeung [15]:
$c_2(i) = (|i-1|+1)^{-1.1}, \quad i = 1, 2, \ldots, n,$
$r_2(j) = \sqrt{-1}(|j-1|+1)^{-1.1}, \quad j = 1, 2, \ldots, n.$

(iii) Example (f) from R. Chan and Yeung [15]:
$c_3(1) = r_3(1) = \frac{1}{5}\pi^4$
$c_3(j) = r_3(j) = 4(-1)^{(j-1)}\left(\frac{\pi^2}{(j-1)^2} - \frac{6}{(j-1)^4}\right), \quad j = 2, 3, \ldots, n.$

The matrix T is defined as

$$T^T = [T_1^T,\ T_2^T,\ T_3^T],$$

where T_1 =Toep(c_1, r_1), T_2 =Toep(c_2, r_2), and T_3 =Toep(c_3, r_3). For $n \times n$ systems R. Chan and Yeung [15] show that $\kappa_2(T_3) = O(n^4)$, while T_1 and T_2 are well conditioned. They also show that the T. Chan preconditioner works well for T_1 and T_2, but not well for T_3.

In Table 1 we show the convergence results for this example, using no preconditioner and C as a preconditioner, for several values of m and n. Figure 2 shows the singular values of T and TC^{-1} for $m = 210$ and $n = 70$. These results illustrate the good convergence properties using the preconditioner C for this example containing an ill-conditioned block. Moreover, our computations verify the fact that $\kappa_2(T)$ remains almost constant as n increases from 40 to 80.

TABLE 1
Numbers of iterations for convergence in Examples 1–3.

	Example 1 ($m = 3n$)		Example 2 ($m = 3n$)		Example 3 ($m = 2n$)	
n	no prec.	with prec.	no prec.	with prec.	no prec.	with prec.
40	33	7	96	14	29	11
50	36	7	126	14	33	15
60	41	7	155	13	44	13
70	41	7	167	13	52	12
80	44	7	186	13	65	14

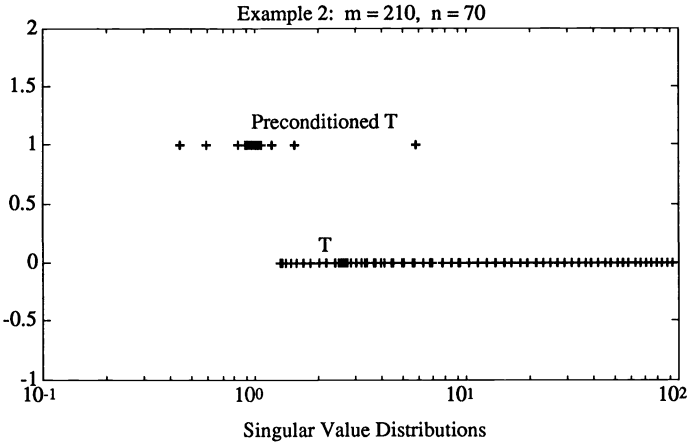


FIG. 2. *Singular values for T and TC^{-1} in Example* 2.

Example 3. In this example we form a $2n \times n$ block Toeplitz matrix using generating functions from R. Chan and Yeung [15] that construct ill-conditioned $n \times n$ Toeplitz matrices. Here $T_1 = T_2$ and thus both blocks of T are ill conditioned. The generating function, which is in the Wiener class, is:

Example (c) from R. Chan and Yeung [15]:

$c_1(1) = r_1(1) = 0$

$c_1(j) = r_1(j) = (|j-1|+1)^{-1.1} + \sqrt{-1}(|j-1|+1)^{-1.1}, \quad j = 2, \ldots, n.$

Using the above generating functions, we let

$$T^T = [T_1,\ T_2]^T,$$

where $T_1 = T_2 = \text{Toep}(c_1, r_1)$.

In Table 1 we show the convergence results for this example, using no preconditioner and C as a preconditioner, for several values of m and n. Figure 3 shows the singular values of T and TC^{-1} for $m = 140$ and $n = 70$. These results illustrate the good convergence properties of C for this example even though it contains all ill-conditioned blocks.

Example 4. Here we consider an application to one-dimensional image or signal-reconstruction computations. In this example we construct the 100×100 Toeplitz matrix T, whose i, j entry is given by

$$t_{ij} = \begin{cases} 0 & \text{if } |i-j| > 8, \\ \frac{4}{51} g(0.15, x_i - x_j) & \text{otherwise,} \end{cases} \tag{19}$$

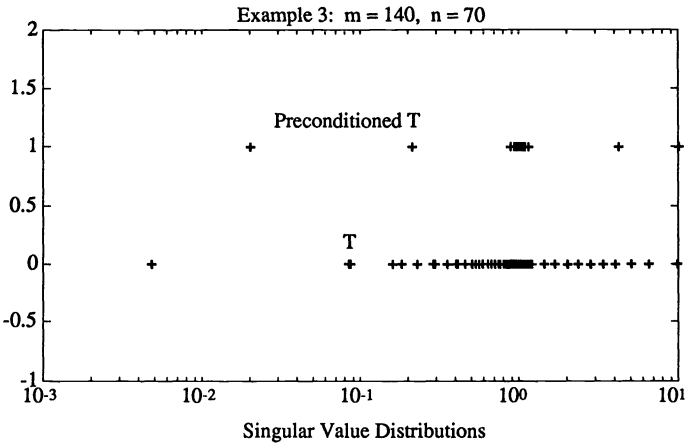


FIG. 3. *Singular values for T and TC^{-1} in Example 3.*

where

$$x_i = \frac{4i}{51}, \quad i = 1, 2, \ldots, 100,$$

and

$$g(\sigma, \gamma) = \frac{1}{2\sqrt{\pi}\sigma} \exp\left(-\frac{\gamma^2}{4\sigma^2}\right).$$

Matrices of this form occur in many image-restoration contexts as a "prototype problem" and are used to model certain degradations in the recorded image [20], [24]. Due to the bandedness of T, its generating function is in the Wiener class. The condition number of T is approximately 2.4×10^6.

Because of the ill conditioning of T, the system $Tx = b$ will be very sensitive to any perturbations in b (see §5). To achieve stability we regularize the problem using the identity matrix as the regularization operator. Eldén [20] uses this approach to solve a linear system by direct methods with the same data matrix T defined in (19). To test our preconditioner we will fix $\mu = 0.01$, where μ is chosen based on some tests made by Eldén.

Let

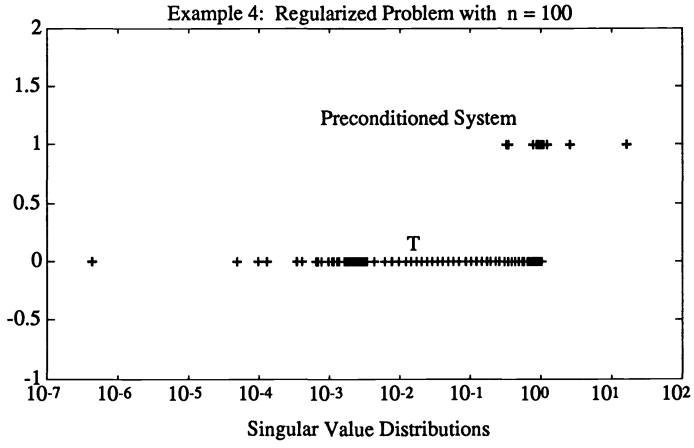
$$\hat{T} = \begin{bmatrix} T \\ \mu I \end{bmatrix} \quad \text{and} \quad \hat{b} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Then $\hat{T}$ is simply a block Toeplitz matrix. Thus we can apply our preconditioner C, and the PCG algorithm, to solve (18). The convergence results for solving $Tx = b$ and $\hat{T}x = \hat{b}$ with no preconditioner and $\hat{T}x = \hat{b}$ using C as a preconditioner are shown in Table 2. The singular values of T and $\hat{T}C^{-1}$ and the convergence history for solving $Tx = b$ and $\hat{T}x = \hat{b}$ using our preconditioner C are shown in Fig. 4. These results indicate that the PCG algorithm with our preconditioner C may be an effective method for solving this regularized least squares problem.

In summary, we have shown how to construct circulant preconditioners for the efficient solution of a wide class of Toeplitz least squares problems. The numerical

TABLE 2
Numbers of iterations for convergence in Example 4.

n	$Tx = b$	$\hat{T}x = \hat{b}$	$\hat{T}C^{-1}x = \hat{b}$
100	> 100	54	14



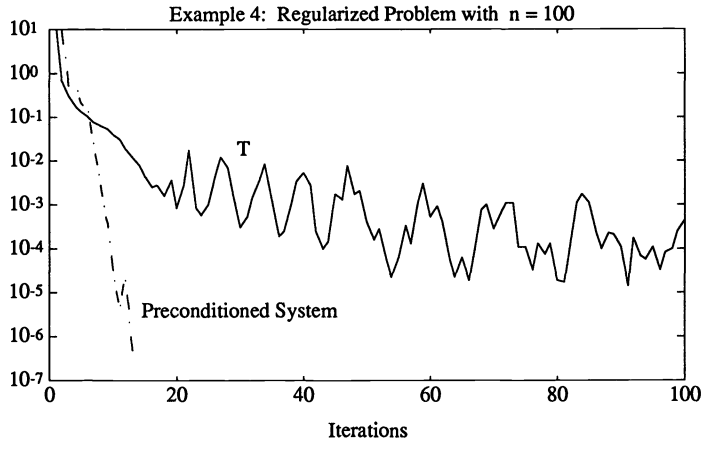


FIG. 4. *Singular values and convergence history for T and $\hat{T}C^{-1}$.*

experiments given collaborate our convergence analysis. Examples 1 and 2 both illustrate superlinear convergence for the PCG algorithm preconditioned by C, even when in Example 2 the matrix T contains an ill-conditioned block. In addition, even though the matrix T in Example 3 contains *all* ill-conditioned blocks, the scheme works well for the computations we performed.

Example 4 illustrates the applicability of the circulant PCG method to regularized least squares problems. The example comes from one-dimensional signal restoration. Two-dimensional signal or image restoration computations often lead to very large least squares problems where the coefficient matrix is block Toeplitz with Toeplitz

blocks. Block circulant preconditioners for this case are considered elsewhere [16].

In this paper we have used the T. Chan [17] optimal preconditioner for the Toeplitz blocks. Other circulant preconditioners such as ones studied by R. Chan [11], Huckle [23], Ku and Kuo [25], Strang [30], Tismenetsky [31], or Tyrtyshnikov [32], can be used, but the class of generating functions may need to be restricted for the convergence analysis to hold.

REFERENCES

[1] J. ABBISS AND P. EARWICKER, *Compact Operator Equations, Regularization and Super-resolution*, in Mathematics in Signal Processing, Clarendon Press, Oxford, 1987.

[2] G. AMMAR AND W. GRAGG, *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

[3] H. ANDREWS AND B. HUNT, *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[4] O. AXELSSON AND V. BARKER, *Finite Element Solution of Boundary Value Problems, Theory and Computation*, Academic Press, London, 1984.

[5] O. AXELSSON AND G. LINDSKOG, *On the rate of convergence of the preconditioned conjugate gradient algorithm*, Numer. Math., 48 (1986), pp. 499–523.

[6] J. BIEDMOND, R. LAGENDIJK, AND R. MESEREAU, *Iterative methods for image deblurring*, Proc. IEEE, 78 (1990), pp. 856–883.

[7] A. BJORCK, *Least Squares Methods*, in Handbook of Numerical Methods, Vol. 1, P. Ciarlet and J. Lions, eds., Elsevier/North Holland, Amsterdam, 1989.

[8] J. BUNCH, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 349–364.

[9] R. CHAN AND G. STRANG, *Toeplitz equations by conjugate gradients with circulant preconditioner*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 104–119.

[10] R. CHAN, *The spectrum of a family of circulant preconditioned Toeplitz systems*, SIAM J. Numer. Anal., 26 (1989), pp. 503–506.

[11] ———, *Circulant preconditioners for Hermitian Toeplitz systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 542–550.

[12] R. CHAN AND X. JIN, *A family of block preconditioners for block systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1218–1235.

[13] R. CHAN AND M. YEUNG, *Circulant preconditioners for Toeplitz matrices with positive continuous generating functions*, Math. Comp., 58 (1992), pp. 233–240.

[14] ———, *Circulant preconditioners constructed from kernels*, SIAM J. Numer. Anal., 29 (1992), pp. 1093–1103.

[15] ———, *Circulant Preconditioners for Complex Toeplitz Matrices*, Research Report 91-6, Dept. of Math., Univ. of Hong Kong, Hong Kong, 1992.

[16] R. CHAN, J. NAGY, AND R. PLEMMONS, *Block Circulant Preconditioners for 2-dimensional Deconvolution Problems*, SPIE Proc., V1770 (1992), pp. 60–71.

[17] T. CHAN, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.

[18] T. CHAN AND J. OLKIN, *Preconditioners for Toeplitz-block Matrices*, preprint, 1991.

[19] P. DAVIS, *Circulant Matrices*, John Wiley & Sons, Inc., New York, 1979.

[20] L. ELDÉN, *An algorithm for the regularization of ill-conditioned, banded least squares problems*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 237–254.

[21] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[22] J. HADAMARD, *Lectures on the Cauchy Problem in Linear Partial Differential Equations*, Yale University Press, New Haven, CT, 1923.

[23] T. HUCKLE, *Circulant and skew-circulant matrices for solving Toeplitz matrix problems*, in Proc. Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, 1990.

[24] A. K. JAIN, *Fundamentals of Digital Image Processing*, Prentice-Hall, Engelwood Cliffs, NJ, 1989.

[25] T. KU AND C. KUO, *Design and analysis of Toeplitz preconditioners*, IEEE Trans. Acoust. Speech Signal Process., V40 (1992), pp. 129–140.

[26] J. NAGY, *Toeplitz Least Squares Computations*, Ph.D. thesis, North Carolina State University, Raleigh, NC, 1991.

[27] J. NAGY AND R. PLEMMONS, *Some fast Toeplitz least squares algorithms*, in Proc. SPIE Con-

ference on Advanced Signal Processing Algorithms, Architectures, and Implementations II, V1566, San Diego, CA, July 1991.

[28] A. OPPENHEIM AND R. SCHAFER, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[29] S. REAVES AND R. MERSEREAU, *Optimal regularization parameter estimation for image reconstruction*, in Proc. SPIE Conference on Image Processing Algorithms and Techniques II, V1452 (1991), pp.127–137.

[30] G. STRANG, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.

[31] M. TISMENETSKY, *A decomposition of Toeplitz matrices and optimal circulant preconditioning,* Linear Algebra Appl., 154–156 (1991), pp. 105–121.

[32] E. TYRTYSHNIKOV, *Optimal and superoptimal circulant preconditioners*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 459–473.

[33] J. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 98–106, January 1994

006

INVERSE OF STRICTLY ULTRAMETRIC MATRICES ARE OF STIELTJES TYPE*

SERVET MARTÍNEZ†, GÉRARD MICHON‡, AND JAIME SAN MARTÍN†

Abstract. This paper shows that a nonnegative ultrametric matrix A is nonsingular and that its inverse is a strictly diagonally dominant Stieltjes matrix. The method consists of studying the spectral decomposition of A by showing that A preserves a maximal filtration.

Key words. ultrametric matrices, Stieltjes matrices

AMS subject classifications. 15A09, 15A18

1. Introduction.

Our main result, which is Theorem 1, deals with the study of properties of strictly ultrametric matrices (see definitions below). We show that these matrices are nonsingular and that their inverses are Stieltjes matrices. In the proof of the main result we show the following properties:

(i) any strictly ultrametric matrix A has an equilibrium potential, i.e., there exists a measure ν such that $A\nu = \mathbf{1}$, the 1-constant vector;

(ii) the matrix Λ given by $\Lambda(i,j) = A(i,j)\mu(j)$, μ being the normalized vector proportional to ν, preserves a maximal filtration of partitions;

(iii) this last result allows us to explicitly obtain the spectral decomposition of Λ, and the monotone properties of its eigenvalues allow us to prove the theorem.

We remark that our results only concern finite ultrametric matrices. But these results can be extended to strictly ultrametric matrices in which an increasing sequence $(h_\alpha : \alpha \geq 1)$, defined analogously as in the proof of Theorem 1, has no finite accumulation point.

Ultrametricity was first introduced in relation with p-adic number theory. In applications like taxonomy [Be, p. 138], ultrametricity is an important notion because of its relation with partitions. On the other hand, strictly ultrametric matrices appear as covariance matrices of random energy models in statistical physics [CCP] as a generalization of the diagonal case. Since most of the relevant quantities depend on the inverse of the covariance matrix, our result concerning the inverse of a strictly ultrametric matrix might be useful in this theory.

Relations between ultrametric matrices and filtrations of partitions (or fields) were first developed by Dellacherie in [De]. A detailed study concerning ultrametric matrices, maximal filtrations, and the associated spectral decomposition for countable probability spaces was made in [DMS]. On the other hand, in [Mi] the study of strictly ultrametric matrices was done in relation with potential theory in compact ultrametric spaces.

Now let us give needed definitions as well as our main result. First, recall that a metric on a set X is said to be ultrametric if it verifies the ultrametric inequality

*Received by the editors July 29, 1991; accepted for publication (in revised form) June 3, 1992. This work was supported by French Cooperation and Fondo Nacional de Ciencia y Tecnologia (FONDECYT) grants 1237-90 and 1208-91.

†Departamento de Ingeniería Matemática, Universidad de Chile, Casilla 170, Correo 3, Santiago 3, Chile (smartine@uchcecum.cec.uchile.cl, jsanmart@uchcecum.cec.uchile.cl).

‡Département de Mathématiques, Faculté de Sciences Mirande, Université de Bourgogne, BP 138-21004 Dijon Cedex, France (topolog@satie.u-bourgogne.fr).

[Di, p. 37]

$$d(x,y) \leq \max\{d(x,z), d(z,y)\} \quad \text{for any } x,y,z \in X.$$

We shall only deal with ultrametric distances on finite sets. Denote by $I := \{1,\dots,N\}$ a finite set.

DEFINITION 1 (see [De]). A symmetric matrix $A = (A(i,j), i,j \in I)$ is said to be ultrametric if there exists an ultrametric distance d on I such that

$$d(i,j) = d(i,k) \quad \text{iff } A(i,j) = A(i,k). \tag{1}$$

DEFINITION 2. A symmetric nonnegative matrix A is said to be strictly ultrametric if there exists an ultrametric distance d on I such that

$$d(i,j) \leq d(i,k) \quad \text{iff } A(i,j) \geq A(i,k). \tag{2}$$

Recall that a symmetric nonnegative matrix A is strictly ultrametric if and only if (iff) it verifies the following two conditions:

$$A(i,j) \geq \inf\{A(i,k), A(k,j)\} \quad \text{for any } i,j,k \in I, \tag{3}$$

$$A(i,i) > \sup\{A(i,k) : k \in I - \{i\}\} \quad \text{for any } i \in I, \tag{4}$$

where, in the case $N = 1$, condition (4) means $A(i,i) > 0$.

In fact, if A is strictly ultrametric, condition (3) follows from property (2). Also (4) is implied by (2) and the strict inequality $d(i,i) < d(i,j)$ for any $i \neq j$. Reciprocally, if (3) and (4) are verified, the following metric d is an ultrametric and it verifies (2): $d(i,i) = 0$ for any $i \in I$ and $d(i,j) = R - A(i,j)$ if $i \neq j$, where $R > \max\{A(i,j) : i \neq j\}$.

DEFINITION 3. A symmetric matrix B is a Stieltjes matrix if its off-diagonal elements are nonpositive, it is nonsingular, and B^{-1} is nonnegative (see [LT, par. 15.2]).

Our main result is the following theorem.

THEOREM 1. *If A is a nonnegative strictly ultrametric matrix, then it is nonsingular, A^{-1} is a strictly diagonally dominant Stieltjes matrix, and $A^{-1}(i,j) = 0$ iff $A(i,j) = 0$ for $i \neq j$.*

2. Proof of the main theorem. Let us first show that a strictly ultrametric matrix has an equilibrium potential.

LEMMA 1. *Let A be a nonnegative, symmetric, strictly ultrametric matrix. Then, there exists a strictly positive vector $\nu = (\nu(i) : i \in I)$, such that*

$$A\nu = \mathbf{1}, \text{ where } \mathbf{1} \text{ is the 1-constant vector.} \tag{5}$$

Proof. Let $q(A) = |\{A(i,j) : i,j \in I\}|$ be the cardinality of the set of values taken by the elements of matrix A. We shall prove the result by induction on $q(A)$.

If $q(A) = 1$, the condition (4) implies $N = 1$. Since $A(N,N) > 0$, the result is evident.

Let $q(A) \geq 2$. Assume we have shown the result for any strictly ultrametric matrix A' with $q(A') < q(A)$.

Let $h := \sup\{A(i,j) : i \neq j\}$. Define the equivalence relation: $i \sim j$ if $i = j$ or $A(i,j) = h$. Denote by $\tilde{i}$ the equivalence class containing i and by $\tilde{I}$ the set of equivalence classes. We remark that if $\tilde{i} \neq \{i\}$, then $A(i,i) > h$.

Condition (3) and the definition of $\tilde{I}$ imply that if $\tilde{i} \neq \tilde{j}$, then $A(i',j') = A(i,j)$ for any $i' \in \tilde{i}, j \in \tilde{j}$. Then, the following matrix $\tilde{A} = (\tilde{A}(\tilde{i},\tilde{j}) : \tilde{i},\tilde{j} \in \tilde{I})$ is well defined:

$$\tilde{A}(\tilde{i},\tilde{j}) = A(i,j) \quad \text{if } \tilde{i} \neq \tilde{j}, \tag{6}$$

$$\tilde{A}(\tilde{i},\tilde{i}) = A(i,i) \quad \text{if } \tilde{i} = \{i\}, \tag{7}$$

$$\tilde{A}(\tilde{i},\tilde{i}) = h + (K(\tilde{i}))^{-1} \quad \text{if } \tilde{i} \neq \{i\}, \tag{8}$$

where

$$K(\tilde{i}) = \sum_{k \in \tilde{i}} (A(k,k) - h)^{-1}. \tag{9}$$

The matrix $\tilde{A}$ is symmetric and its coefficients are nonnegative. Let us show that it satisfies properties (3) and (4).

First, let us prove (4). Take $\tilde{i} \neq \tilde{j}$. Then $\tilde{A}(\tilde{i},\tilde{j}) = A(i,j)$. If $\tilde{i} = \{i\}$, it is evident that $\tilde{A}(\tilde{i},\tilde{i}) > \tilde{A}(\tilde{i},\tilde{j})$. If $\tilde{i} \neq \{i\}$, we have by definition that $\tilde{A}(\tilde{i},\tilde{i}) > h > \tilde{A}(\tilde{i},\tilde{j})$.

Now let us show inequality (3), which is evident in the cases $\tilde{i} = \tilde{j} = \hat{k}$ or $\tilde{i} \neq \tilde{j} \neq \tilde{k} \neq \tilde{i}$. The case $\tilde{i} = \tilde{j} \neq \tilde{k}$ is implied by property (4). If $\tilde{i} = \tilde{k} \neq \tilde{j}$, it is deduced from the equality $A(i,j) = A(j,k)$, and if $\tilde{i} \neq \tilde{j} = \tilde{k}$, it is obtained from $A(i,j) = A(i,k)$.

Thus, the matrix $\tilde{A}$ satisfies properties (3) and (4). On the other hand, $q(\tilde{A}) = |\{\tilde{A}(\tilde{i},\tilde{j}) : \tilde{i},\tilde{j} \in \tilde{I}\}| < q(A)$, so, from induction hypotheses, we deduce that there exists a strictly positive vector $\tilde{\nu} = (\tilde{\nu}(\tilde{i}) : \tilde{i} \in \tilde{I})$ such that $\tilde{A}\tilde{\nu} = \mathbf{1}$. This means that

$$\sum_{\tilde{j} \in \tilde{I}} \tilde{A}(\tilde{i},\tilde{j})\tilde{\nu}(\tilde{j}) = 1 \quad \text{for any } \tilde{i} \in \tilde{I}.$$

Now, define

$$\nu(i) = \tilde{\nu}(\tilde{i}) \quad \text{if } \tilde{i} = \{i\} \tag{10}$$

and

$$\nu(i) = \tilde{\nu}(\tilde{i})(A(i,i) - h)^{-1}(K(\tilde{i}))^{-1} \quad \text{if } \tilde{i} \neq \{i\}. \tag{11}$$

From the definition of $K(\tilde{i})$ given in (9), we get $\sum_{k \in \tilde{i}} \nu(k) = \tilde{\nu}(\tilde{i})$ for any $\tilde{i} \in \tilde{I}$. We have

$$\sum_{j \in I} A(i,j)\nu(j) = \sum_{\tilde{j} \neq \tilde{i}} \sum_{\ell \in \tilde{j}} A(i,\ell)\nu(\ell) + \sum_{k \in \tilde{i}} A(i,k)\nu(k).$$

Since $A(i,\ell) = \tilde{A}(\tilde{i},\tilde{\ell})$ when $\ell \in \tilde{j} \neq \tilde{i}$ and $\tilde{\nu}(\tilde{j}) = \sum_{\ell \in \tilde{j}} \nu(\ell)$, then to complete the induction, it suffices to show the equality

$$\tilde{A}(\tilde{i},\tilde{i})\tilde{\nu}(\tilde{i}) = \sum_{k \in \tilde{i}} A(i,k)\nu(k).$$

If $\tilde{i} = \{i\}$, the equality holds trivially, so assume $\tilde{i} \neq \{i\}$. Since $A(i,k) = h$ for any $k \in \tilde{i} - \{i\}$, we deduce

$$\begin{aligned}\sum_{k\in\tilde{i}} A(i,k)\nu(k) &= \tilde{\nu}(\tilde{i})(K(\tilde{i}))^{-1}\left(\sum_{k\in\tilde{i}-\{i\}} h(A(k,k)-h)^{-1} + A(i,i)(A(i,i)-h)^{-1}\right)\\ &= \nu(\tilde{i})(K(\tilde{i}))^{-1}(hK(\tilde{i}) + (A(i,i)-h)(A(i,i)-h)^{-1})\\ &= \nu(\tilde{i})(h + (K(\tilde{i}))^{-1}) = \tilde{\nu}(\tilde{i})\tilde{A}(\tilde{i},\tilde{i}).\end{aligned}$$

Then the result holds. □

Now, let us show our main result.

Proof of Theorem 1. Let $A = (A(i,j) : i,j \in I)$ be a nonnegative, symmetric, strictly ultrametric matrix. We must show that it is nonsingular and that its inverse $A^{-1} = (A^{-1}(i,j) : i,j \in I)$ satisfies

$$A^{-1}(i,j) \leq 0 \quad \text{for any pair } i \text{ and } j \text{ with } i \neq j; \tag{12}$$

$$A^{-1}(i,i) > \sum_{\substack{j=1\\ j\neq i}}^{N} |A^{-1}(i,j)|, \tag{13}$$

i.e., A^{-1} is strictly diagonally dominant;

$$A^{-1}(i,j) = 0 \text{ iff } A(i,j) = 0 \quad \text{for any pair } i \neq j. \tag{14}$$

From Lemma 1 there exists a strictly positive vector $\nu = (\nu(i) : i \in I)$ verifying (5). Define the probability vector $\mu = (\mu(i) : i \in I)$ by $\mu(i) = \nu(i)(\sum_{j\in I}\nu(j))^{-1}$. We have

$$A\mu = \rho\mathbf{1},$$

where

$$\rho = \left(\sum_{j\in I}\nu(j)\right)^{-1} > 0.$$

Define the following operator Λ acting on $\mathbb{R}^N$:

$$(\Lambda f)(i) = \sum_{j\in I} A(i,j)f(j)\mu(j) \quad \text{for } f \in \mathbb{R}^N. \tag{15}$$

Since A is symmetric, we see that Λ is selfadjoint with respect to the inner product $\langle , \rangle_\mu$ defined by

$$\langle f, g\rangle_\mu = \sum_{i\in I} f(i)g(i)\mu(i). \tag{16}$$

Let us order the set of values $\{A(i,j) : i \neq j\}$ and denote them by $h_1 < \cdots < h_p$. We remark that $0 \leq h_1$. For any $\alpha = 1,\ldots,p$, define the relation $i[\alpha]k$ if $i = k$ or $A(i,k) > h_\alpha$. Property (3) implies that the relation $[\alpha]$ is an equivalence relation.

Denote by $B_\alpha(i) := \{k \in I : i[\alpha]k\}$ the equivalence class of i. The partition constituted by this set of atoms is denoted by $\mathcal{B}_\alpha$, i.e., $\mathcal{B}_\alpha = \{B_\alpha(i)\}$. Observe that $\mathcal{B}_p = \{\{i\} : i \in I\}$ is the finest partition.

Let us introduce the coarsest partition $\mathcal{B}_0 = \{I\}$. Note that $\mathcal{B}_1$ is finer than $\mathcal{B}_0$. The sequence of partitions $\mathcal{B}_0 \subset \mathcal{B}_1 \subset \cdots \subset \mathcal{B}_p$ is strictly increasing (where $\subset$ means being strictly coarser than). Let $i \in I$. The sequence of atoms $(B_\alpha(i) : \alpha = 0, \ldots, p)$ is nondecreasing, i.e., $B_{\alpha-1}(i) \supseteq B_\alpha(i)$ for $\alpha = 1, \ldots, p$. We remark that if $k \in B_{\alpha-1}(i) - B_\alpha(i)$, then $A(i,k) = h_\alpha$. For $k \neq i$ denote by $T(i,k)$ the unique index $\alpha \in \{0, \ldots, p\}$, such that $k \in B_{\alpha-1}(i) - B_\alpha(i)$ (it does exist because $B_p(i) = \{i\}$). Then $A(i,k) = h_{T(i,k)}$. Thus, we deduce

$$k \in B_{\alpha-1}(i) \text{ implies } A(i,k) = h_{T(i,k)} \geq h_\alpha \tag{17}$$

and

$$k \notin B_\alpha(i) \text{ implies } A(i,k) = h_{T(i,k)} \leq h_\alpha. \tag{18}$$

Let us point out that (4) implies that $A(i,i) > h_{T(i,k)}$ for any $k \neq i$. On the other hand from property (3), we also get that if $B_\alpha(j) = B_\alpha(\ell) \neq B_\alpha(i)$, then $T(j,k) = T(\ell,k) \leq \alpha$ for any $k \in B_\alpha(i)$, so

$$A(j,k) = A(\ell,k) \leq h_\alpha.$$

Now we shall identify a partition $\mathcal{B}$ of I with the field it generates. Since I is finite, the field generated by $\mathcal{B}$ is $\{\bigcup_{B \in J} B : J \subset \mathcal{B}\}$. We remark that a function $f \in \mathbb{R}^I$ is $\mathcal{B}$-measurable iff $f(i) = f(j)$ for any i, j belonging to the same atom of $\mathcal{B}$. We say that an operator acting on $\mathbb{R}^I$ preserves $\mathcal{B}$ if it preserves the $\mathcal{B}$-measurable functions.

Let us show that the operator Λ defined in (15) preserves the filtration of fields $\{\mathcal{B}_\alpha : \alpha = 0, \ldots, p\}$. This means that for all $\alpha = 0, \ldots, p$, for all $f \in \mathbb{R}^I$ $\mathcal{B}_\alpha$-measurable, the function Λf is also $\mathcal{B}_\alpha$-measurable. For $\alpha = 0$ this is satisfied because a $\mathcal{B}_0$-measurable function is a constant $c\mathbf{1}$, and $\Lambda(c\mathbf{1}) = cA\mu = \rho c\mathbf{1}$. For $\alpha = p$ the property is also verified because $\mathcal{B}_p$ is the finest field. Now let us show the result for $\alpha \in \{1, \ldots, p-1\}$. We remark that it suffices to prove that $\Lambda \mathbf{1}_{B_\alpha(i)}$ is $\mathcal{B}_\alpha$-measurable for any $i \in I$, where $\mathbf{1}_{B_\alpha(i)}$ is the characteristic function of the atom $B_\alpha(i) \in \mathcal{B}_\alpha$.

Then, we must show that if

$$B_\alpha(j) = B_\alpha(\ell), \quad \text{then } (\Lambda \mathbf{1}_{B_\alpha(i)})(j) = (\Lambda \mathbf{1}_{B_\alpha(i)})(\ell). \tag{19}$$

First assume that $B_\alpha(j) = B_\alpha(\ell) \neq B_\alpha(i)$. Then $A(j,k) = A(\ell,k)$ for any $k \in B_\alpha(i)$. Thus,

$$(\Lambda \mathbf{1}_{B_\alpha(i)})(j) = \sum_{k \in B_\alpha(i)} A(j,k)\mu(k) = (\Lambda \mathbf{1}_{B_\alpha(i)})(\ell).$$

Now assume $B_\alpha(j) = B_\alpha(\ell) = B_\alpha(i)$. Since $(\Lambda \mathbf{1}_B)(j) = (\Lambda \mathbf{1}_B)(\ell)$ for any atom $B \in \mathcal{B}_\alpha$ such that $B \neq B_\alpha(i)$, and $(\Lambda \mathbf{1})(j) = (\Lambda \mathbf{1})(\ell)$ (because $\Lambda \mathbf{1}$ is constant), we deduce

$$(\Lambda \mathbf{1}_{B_\alpha}(i))(j) = (\Lambda \mathbf{1})(j) - \sum_{B \in \mathcal{B}_\alpha - \{B_\alpha(i)\}} (\Lambda \mathbf{1}_B)(j) = (\Lambda \mathbf{1}_{B_\alpha(i)})(\ell).$$

Then Λ preserves any $\mathcal{B}_\alpha$ for $\alpha = 0, \ldots, p$.

Now the filtration of fields $\mathcal{B}_0 \subset \cdots \subset \mathcal{B}_p$ is contained in a maximal filtration of fields $\mathcal{C}_0 \subset \cdots \subset \mathcal{C}_{N-1}$ (which is not necessarily unique). This means that $\{\mathcal{B}_\alpha : \alpha = 0, \ldots, p\} \subset \{\mathcal{C}_\gamma : \gamma = 0, \ldots, N-1\}$ and that if $\mathcal{C}$ is any field satisfying $\mathcal{C}_\gamma \subseteq \mathcal{C} \subseteq \mathcal{C}_{\gamma+1}$, then $\mathcal{C} = \mathcal{C}_\gamma$ or $\mathcal{C} = \mathcal{C}_{\gamma+1}$. This implies that $\mathcal{C}_{\gamma+1}$ is formed from $\mathcal{C}_\gamma$ by splitting a unique atom into two new atoms. This is why maximal filtrations of fields are also of cardinality N (the cardinality of the set I).

Recall that for any $\gamma \in \{0, \ldots, N-2\}$ there exists $\alpha(\gamma) \in \{1, \ldots, p\}$, such that $\mathcal{B}_{\alpha(\gamma)-1} \subseteq \mathcal{C}_\gamma \subset \mathcal{B}_{\alpha(\gamma)}$.

In the rest of this paper, $C_\gamma(i)$ shall denote the atom of $\mathcal{C}_\gamma$ containing i.

Now let us show that the operator Λ preserves the maximal filtration of fields $\{\mathcal{C}_\gamma : \gamma = 0, \ldots, N-1\}$. This is evident for $\mathcal{C}_0 = \mathcal{B}_0 = \{I\}$ and also for $\mathcal{C}_{N-1} = \mathcal{B}_p = \{\{i\} : i \in I\}$. So fix $\gamma \in \{1, \ldots, N-2\}$. Denote by $\alpha(\gamma) \in \{1, \ldots, p\}$ the point verifying $\mathcal{B}_{\alpha(\gamma)-1} \subseteq \mathcal{C}_\gamma \subset \mathcal{B}_{\alpha(\gamma)}$. We must prove that if $C_\gamma(j) = C_\gamma(\ell)$, then $(\Lambda \mathbf{1}_{C_\gamma(i)})(j) = (\Lambda \mathbf{1}_{C_\gamma(i)})(\ell)$ for any $C_\gamma(i) \in \mathcal{C}_\gamma$.

First, assume $C_\gamma(i) \neq C_\gamma(j) = C_\gamma(\ell)$. Then $B_{\alpha(\gamma)-1}(j) = B_{\alpha(\gamma)-1}(\ell)$ and $B_{\alpha(\gamma)}(j) \neq B_{\alpha(\gamma)}(i) \neq B_{\alpha(\gamma)}(\ell)$. If $B_{\alpha(\gamma)-1}(j) = B_{\alpha(\gamma)-1}(\ell) \neq B_{\alpha(\gamma)-1}(i)$, then $A(j,k) = A(\ell,k)$ for any $k \in B_{\alpha(\gamma)-1}(i) \supset C_\gamma(i)$. Thus

$$(\Lambda \mathbf{1}_{C_\gamma(i)})(j) = \sum_{k \in C_\gamma(i)} A(j,k)\mu(k) = (\Lambda \mathbf{1}_{C_\gamma(i)})(\ell).$$

Now, assume $B_{\alpha(\gamma)-1}(j) = B_{\alpha(\gamma)-1}(\ell) = B_{\alpha(\gamma)-1}(i)$. For any $k \in C_\gamma(i) \subset B_{\alpha(\gamma)-1}(i)$ we have $B_{\alpha(\gamma)}(j) \neq B_{\alpha(\gamma)}(k) \neq B_{\alpha(\gamma)}(\ell)$ and so $j, \ell \in B_{\alpha(\gamma)-1}(k) - B_{\alpha(\gamma)}(k)$. Then $A(j,k) = A(\ell,k)$, which implies $(\Lambda \mathbf{1}_{C_\gamma(i)})(j) = (\Lambda \mathbf{1}_{C_\gamma(i)})(\ell)$.

Now if j, ℓ are such that $C_\gamma(j) = C_\gamma(\ell) = C_\gamma(i)$, then the result follows from the equalities

$$(\Lambda \mathbf{1}_{C_\gamma(i)})(j) = (\Lambda \mathbf{1})(j) - \sum_{C \in \mathcal{C}_\gamma - \{C_\gamma(i)\}} (\Lambda \mathbf{1}_C)(j), \ (\Lambda \mathbf{1})(j) = (\Lambda \mathbf{1})(\ell),$$

and $\Lambda \mathbf{1}_C(j) = \Lambda \mathbf{1}_C(\ell)$, for $C \in \mathcal{C}_\gamma - \{C_\gamma(i)\}$. Thus Λ preserves $\{\mathcal{C}_\gamma : \gamma = 0, \ldots, N-1\}$.

Let us denote by $E_\gamma = E_\mu^{\mathcal{C}_\gamma}$ the mean expected value operator with respect to the field $\mathcal{C}_\gamma$ and the measure μ, i.e.,

$$(E_\gamma f)(i) = (\mu(C_\gamma(i)))^{-1} \sum_{k \in C_\gamma(i)} f(k)\mu(k).$$

Since Λ preserves $\mathcal{C}_\gamma$, we have $\Lambda E_\gamma = E_\gamma \Lambda E_\gamma$, and since Λ is selfadjoint, we also have $\Lambda(E_\gamma - E_{\gamma-1}) = (E_\gamma - E_{\gamma-1})\Lambda(E_\gamma - E_{\gamma-1})$ for any $\gamma \in \{0, \ldots, N-1\}$, with the convention $E_{-1} \equiv 0$. Since $(E_\gamma - E_{\gamma-1})\mathbb{R}^N$ is a one-dimensional subspace, we conclude that $(\Lambda(E_\gamma - E_{\gamma-1}))f = \rho_\gamma(E_\gamma - E_{\gamma-1})f$ for certain $\rho_\gamma \in \mathbb{R}$, for any $\gamma \in \{0, \ldots, N-1\}$. Then the spectral decomposition of Λ is

$$\Lambda = \sum_{\gamma=0}^{N-1} \rho_\gamma (E_\gamma - E_{\gamma-1}). \tag{20}$$

Let us compute the eigenvalues $\{\rho_\gamma : \gamma = 0, \ldots, N-1\}$ of the operator Λ. Since $A\mu = \Lambda \mathbf{1} = \rho_0 \mathbf{1}$, we get

$$\rho_0 = \left(\sum_{i \in I} \nu(i) \right)^{-1}, \tag{21}$$

where ν is the vector given by Lemma 1. An eigenvector associated to ρ_0 is $f_0 = 1$.

Now let us obtain the other values ρ_γ for $\gamma \in \{1, \dots, N-1\}$. Recall that the atoms of $\mathcal{C}_{\gamma-1}$ are the same as the atoms of $\mathcal{C}_\gamma$, except that one atom of $\mathcal{C}_{\gamma-1}$ has been split into two new atoms of $\mathcal{C}_\gamma$, which is denoted by $C_\gamma(i)$ and $C_\gamma(i')$. These last two atoms are disjoint and $C_{\gamma-1}(i) = C_{\gamma-1}(i') = C_\gamma(i) \cup C_\gamma(i')$. Take

$$f_\gamma = \mathbf{1}_{C_\gamma(i)} - \mu(C_\gamma(i))(\mu(C_{\gamma-1}(i)))^{-1}\mathbf{1}_{C_{\gamma-1}(i)}. \tag{22}$$

Then

$$(E_\gamma - E_{\gamma-1})f_\gamma = f_\gamma$$

and

$$\begin{aligned}\Lambda f_\gamma &= \Lambda \mathbf{1}_{C_\gamma(i)} - \mu(C_\gamma(i))(\mu(C_{\gamma-1}(i)))^{-1}\Lambda \mathbf{1}_{C_{\gamma-1}(i)} \\ &= \rho_\gamma\{\mathbf{1}_{C_\gamma(i)} - \mu(C_\gamma(i))(\mu(C_{\gamma-1}(i)))^{-1}\mathbf{1}_{C_{\gamma-1}(i)}\}.\end{aligned}$$

If we evaluate both functions at the point $i' \in C_{\gamma-1}(i) - C_\gamma(i)$, we get

$$\rho_\gamma = (\Lambda \mathbf{1}_{C_{\gamma-1}(i')})(i') - \mu(C_{\gamma-1}(i))(\mu(C_\gamma(i)))^{-1}(\Lambda \mathbf{1}_{C_\gamma(i)})(i').$$

For any $\gamma \geq 1$ denote by $\bar{\alpha}(\gamma)$ the point of $\{1, \dots, p\}$ satisfying $\mathcal{B}_{\bar{\alpha}(\gamma)-1} \subset \mathcal{C}_\gamma \subseteq \mathcal{B}_{\bar{\alpha}(\gamma)}$. Since $i' \in C_\gamma(i') = C_{\gamma-1}(i) - C_\gamma(i)$, for any $k \in C_\gamma(i)$ we have $i' \in B_{\bar{\alpha}(\gamma)-1}(k) - B_{\bar{\alpha}(\gamma)}(k)$. Then $A(i', k) = h_{\bar{\alpha}(\gamma)}$, and we find

$$\rho_\gamma = h_{\bar{\alpha}(\gamma)}\mu(C_\gamma(i)) + \sum_{\ell \in C_\gamma(i')} A(i', \ell)\mu(\ell) - \mu(C_{\gamma-1}(i))\mu(C_\gamma(i))^{-1}h_{\bar{\alpha}(\gamma)}\mu(C_\gamma(i)).$$

So,

$$\rho_\gamma = \sum_{\ell \in C_\gamma(i')} (A(i', \ell) - h_{\bar{\alpha}(\gamma)})\mu(\ell) = \rho_0 \left\{ \sum_{\ell \in C_\gamma(i')} (A(i', \ell) - h_{\bar{\alpha}(\gamma)})\nu(\gamma) \right\}. \tag{23}$$

Now, for $\ell \in C_\gamma(i') \subset B_{\bar{\alpha}(\gamma)-1}(i')$, inequality (14) implies $A(i', \ell) \geq h_{\bar{\alpha}(\gamma)}$. On the other hand, $A(i', i') > h_{\bar{\alpha}(\gamma)}$; hence $\rho_\gamma > 0$ for any $\gamma \in \{0, \dots, N-1\}$. Note that $C_0(i') = I$. Putting $\bar{\alpha}(0) = 0, h_0 = 0$, it is easy to verify that formula (23) also holds for $\gamma = 0$ because $A\nu = \mathbf{1}$.

Since $\rho_\gamma > 0$ for any $\gamma = 1, \dots, N-1$, we deduce that Λ is nonsingular; so, from (20),

$$\Lambda^{-1} = \sum_{\gamma=0}^{N-1} \rho_\gamma^{-1}(E_\gamma - E_{\gamma-1}).$$

Now, let G be a diagonal matrix with $G(j, j) = \mu(j)$ for any $j \in I$. From the definition of Λ given in (15), we get $\Lambda = AG$, so $A^{-1} = G\Lambda^{-1}$. Let $A^{-1}(i, j)$ and $\Lambda^{-1}(i, j)$ denote the (i, j) coefficients of the matrices A^{-1} and Λ^{-1}, respectively. Then $A^{-1}(i, j) = \mu(i)\Lambda^{-1}(i, j)$.

We have $A^{-1}\mathbf{1} = G(\Lambda^{-1}\mathbf{1}) = \rho_0^{-1}\mu$. Thus

$$\sum_{j \in I} A^{-1}(i, j) = \rho_0^{-1}\mu(i) > 0 \quad \text{for any } i \in I. \tag{24}$$

We shall prove now that (12) is satisfied. We have

$$A^{-1}(i,j) = \Lambda^{-1}(i,j)\mu(i) =< \Lambda^{-1}\mathbf{1}_{\{j\}}, \mathbf{1}_{\{i\}} >_\mu = \sum_{\gamma=0}^{N-1} \rho_\gamma^{-1} < (E_\gamma - E_{\gamma-1})\mathbf{1}_{\{j\}}, \mathbf{1}_{\{i\}} >_\mu .$$

Now, $(E_0 - E_{-1})\mathbf{1}_{\{j\}} = E_0\mathbf{1}_{\{j\}} = \mu(j)$, and for any $\gamma \in \{1, \dots, N-1\}$

$$(E_\gamma - E_{\gamma-1})\mathbf{1}_{\{j\}} = \mu(j)\{(\mu(C_\gamma(j)))^{-1}\mathbf{1}_{C_\gamma(j)} - (\mu(C_{\gamma-1}(j)))^{-1}\mathbf{1}_{C_{\gamma-1}}(j)\}.$$

Note that if we set $C_{-1}(j) = \phi$ for any $j \in I$, then the last formula also holds for $\gamma = 0$.

If $C_\gamma(j) = C_{\gamma-1}(j)$, we have $(E_\gamma - E_{\gamma-1})\mathbf{1}_{\{j\}} = 0$. When $i \notin C_{\gamma-1}(j)$, we also have $i \notin C_\gamma(j)$, so $< (E_\gamma - E_{\gamma-1})\mathbf{1}_{\{j\}}, \mathbf{1}_{\{i\}} >_\mu = 0$. Define the set

$$J(i,j) := \{0\} \cup \{\gamma \in \{1, \dots, N-1\} : C_\gamma(j) \neq C_{\gamma-1}(j) \quad \text{and} \quad i \in C_{\gamma-1}(j)\},$$

and denote its elements by $\gamma_0 = 0 < \gamma_1 < \cdots < \gamma_m$. We remark that $m \geq 1$. Set $\gamma_{-1} := -1$. Then from the definition of $J(i,j)$, we have that the equality $C_{\gamma_t-1}(j) = C_{\gamma_{(t-1)}}(j)$ holds for any $t = 0, \dots, m$. Hence

$$\begin{aligned} A^{-1}(i,j) &= \sum_{\gamma \in J(i,j)} \rho_\gamma^{-1} < (E_\gamma - E_{\gamma-1})\mathbf{1}_{\{j\}}, \mathbf{1}_{\{i\}} >_\mu \\ &= \mu(i)\mu(j)\Bigg\{\sum_{t=0}^{m-1} \rho_{\gamma_t}^{-1}[(\mu(C_{\gamma_t}(j)))^{-1} - (\mu(C_{\gamma_t-1}(j)))^{-1}] \\ &\qquad - \rho_{\gamma_m}^{-1}(\mu(C_{\gamma_m-1}(j)))^{-1}\Bigg\}. \end{aligned}$$

Therefore,

$$A^{-1}(i,j) = \mu(i)\mu(j)\sum_{t=0}^{m-1}(\mu(C_{\gamma_t}(j)))^{-1}(\rho_{\gamma_t}^{-1} - \rho_{\gamma_{t+1}}^{-1}). \tag{25}$$

From (20) we get that for any $t = 0, \dots, m-1$, we have

$$\rho_{\gamma_t} = \sum_{\ell \in C_{\gamma_t}(j)} (A(j,\ell) - h_{\bar{\alpha}(\gamma_t)})\mu(\ell). \tag{26}$$

For any $\ell \in C_{\gamma_t}(j)$, we have $A(j,\ell) - h_{\bar{\alpha}(\gamma_t)} \geq 0$. On the other hand, since $C_{\gamma_t}(j)$ is decreasing with t and $h_{\bar{\alpha}(\gamma_t)}$ is nondecreasing with t (in fact, strictly increasing, except perhaps at $t = 0$ when $\gamma_1 = 1$ since $h_1 = h_0 = 0$), we deduce that ρ_{γ_t} is nonincreasing: $\rho_0 = \rho_{\gamma_0} \geq \rho_{\gamma_1} \geq \cdots \geq \rho_{\gamma_m} > 0$. Hence $\rho_{\gamma_t}^{-1}$ is a nondecreasing sequence, and we get $(A^{-1})(i,j) \leq 0$. Now condition (24) implies $A^{-1}(i,i) > 0$ and $A^{-1}(i,i) > \sum_{j \neq i} |A^{-1}(i,j)|$ for any $i \in I$, so property (13) also holds. Let us now prove (14): $A^{-1}(i,j) = 0$ iff $A(i,j) = 0$ for $i \neq j$.

We have $i \in C_{\gamma_m-1}(j) - C_{\gamma_m}(j)$. Note that $B_{\bar{\alpha}(\gamma_m)-1}(j) \supset C_{\gamma_m}(j) \supseteq B_{\bar{\alpha}(\gamma_m)}(j)$, so $B_{\bar{\alpha}(\gamma_m)-1}(j) \supseteq C_{\gamma_m-1}(j) \supset C_{\gamma_m}(j) \supseteq B_{\bar{\alpha}(\gamma_m)}(j)$. Then $i \in B_{\bar{\alpha}(\gamma_m)-1}(j) - B_{\bar{\alpha}(\gamma_m)}(j)$. We remark that $T(i,j) = \bar{\alpha}(\gamma_m)$, so $A(i,j) = h_{\bar{\alpha}(\gamma_m)}$.

If $A(i,j) > 0$, then $h_{\bar{\alpha}(\gamma_m)} > h_{\bar{\alpha}(0)} = 0$, so there must exist an index $t \in \{0,\dots,m-1\}$ such that $h_{\bar{\alpha}(\gamma_t)} < h_{\bar{\alpha}(\gamma_{t+1})}$, which implies $\rho_{\gamma_t} > \rho_{\gamma_{t+1}}$ (see formula (26)). Equality (25) implies $A^{-1}(i,j) < 0$.

If $A(i,j) = 0$, then $m = \gamma_m = \bar{\alpha}(\gamma_m) = 1$ and $h_1 = 0$. Now, for any $\ell \in C_0(j) - C_1(j)$, we have $T(j,\ell) = 1$, which implies that $A(j,\ell) = 0$. Then

$$\begin{aligned}\rho_{\gamma_1} &= \sum_{\ell \in C_1(j)} (A(j,\ell) - h_1)\mu(\ell) = \sum_{\ell \in C_1(j)} A(j,\ell)\mu(\ell) \\ &= \sum_{\ell \in C_0(j)} A(j,\ell)\mu(\ell) = \sum_{\ell \in I} A(j,\ell)\mu(\ell) = \rho_0.\end{aligned}$$

From equality (25), we conclude $A^{-1}(i,j) = 0$. □

Remark. The proof also shows that any strictly ultrametric matrix is positive definite, but this is a general property of any Stieltjes matrix [LT, p. 532].

REFERENCES

[Be] J.P. BENZECRI ET COLLABORATEURS, *L'Analyse des données*, 1 *La Taxinomie*, Dunod, Paris, 1973.

[CCP] D. CAPOCACIA, M. CASSANDRO, AND P. PICCO, *On the existence of thermodynamics for the generalized random energy model*, J. Statist. Physics, 46 (1987), pp. 493–505.

[De] C. DELLACHERIE, *private communication*, 1985.

[Di] J. DIEUDONNÉ, *Fondements de l'analyse moderne*, Gauthier–Villars, Paris, 1968.

[DMS] P. DARTNELL, S. MARTÍNEZ, AND J. SAN MARTÍN, *Opérateurs filtrés et chaînes de tribus invariantes sur un espace probabilisé dénombrable*, Lecture Notes in Math., 1321 (1988), pp. 197–213.

[LT] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, 2nd ed., Academic Press, New York, 1985.

[Mi] G. MICHON, *Arbre, Cantor, Dimension*, Thesis, Université de Bourgogne, 1989.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 107–113, January 1994

007

A LINEAR ALGEBRA PROOF THAT THE INVERSE OF A STRICTLY ULTRAMETRIC MATRIX IS A STRICTLY DIAGONALLY DOMINANT STIELTJES MATRIX*

REINHARD NABBEN† AND RICHARD S. VARGA‡

Abstract. It is well known that every $n \times n$ Stieltjes matrix has an inverse that is an $n \times n$ nonsingular symmetric matrix with nonnegative entries, and it is also easily seen that the converse of this statement fails in general to be true for $n > 2$. In the preceding paper by Martínez, Michon, and San Martín [*SIAM J. Matrix Anal. Appl.*, 15 (1994), pp. 98–106], such a converse result is in fact shown to be true for the new class of *strictly ultrametric matrices*. A simpler proof of this basic result is given here, using more familiar tools from linear algebra.

Key words. Stieltjes matrices, ultrametric matrices, inverse M-matrix problem

AMS subject classifications. 15A57, 15A48

1. Introduction.

It is well known (cf. [3, p. 85]) that a *Stieltjes matrix* $A = [a_{i,j}]$ in $\mathbb{R}^{n,n}$, which is defined to be a real symmetric and positive definite matrix with $a_{i,j} \leq 0$ for all $i \neq j$ $(1 \leq i, j \leq n)$, has the property that its inverse is a real nonsingular and symmetric matrix, all of whose entries are nonnegative. Now, the converse of this result is not generally true for any $n \geq 3$, as the following simple matrix below shows. For $n = 3$, define the symmetric matrix B in $\mathbb{R}^{3,3}$ by

$$B := \begin{bmatrix} 4 & 0 & 2 \\ 0 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix},$$

so that B possesses only nonnegative entries. As the eigenvalues of B are $(4 + \sqrt{13}, 4, 4 - \sqrt{13})$, then B is positive definite. But its inverse,

$$B^{-1} = \frac{1}{12} \begin{bmatrix} 7 & 6 & -8 \\ 6 & 12 & -12 \\ -8 & -12 & 16 \end{bmatrix},$$

fails to be a Stieltjes matrix since its off-diagonal entries are not all nonpositive. For $n > 3$, the matrix

$$\begin{bmatrix} B & O \\ O & I_{n-3} \end{bmatrix} \quad \text{and its inverse} \quad \begin{bmatrix} B^{-1} & O \\ O & I_{n-3} \end{bmatrix},$$

where I_{n-3} is the identity matrix in $\mathbb{R}^{n-3,n-3}$, similarly furnishes a counterexample in $\mathbb{R}^{n,n}$.

In the preceding paper [2, Thm. 1] by Martínez, Michon, and San Martín, it is shown that a strictly ultrametric square matrix (to be defined below) is a nonsingular

* Received by the editors March 24, 1992; accepted for publication (in revised form) April 8, 1992.

† Fakultät für Mathematik, Universität Bielefeld, Postfach 10 01 31, 33 501 Bielefeld, Germany (na.nabben@na-net.ornl.gov). This work was performed while the author was at the Institute for Computational Mathematics, Kent State University, Kent, Ohio 44242. This author's research was supported by the Deutsche Forschungsgemeinschaft.

‡ Institute for Computational Mathematics, Kent State University, Kent, Ohio 44242 (varga@mcs.kent.edu). The research of this author was supported by the National Science Foundation.

matrix, with nonnegative entries, whose inverse is a strictly diagonally dominant Stieltjes matrix! As can be seen from their paper, their interesting result is proved by using a variety of impressive tools from topology and real analysis, tools that may prove useful for infinite dimensional extensions. The beauty of their result gave us the stimulus to try to find a proof of their result that was fashioned solely from more familiar tools from linear algebra, as such a proof might be more accessible to numerical analysts and linear algebraists. We give such a linear algebra proof below.

With the notation that $N := \{1, 2, \ldots, n\}$ for any positive integer n, we begin with the following definition of [2].

DEFINITION 1.1. *A matrix $A = [a_{i,j}]$ in $\mathbb{R}^{n,n}$ is strictly ultrametric if*

$$\text{(1.1)} \qquad \begin{cases} \text{(i)} & A \text{ is symmetric with nonnegative entries,} \\ \text{(ii)} & a_{i,j} \geq \min\{a_{i,k}; a_{k,j}\} \text{ for all } i, j, k \in N, \\ \text{(iii)} & a_{i,i} > \max\{a_{i,k} : k \in N \backslash \{i\}\} \text{ for all } i \in N, \end{cases}$$

where, if $n = 1$, (1.1)(iii) is interpreted as $a_{1,1} > 0$.

The result of [2, Thm. 1] is stated in the following theorem.

THEOREM 1.2. *If $A = [a_{i,j}]$ in $\mathbb{R}^{n,n}$ is strictly ultrametric, then A is nonsingular and its inverse, $A^{-1} := [\alpha_{i,j}]$ in $\mathbb{R}^{n,n}$, is a strictly diagonally dominant Stieltjes matrix (i.e., $\alpha_{i,j} \leq 0$ for all $i \neq j$ and $\alpha_{i,i} > \sum_{\substack{k=1 \\ k \neq i}}^{n} |\alpha_{i,k}|$, for all $1 \leq i, j \leq n$), with the additional property that*

$$\text{(1.2)} \qquad \alpha_{i,j} = 0 \quad \text{if and only if} \quad a_{i,j} = 0.$$

Our proof of Theorem 1.2 is given in §3, after some necessary constructions are given in §2.

2. Some constructions. For notation, on setting $\xi_n := (1, 1, \ldots, 1)^T$ in $\mathbb{R}^n$, then

$$\text{(2.1)} \qquad \xi_n \xi_n^T$$

is a rank-one matrix in $\mathbb{R}^{n,n}$, all of whose entries are unity.

Our first result, which is independent of results or techniques in [2], is necessary for our complete characterization of strictly ultrametric matrices.

PROPOSITION 2.1. *Let $A = [a_{i,j}]$ in $\mathbb{R}^{n,n}$ be symmetric with all its entries nonnegative, and set*

$$\text{(2.2)} \qquad \tau(A) := \min\{a_{i,j} : i, j \in N\}.$$

If $n > 1$, then A is strictly ultrametric if and only if $A - \tau(A)\xi_n\xi_n^T$ is completely reducible, i.e., there exists a positive integer r with $1 \leq r < n$ and a permutation matrix P in $\mathbb{R}^{n,n}$ such that

$$\text{(2.3)} \qquad P(A - \tau(A)\xi_n\xi_n^T)P^T = \begin{bmatrix} C & O \\ O & D \end{bmatrix},$$

where $C \in \mathbb{R}^{r,r}$ and $D \in \mathbb{R}^{n-r,n-r}$ are each strictly ultrametric.

Proof. For $n > 1$, assume that A is strictly ultrametric. Then, from (1.1) and (2.2), it follows that $\tilde{A} = [\tilde{a}_{i,j}] := A - \tau(A)\xi_n\xi_n^T$ is strictly ultrametric with $\tau(\tilde{A}) = 0$. Moreover, as $n > 1$ and as $\tau(\tilde{A}) = 0$, some off-diagonal entry of $\tilde{A}$ is necessarily zero.

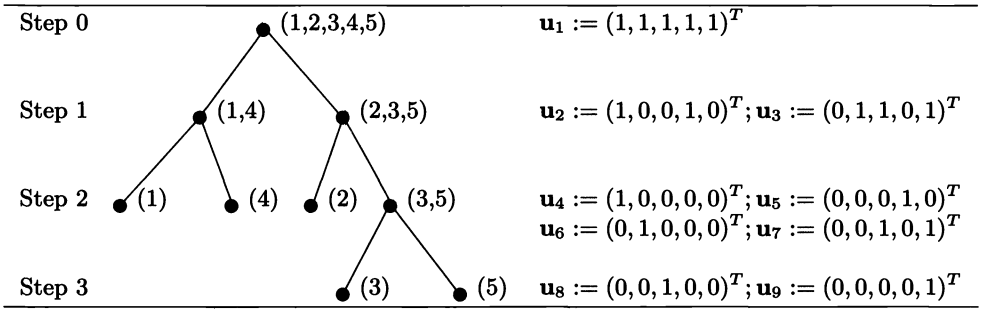


FIG. 1

By a suitable permutation of indices, we may assume, without loss of generality, that $\tilde{a}_{1,n} = 0$. Set

$$S := \{j \in N : \tilde{a}_{1,j} = 0\} \quad \text{and} \quad T := \{j \in N : \tilde{a}_{1,j} > 0\}. \tag{2.4}$$

As $\tilde{a}_{1,n} = 0$, then $n \in S$, and similarly, since $\tilde{A}$ is strictly ultrametric, then (cf. (1.1)(iii)) $\tilde{a}_{1,1} > 0$, so that $1 \in T$. Thus, S and T form a partition of N, i.e., S and T are nonempty disjoint sets with $S \cup T = N$. Again, by a suitable permutation of indices, we may assume, without loss of generality, that

$$T = \{1, 2, \ldots, r\} \quad \text{and} \quad S = \{r+1, r+2, \ldots, n\}, \tag{2.5}$$

where $1 \leq r < n$.

Next, consider any $j \in T$ and any $k \in S$. Since $\tilde{A}$ is a nonnegative matrix, (1.1)(ii) implies that

$$0 = \tilde{a}_{1,k} \geq \min\{\tilde{a}_{1,j}; \tilde{a}_{j,k}\} \geq 0 \qquad (j \in T, k \in S). \tag{2.6}$$

But as $\tilde{a}_{1,j} > 0$ from (2.4), the inequalities of (2.6) and the symmetry of $\tilde{A}$ give that

$$\tilde{a}_{j,k} = 0 = \tilde{a}_{k,j} \qquad (j \in T, k \in S), \tag{2.7}$$

which gives the desired representation of (2.3). That the block diagonal submatrices C and D in (2.3) are each strictly ultrametric is a consequence of the fact that $\tilde{A}$ is strictly ultrametric.

Conversely, if $n > 1$, if $C \in \mathbb{R}^{r,r}$ and if $D \in \mathbb{R}^{n-r,n-r}$ (with $1 \leq r < n$) are each strictly ultrametric, and if $\tau \geq 0$, then from Definition 1.1, the matrix

$$\begin{bmatrix} C & O \\ O & D \end{bmatrix} + \tau \xi_n \xi_n^T$$

is also strictly ultrametric. □

It is evident that the steps leading to the representation (2.3) can be similarly applied to each of the strictly ultrametric block submatrices C and D of (2.3), provided that their orders each exceed unity. More precisely, if $C \in \mathbb{R}^{r,r}$ and if $D \in \mathbb{R}^{n-r,n-r}$ where $1 < r < n-1$, then $C - \tau(C)\xi_r \xi_r^T$ and $D - \tau(D)\xi_{n-r}\xi_{n-r}^T$ are, from the proof of Proposition 2.1, each completely reducible strictly ultrametric matrices. This process can be continued until only 1×1 *positive* matrices remain. This entire reduction procedure can be described in terms of graph theory, as follows.

For illustration, consider an ultrametric matrix A in $\mathbb{R}^{5,5}$, and suppose that the block submatrices C and D in (2.3) are of orders 2 and 3, respectively. This is shown in (reduction) Step 1 in the rooted *reduction tree* of Fig. 1, where the top vertex of Step 0 is associated with the set (1,2,3,4,5). At Step 1, the set (1,2,3,4,5) is decomposed into the two nonempty disjoint sets (1,4) and (2,3,5), giving rise to two vertices in the tree at Step 1. This step corresponds to the complete reducibility of $A - \tau(A)\xi_5\xi_5^T$ in (2.3). In Step 2, each of the sets (1,4) and (2,3,5) is further decomposed into two disjoint nonempty sets, giving rise to four vertices in the tree at Step 2, and this procedure is continued until all remaining sets have single elements. In this way, the 5×5 ultrametric matrix A has the representation

$$A = \sum_{\ell=1}^{9} \tau_\ell \mathbf{u}_\ell \mathbf{u}_\ell^T, \tag{2.8}$$

where the sum over nine terms in (2.8) comes from the fact that there are nine vertices in the tree of Fig. 1. The associated vectors $\mathbf{u}_\ell$ are also explicitly given in Fig. 1. The scalars $\{\tau_\ell\}_{\ell=1}^9$ are nonnegative, with (cf. (1.1)(iii)) $\tau_4, \tau_5, \tau_6, \tau_8$, and τ_9 necessarily positive numbers. In fact, if the constants $\{\tau_1, \tau_2, \ldots, \tau_9\}$ in (2.8) are chosen to be $\{1,0,0,1,1,1,2,1,1\}$, then A can be computed from (2.8) to be

$$A = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 4 & 1 & 3 \\ 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 3 & 1 & 4 \end{bmatrix}.$$

But, it is easy to verify (by induction) that for $N = (1, 2, \ldots, n)$, the reduction steps, as indicated in Fig. 1 for $n = 5$, give exactly $2n - 1$ vertices for its associated reduction tree. Hence, Proposition 2.1 gives the following representation for strictly ultrametric matrices in $\mathbb{R}^{n,n}$ for *all* $n \geq 1$, which goes beyond the results of [2].

THEOREM 2.2. *Given any strictly ultrametric matrix A in $\mathbb{R}^{n,n}$ ($n \geq 1$), there is an associated rooted tree for $N = \{1, 2, \ldots, n\}$, consisting of $2n - 1$ vertices, such that*

$$A = \sum_{\ell=1}^{2n-1} \tau_\ell \mathbf{u}_\ell \mathbf{u}_\ell^T, \tag{2.9}$$

where the vectors $\mathbf{u}_\ell$ in (2.9), determined from the vertices of the tree, are nonzero vectors in $\mathbb{R}^n$ having only 0 and 1 components, and, with the notation that

$$\chi(\mathbf{u}_\ell) := \textit{ sum of the components of } \mathbf{u}_\ell, \tag{2.10}$$

where the τ_ℓ's in (2.9) are nonnegative with $\tau_\ell > 0$ when $\chi(\mathbf{u}_\ell) = 1$. Conversely, given any tree for $N = \{1, 2, \ldots, n\}$, which determines the vectors $\mathbf{u}_\ell$ in $\mathbb{R}^n$, and given any nonnegative constants $\{\tau_\ell\}_{\ell=1}^{2n-1}$ with $\tau_\ell > 0$ when $\chi(\mathbf{u}_\ell) = 1$, then $\sum_{\ell=1}^{2n-1} \tau_\ell \mathbf{u}_\ell \mathbf{u}_\ell^T$ is strictly ultrametric in $\mathbb{R}^{n,n}$.

COROLLARY 2.3. *Any strictly ultrametric matrix in $\mathbb{R}^{n,n}$ is a real symmetric and positive definite matrix.*

Proof. From Theorem 2.2, any strictly ultrametric matrix admits a representation (2.9) as a sum of rank-one nonnegative definite symmetric matrices. But, as the condition that τ_ℓ be positive whenever $\chi(\mathbf{u}_\ell) = 1$ implies that the sum in (2.9) contains a positive diagonal matrix, the sum (2.9) is necessarily positive definite. □

3. Proof of Theorem 1.2. With the constructions of §2, we come to the proof of Theorem 1.2. The proof is an induction on n. If A is an $n \times n$ strictly ultrametric matrix, then from Corollary 2.3, A is nonsingular and A^{-1} exists. That A^{-1} is a strictly diagonally dominant Stieltjes matrix that also satisfies (1.2) of Theorem 1.2 is obvious for $n = 1$. Thus, by the inductive hypothesis, assume that Theorem 1.2 is valid for all ultrametric matrices in $\mathbb{R}^{j,j}$ with $1 \leq j \leq n-1$ where $n \geq 2$, and consider any strictly ultrametric matrix $A = [a_{i,j}]$ in $\mathbb{R}^{n,n}$. Up to a suitable permutation, we have from (2.2) and (2.3) that

$$A = \begin{bmatrix} C & O \\ O & D \end{bmatrix} + \tau(A)\xi_n\xi_n^T \quad \text{with } \xi_n := (1, 1, \ldots, 1)^T \in \mathbb{R}^n, \tag{3.1}$$

where, from Proposition 2.1, C in $\mathbb{R}^{r,r}$ and D in $\mathbb{R}^{n-r,n-r}$ (with $1 \leq r < n$) are both strictly ultrametric and nonsingular. But as r and $n - r$ are both less than n, the inductive hypothesis, applied to C and D, gives that C^{-1} and D^{-1} are strictly diagonally dominant Stieltjes matrices. Hence, if

$$M := \begin{bmatrix} C & O \\ O & D \end{bmatrix} \quad \text{so that } M^{-1} = \begin{bmatrix} C^{-1} & O \\ O & D^{-1} \end{bmatrix}, \tag{3.2}$$

then M^{-1} is also a strictly diagonally dominant Stieltjes matrix. Next, the Sherman–Morrison formula (cf. Golub and Van Loan [1, p. 51]), applied to (3.1), gives the following representation for A^{-1} of (3.1):

$$\left(M + \tau(A)\xi_n \cdot \xi_n^T\right)^{-1} = A^{-1} = M^{-1} - \frac{\tau(A)M^{-1}\xi_n\xi_n^T M^{-1}}{[1 + \tau(A)\xi_n^T M^{-1}\xi_n]}. \tag{3.3}$$

We first claim that the term in brackets in the denominator above is *positive*. To see this, M^{-1}, as previously noted, is a strictly diagonally dominant Stieltjes matrix, so that $M^{-1}\xi_n$ is a positive vector in $\mathbb{R}^n$. On writing $M^{-1}\xi_n := \mathbf{p} > 0$, this denominator is just

$$\left[1 + \tau(A)\xi_n^T M^{-1}\xi_n\right] = 1 + \tau(A)\xi_n^T \mathbf{p} \geq 1. \tag{3.4}$$

Moreover, since $M\mathbf{p} = \xi_n$ and since M is real symmetric, then the last term in (3.3) can be expressed as the matrix

$$-\frac{\tau(A)}{[1 + \tau(A)\xi_n^T \mathbf{p}]}\mathbf{p}\mathbf{p}^T, \tag{3.5}$$

which is obviously a real nonpositive definite symmetric matrix in $\mathbb{R}^{n,n}$, all of whose terms are zero if $\tau(A) = 0$, or negative if $\tau(A) > 0$. But, as the matrix of (3.5) is *added* in (3.3) to M^{-1}, which as noted above is a Stieltjes matrix, then all off-diagonal entries of A^{-1} are necessarily nonpositive.

To show that A^{-1} is strictly diagonally dominant, let

$$M^{-1}\xi_n = \mathbf{p} =: (p_1, p_2, \ldots, p_n)^T > \mathbf{0}.$$

For the ith row sum of A^{-1}, it follows from the second part of (3.3) and (3.5) that

$$(A^{-1}\xi_n)_i = p_i - \frac{\tau(A)p_i \sum_{j=1}^{n} p_j}{\left[1 + \tau(A)\sum_{j=1}^{n} p_j\right]} = \frac{p_i}{\left[1 + \tau(A)\sum_{j=1}^{n} p_j\right]} > 0 \quad (i \in N). \tag{3.6}$$

But, as all off-diagonal entries $\alpha_{i,j}$ of A^{-1} are nonpositive, (3.6) succinctly and precisely gives that A^{-1} is strictly diagonally dominant!

Finally, we establish (cf. (1.2)) that $\alpha_{i,j} = 0$ if and only if $a_{i,j} = 0$. First, if $\tau(A) > 0$, then the strictly ultrametric matrix $A = [a_{i,j}]$ is, up to a permutation matrix P, given from (3.1) by the sum

$$A = \begin{bmatrix} C & O \\ O & D \end{bmatrix} + \tau(A)\xi_n\xi_n^T, \tag{3.7}$$

which has only positive entries, i.e., $a_{i,j} > 0$ for all i, j in N. On the other hand, from (3.3) and (3.4),

$$A^{-1} = \begin{bmatrix} C^{-1} & O \\ O & D^{-1} \end{bmatrix} - \frac{\tau(A)\mathbf{p}\mathbf{p}^T}{[1 + \tau(A)\xi_n^T\mathbf{p}]}, \tag{3.8}$$

where every entry of the last matrix is negative. As the matrices C and D in (3.7) are strictly ultrametric from Proposition 2.1, then C^{-1} and D^{-1} are Stieltjes matrices. Thus, from (3.8), the entries $\alpha_{i,j}$ of A^{-1} satisfy $\alpha_{i,j} < 0$ for all $i \neq j$. Moreover, since A^{-1} is a strictly diagonal dominant matrix, then $\alpha_{i,i} > 0$ for all $1 \leq i \leq n$. Hence, in this case where $\tau(A) > 0$, (1.2) of Theorem 1.2 vacuously holds.

If $\tau(A) = 0$, then from (3.7) we have that

$$A = \begin{bmatrix} C & O \\ O & D \end{bmatrix} \quad \text{and} \quad A^{-1} = \begin{bmatrix} C^{-1} & O \\ O & D^{-1} \end{bmatrix},$$

so that A and A^{-1} have the same off-diagonal blocks of zeros. But we can evidently apply the inductive hypothesis to the block submatrices C and D, and we thus establish (1.2), namely, that the zero entries of A and A^{-1} are the same. □

Having established Theorem 1.2, we deduce from it the following corollary, which appears in [2, Lemma 1] as a step in establishing proof of Theorem 1.2.

COROLLARY 3.1. *Let A in $\mathbb{R}^{n,n}$ be strictly ultrametric. If $\xi_n := (1, 1, \ldots, 1)^T$ in $\mathbb{R}^n$, then there exists a vector $\mathbf{p}$ in $\mathbb{R}^n$, with all positive components, such that*

$$A\mathbf{p} = \xi_n. \tag{3.9}$$

Proof. From Theorem 1.2, A^{-1} is a strictly diagonally dominant Stieltjes matrix in $\mathbb{R}^{n,n}$. Hence $A^{-1}\xi_n =: \mathbf{p} > \mathbf{0}$, from which (3.9) directly follows. □

In conclusion, we note that the more general problem of determining which nonsingular matrices in $\mathbb{R}^{n,n}$, with nonnegative coefficients, have inverses that are M-matrices, has been studied by a number of authors over the years. Although we know of no overlap between the results of this paper and results from these more general investigations, we have nonetheless listed, for the benefit of interested readers, a number of papers [4]–[8] that deal with this more general problem.

Acknowledgment. We thank Professor C. R. Johnson for stimulating discussions related to this research.

REFERENCES

[1] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[2] S. MARTÍNEZ, G. MICHON, AND J. SAN MARTÍN, *Inverses of ultrametric matrices are of Stieltjes type*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 98–106.

[3] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1962.
[4] C. R. JOHNSON, *Inverse M-matrices*, Linear Algebra Appl., 47 (1982), pp. 195–216.
[5] I. KOLTRACHT AND M. NEUMANN, *On the inverse M-matrix problems for real symmetric positive-definite Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 310–320.
[6] T. L. MARKHAM, *Nonnegative matrices whose inverses are M-matrices*, Proc. Amer. Math. Soc., 36 (1972), pp. 326–330.
[7] M. NEUMANN AND R. J. PLEMMONS, *Generalized inverse-positivity and splittings of M-matrices*, Linear Algebra Appl., 23 (1979), pp. 21–35.
[8] R. A. WILLOUGHBY, *The inverse M-matrix problem*, Linear Algebra Appl., 18 (1977), pp. 75–94.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 114–128, January 1994

008

GENERALIZED DISPLACEMENT STRUCTURE FOR BLOCK-TOEPLITZ, TOEPLITZ-BLOCK, AND TOEPLITZ-DERIVED MATRICES*

Dedicated to Gene H. Golub on the occasion of his 60th birthday.

T. KAILATH† AND J. CHUN†‡

Abstract. The concept of displacement structure has been used to solve several problems connected with Toeplitz matrices and with matrices obtained in some way from Toeplitz matrices (e.g., by combinations of multiplication, inversion, and factorization). Matrices of the latter type will be called Toeplitz-derived (or Toeplitz-like, close-to-Toeplitz). This paper introduces a generalized definition of displacement for block-Toeplitz and Toeplitz-block arrays. It will turn out that Toeplitz-derived matrices are perhaps best regarded as particular Schur complements obtained from suitably defined block matrices. The new displacement structure is used to obtain a generalized Schur algorithm for fast triangular and orthogonal factorizations of all such matrices and well-structured fast solutions of the corresponding exact and overdetermined systems of linear equations. Furthermore, this approach gives a natural generalization of the so-called Gohberg–Semencul formulas for Toeplitz-derived matrices.

Key words. Toeplitz matrix, displacement structure, factorization, generalized Gohberg–Semencul formulas, Schur complements

AMS subject classifications. primary 65F05, 65F30; secondary 15A06

1. Introduction. Fast algorithms for triangular and orthogonal matrix factorization, matrix inversions (least-squares) solutions of linear equations, and several related results are now widely known for Toeplitz matrices. The concept of displacement structure [30] was introduced to show, among other things, that fast algorithms could also be obtained for several related matrices that do not have such structure; for example, though not Toeplitz, matrices of the form T_1^{-1}, T_1T_2, $T_1 - T_2T_3^{-1}T_4$, $T_1T_2 - T_3T_4$, where the $\{T_i\}$ are Toeplitz matrices, all possess fast algorithms. The reason basically is that all these matrices have low displacement rank. The *displacement* of a (square) matrix A was defined in [30] as

$$\nabla A = A - Z_n A Z_n^T, \tag{1}$$

where Z_n is the $n \times n$ (lower) shift matrix with 1's on the first subdiagonal and 0's elsewhere. For a Toeplitz matrix, T, it is easy to see that ∇T will be identically zero except for the first row and first column, so that the displacement rank of T = rank $\nabla T \leqq 2$ no matter what the size of T. A significant fact is that, though T^{-1} is not in general Toeplitz, rank $\nabla T^{-1} \leqq 2$. So, also, though T_1T_2 is not in general Toeplitz, rank $\nabla T_1T_2 \leqq 4$. These facts have been exploited to obtain fast $O(n^2)$ algorithms for factoring matrices such as T^{-1} and T_1T_2 and others. Nevertheless we show in this article that matrices such as T^{-1}, $T_1 - T_2T_3^{-1}T_4$, $T_1T_2 - T_3T_4$, and so on may be better studied by first trying to find an appropriate "Toeplitz-block" matrix in which these

* Received by the editors August 2, 1989; accepted for publication (in revised form) April 15, 1992.

† Information Systems Laboratory, Stanford University, Stanford, California 94305 (tk@isl.stanford.edu,, chun@rascals.stanford.edu). J. Chun is currently with KAIST, Dept. of Information and Comm. Eng., P.O. Box 201, Cheongryang, Seoul, Korea.

‡ This work was supported by the U.S. Army Research Office, under contract DAAL03-86-K-0045, and by the Strategic Defense Initiative Organization/Innovative Science and Technology, managed by the Army Research Office under contract DAAL03-87-K-0033. This manuscript is submitted for publication with the understanding that the U.S. Government is authorized to reproduce and distribute reprints for Government purpose notwithstanding any copyright notation therein.

matrices appear as certain Schur-complement matrices and then analyzing these Toeplitz-block matrices by introducing a *suitably modified* definition of displacement; the key fact then used is that displacement structure is preserved under Schur complementation. Besides enabling us to solve several new problems, this procedure also provides a new and simpler approach to many of the problems studied in [4]–[6], [9]–[11], [13]–[15], [19]–[21], [23], [27], [36], [37], [41], [43], [44]. It will perhaps be clearest to present two simple examples.

Example 1. Study of T^{-1}. First note that T^{-1} is the Schur complement of the (1,1) block in the Toeplitz-block matrix

$$A = \begin{bmatrix} -T & 1 \\ 1 & O \end{bmatrix}. \tag{2}$$

It is a known result (see, e.g., [13], [37]) that the displacement rank of a Schur complement of A cannot exceed the displacement rank of A, which is 4, in general, because

$$\nabla A = \left[\begin{array}{ccc|ccc} -t_0 & -t_1 & -t_{n-1} & 1 & 0 & 0 \\ -t_1 & & & t_{n-1} & & \\ \vdots & & 0 & \vdots & & 0 \\ -t_{n-1} & & & t_1 & & \\ \hline 1 & t_{n-1} & t_1 & t_0 & 0 & 0 \\ 0 & & 0 & 0 & & \\ \vdots & & \vdots & \vdots & & 0 \\ 0 & & 0 & 0 & & \end{array}\right].$$

However, it is well known that the displacement rank of T^{-1} cannot exceed 2. Therefore, though the idea of studying matrices such as T^{-1} and $T_1 - T_2 T_3^{-1} T_4$ as Schur complements of suitable block matrices is not new (see especially the work of Delosme [13] and [14] and others [5], [37], [43], [44]), doing this with the definition (1) will lead to more complex algorithms than necessary.

On the other hand, suppose we define the modified displacement rank of A by the rank of the matrix

$$\nabla_{(F,F)} A = A - FAF^T, \qquad F = \begin{bmatrix} Z_n & O \\ O & Z_n \end{bmatrix} = Z_n \oplus Z_n.$$

Then note that

$$\nabla_{(F,F)} A = \begin{bmatrix} -\nabla_{(Z_n,Z_n)} T & \nabla_{(Z_n,Z_n)} I \\ \nabla_{(Z_n,Z_n)} I & O \end{bmatrix}$$

$$= \left[\begin{array}{ccc|ccc} -t_0 & -t_1 & -t_{n-1} & 1 & 0 & 0 \\ -t_1 & & & & & \\ \vdots & & 0 & & & 0 \\ -t_{n-1} & & & & & \\ \hline 1 & & & & & \\ 0 & & & & & \\ \vdots & & 0 & & & 0 \\ 0 & & & & & \end{array}\right]$$

so that the F displacement rank of A is 2. Now the previously mentioned result on Schur complements will show that $\nabla_{(Z_n,Z_n)} T^{-1} \leqq 2$. □

Example 2. Displacement rank of products. In [9], we gave a rather uninspiring proof of the inequality

$$\text{rank } [\nabla_{(Z_n,Z_n)}(B_1B_2)] \leqq \text{rank } [\nabla_{(Z_n,Z_n)}B_1] + \text{rank } [\nabla_{(Z_n,Z_n)}B_2] + 1. \tag{3}$$

An interesting proof follows by considering the block matrix

$$A = \begin{bmatrix} -I & B_2 \\ B_1 & O \end{bmatrix}. \tag{4}$$

Note that B_1B_2 is the Schur complement of $-I$ in A and that

$$\nabla_{(F,F)}A = \left[\begin{array}{c|c} -1 & \nabla_{(Z_n,Z_n)}B_2 \\ \hline 0 & \\ \hline \nabla_{(Z_n,Z_n)}B_1 & O \end{array}\right], \qquad F = Z_n \oplus Z_n.$$

Now, (3) follows from the fact that rank $[\nabla_{(Z_n,Z_n)}(B_1B_2)] \leqq$ rank $[\nabla_{(F,F)}A]$.

On the other hand, note that if (consistent with the original definition (1)) we had used Z_{2n} instead of $Z_n \oplus Z_n$, we would have obtained a looser bound,

$$\text{rank } [\nabla_{(Z_n,Z_n)}(A_1A_2)] \leqq \text{rank } [\nabla_{(Z_n,Z_n)}A_1] + \text{rank } [\nabla_{(Z_n,Z_n)}A_2] + 3.$$

Tighter bounds on the displacement rank of matrices are important because the operation count of fast algorithms increases according to these bounds rather than to the displacement rank. This is the feature made possible by using properly extended definitions of displacement rank. □

An appropriate generalization of the ideas in these simple examples is introduced in this article along with several applications. Section 2 gives the general definitions. The heart of the article is § 3, where a generalized Schur algorithm is derived. Several applications are given in § 4. The concluding section reviews the main idea and makes comparisons with earlier approaches also using Schur complements (e.g., [5], [13], [37]).

2. Definitions and notations. Let $A \in \mathbf{R}^{m\times n}$ be a given matrix and let F^f and F^b be *strictly* lower triangular matrices. The matrix

$$\nabla_{(F^f,F^b)}A \equiv A - F^fAF^{bT}, \qquad F^{bT} \equiv (F^b)^T \tag{5}$$

is called the *displacement* of A with respect to the *displacement operators* $\{F^f, F^b\}$. Any matrix pair $\{X, Y\}$ such that

$$\nabla_{(F^f,F^b)}A = XY^T, \quad X \equiv [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\alpha], \quad Y \equiv [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\alpha] \tag{6}$$

is called a *generator* of A (with respect to $\{F^f, F^b\}$). The number α is called the *length* of the generator (with respect to $\{F^f, F^b\}$). A generator of A with the minimal possible length is called a *minimal* generator. The length of the minimal generator of A (i.e., rank $(\nabla_{(F^f,F^b)}A) \leqq \alpha$) is called the *displacement rank* of A (with respect to $\{F^f, F^b\}$) and is denoted as $\alpha_{(F^f,F^b)}(A)$.

If $\{X, Y\}$ is a generator of A with respect to $\{F^f, F^b\}$, then for any nonsingular matrix $S \in R^{\alpha\times\alpha}$, the matrix pair $\{XS, YS^{-T}\}$ is also a generator of A because

$$\nabla_{(F^f,F^b)}A = XY^T = XSS^{-1}Y^T.$$

Hence, generators (even minimal ones) are not unique. For block-Toeplitz or Toeplitz-block matrices, it is straightforward to obtain generators from the displacements by inspection.

Note that the displacement of a symmetric matrix A can be written as $\nabla_{(F,F)}A = X\Sigma X^T$ where Σ is a diagonal matrix with 1 or -1 along the main diagonal; we say that A has a *symmetric generator*, $\{X, X\Sigma\}$, with respect to F.

We should note that the following sum-of-products representation of a matrix A solves (5),

$$A = \sum_{i=1}^{\alpha} K_n(\mathbf{x}_i, F^f)K_n^T(\mathbf{y}_i, F^b), \qquad A \in \mathbf{R}^{m\times n}, \tag{7}$$

where $K_n(\mathbf{x}_i, F^f) \in \mathbf{R}^{m\times n}$ and $K_n(\mathbf{y}_i, F^b) \in \mathbf{R}^{n\times n}$ are the so-called Krylov matrices

$$K_n(\mathbf{x}_i, F^f) \equiv [\mathbf{x}_i, F^f\mathbf{x}_i, \ldots, (F^f)^{n-1}\mathbf{x}_i], \quad K_n(\mathbf{y}_i, F^b) \equiv [\mathbf{y}_i, F^b\mathbf{y}_i, \ldots, (F^b)^{n-1}\mathbf{y}_i].$$

The *strict* triangularity of $\{F^f, F^b\}$ is important in this result; otherwise the Krylov matrices would have an infinite number of columns. We shall show in § 4 that the representation (7) yields generalizations of the celebrated Gohberg–Semencul formula for the inverse of a Toeplitz matrix (see [22], [24]).

Choice of displacement operators. Let $\{X, Y\}$ be a generator of length α of A with respect to F^f and F^b. If the matrix-vector multiplications $F^f\mathbf{u}$ and $F^b\mathbf{v}$ take $f(n)$ and $b(n)$ operations, respectively, then the algorithms to be presented in § 3 will need $O(\alpha n g(n))$ operations, where $g(n) = \max(f(n), b(n))$. Therefore, the objective is to choose the "simplest" or sparse (to make $g(n)$ small) strictly lower triangular matrices F^f and F^b that also make α as small as possible. For a scalar $n \times n$ Toeplitz matrix, a natural choice of displacement operator is the $n \times n$ shift matrix, Z_n, with 1's along the first subdiagonal and 0's elsewhere. We give some heuristic choices for block-Toeplitz matrices, Toeplitz-block matrices, and their combinations.

For an $M \times N$ *Toeplitz-block array* with $m_i \times n_j$ Toeplitz matrices $T_{i,j}$,

$$A = \begin{bmatrix} T_{1,1} & T_{1,2} & \cdot & T_{1,N} \\ T_{2,1} & T_{2,2} & \cdot & T_{2,N} \\ \cdot & \cdot & \cdot & \cdot \\ T_{M,1} & T_{M,2} & \cdot & T_{M,N} \end{bmatrix} \in \mathbf{R}^{m\times n}, \tag{8}$$

we shall use the displacement operators,

$$F^f = \bigoplus_{i=1}^{M} Z_{m_i}, \qquad F^b = \bigoplus_{i=1}^{N} Z_{n_i},$$

where $\oplus_{i=1}^{M} F_i$ denotes the concatenated direct sum, i.e., the block diagonal matrix whose ith diagonal block is F_i.

For an $M \times N$ *block-Toeplitz array* with $r \times s$ rectangular blocks,

$$A = \begin{bmatrix} B_0 & B_{-1} & \cdot & B_{-N+1} \\ B_1 & B_0 & \cdot & B_{-N+2} \\ \cdot & \cdot & \cdot & \cdot \\ B_{M-1} & B_{M-2} & \cdot & B_{-N+M} \end{bmatrix}, \tag{9}$$

a natural choice is

$$F^f = Z_{Mr}^{M}, \qquad F^b = Z_{Ns}^{N},$$

where $Z_{kr}^{k} = [Z_{k\cdot r}]^k$ can be seen as a *block shift matrix*, i.e., a $k \times k$ array with $r \times r$ identity matrices on the first block subdiagonal and 0's elsewhere.

Example 3. Consider the matrix A,

$$A = \begin{bmatrix} I & B & O \\ B^T & O & I \\ O & I & O \end{bmatrix}, \qquad B \in \mathbf{R}^{m \times n}.$$

If B is a Toeplitz matrix, then choosing

$$F^f = F^b = Z_m \oplus Z_n \oplus Z_n$$

will give a displacement rank of 4 for A. If B is an $M \times N$ block-Toeplitz array with $r \times s$ blocks, we could choose

$$F^f = F^b = Z_{Mr}^M \oplus Z_{Ns}^N \oplus Z_{Ns}^N.$$

If B is a Toeplitz-block array, for example, $B = [T_1 \quad T_2]$, where $T_1 \in \mathbf{R}^{m_1 \times n}$ and $T_2 \in \mathbf{R}^{m_2 \times n}$, then we could choose

$$F^f = F^b = Z_{m_1} \oplus Z_{m_2} \oplus Z_n \oplus Z_n.$$

We can obtain a generator of A for each case by inspection. For example, for the case of Toeplitz $B = (b_{i-j})$, the following matrix

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & b_1 & 0 & b_1 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & b_{m-1} & 0 & b_{m-1} \\ b_0 & .5 & b_0 & -.5 \\ b_{-1} & 0 & b_{-1} & 0 \\ \cdot & \cdot & \cdot & \cdot \\ b_{1-n} & 0 & b_{1-n} & 0 \\ 0 & 1 & 0 & 1 \\ \cdot & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{bmatrix}$$

is a generator of A with respect to $Z_m \oplus Z_n \oplus Z_n$. More systematic procedure is described in the Appendix.

Proper generators. Let $\{X, Y\}$ be a generator of a matrix A. We say that a generator is *proper* (with respect to the pivoting column j) if, for a certain i, all the elements in the ith row of X and above, except for the element $[X]_{i,j}$, are zero and all elements in the ith row of Y and above, except the element $[Y]_{i,j}$, are zero. Often we shall denote a proper generator as $\{X_p, Y_p\}$. If $\{X, Y\}$ is not proper, then by choosing an appropriate S, we can obtain a proper generator $\{XS, YS^{-T}\}$ under certain conditions on the matrix A. A procedure for doing this is described in § 3.

3. Generalized Schur algorithm. A fundamental method for triangular matrix factorization is the so-called *Schur reduction* process (see [40] and, e.g., [13], [14], [33], [34], [37], [38]), which successively computes the Schur complements of the leading submatrices iteratively; displacement structure allows the computation to be speeded up. Our fast algorithms will be based on the following theorem.

THEOREM 1. *Let* $\{X_p^{(1)}, Y_p^{(1)}\}$ *be a proper generator of a rectangular matrix* $A^{(1)} \in \mathbf{R}^{m \times n}$ *with respect to* $\{F^f, F^b\}$. *Also assume that* $\{X_p^{(1)}, Y_p^{(1)}\}$ *has been made proper with respect to a particular* (*pivoting*) *column, which we shall index as "pvt." If*

we denote the columns of $X_p^{(1)}$ *and* $Y_p^{(1)}$ *by*

$$X_p^{(1)} = [\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_{\text{pvt}}^{(1)}, \ldots, \mathbf{x}_\alpha^{(1)}], \qquad Y_p^{(1)} = [\mathbf{y}_1^{(1)}, \ldots, \mathbf{y}_{\text{pvt}}^{(1)}, \ldots, \mathbf{y}_\alpha^{(1)}],$$

then the matrix $A^{(2)}$ *defined by*

$$A^{(2)} \equiv A^{(1)} - \mathbf{x}_{\text{pvt}}^{(1)} \mathbf{y}_{\text{pvt}}^{(1)T}$$

has null first column and row, and has a generator $\{X^{(2)}, Y^{(2)}\}$, *with respect to* $\{F^f, F^b\}$ *of the form*

$$X^{(2)} = [\mathbf{x}_1^{(1)}, \ldots, F^f \mathbf{x}_{\text{pvt}}^{(1)}, \ldots, \mathbf{x}_\alpha], \qquad Y^{(2)} = [\mathbf{y}_1^{(1)}, \ldots, F^b \mathbf{y}_{\text{pvt}}^{(1)}, \ldots, \mathbf{y}_\alpha^{(1)}].$$

Remark 1. The matrix $A^{(2)}$ is the Schur complement of $A^{(1)}$ with respect to the (1,1) element of $A^{(1)}$.

Proof.

$$\begin{aligned} A^{(2)} - F^f A^{(2)} F^{bT} &= [A^{(1)} - \mathbf{x}_{\text{pvt}}^{(1)} \mathbf{y}_{\text{pvt}}^{(1)T}] - F^f [A^{(1)} - \mathbf{x}_{\text{pvt}}^{(1)} \mathbf{y}_{\text{pvt}}^{(1)T}] F^{bT} \\ &= A^{(1)} - F^f A^{(1)} F^{bT} - \mathbf{x}_{\text{pvt}}^{(1)} \mathbf{y}_{\text{pvt}}^{(1)T} + F^f \mathbf{x}_{\text{pvt}}^{(1)} \mathbf{y}_{\text{pvt}}^{(1)T} F^{bT} \\ &= X^{(1)} Y^{(1)T} - \mathbf{x}_{\text{pvt}}^{(1)} \mathbf{y}_{\text{pvt}}^{(1)T} + F^f \mathbf{x}_{\text{pvt}}^{(1)} \mathbf{y}_{\text{pvt}}^{(1)T} F^{bT} \\ &= X^{(2)} Y^{(2)T}. \end{aligned}$$

The first column and row of $A^{(2)}$ are null because

$$A^{(2)} \mathbf{e}_1 = [X^{(2)} Y^{(2)T}] \mathbf{e}_1 = 0, \qquad \mathbf{e}_1^T A^{(2)} = \mathbf{e}_1^T [X^{(2)} Y^{(2)T}] = 0,$$

where we have used the fact that F^f and F^b are strictly lower triangular and $\{X_p^{(1)}, Y_p^{(1)}]$ is proper. □

By applying the previous theorem using such a proper generator we can obtain a (possibly nonproper) generator of $A^{(2)}$. Converting this to proper form, we can proceed to find a generator $A^{(3)}$. By repeating this process r times, we shall generate the matrices

$$\begin{aligned} A^{(2)} &= A^{(1)} - \mathbf{x}_{\text{pvt}_1}^{(1)} \mathbf{y}_{\text{pvt}_1}^{(1)T}, \\ &\quad \cdots\cdots, \\ A^{(r)} &= A^{(r-1)} - \mathbf{x}_{\text{pvt}_{r-1}}^{(r-1)} \mathbf{y}_{\text{pvt}_{r-1}}^{(r-1)T}, \\ A^{(r+1)} &= A^{(r)} - \mathbf{x}_{\text{pvt}_r}^{(r)} \mathbf{y}_{\text{pvt}_r}^{(r)T}. \end{aligned} \tag{10}$$

It turns out that this process gives a *partial triangular factorization* of $A^{(1)}$, because (10) shows that

$$A^{(1)} = \sum_{i=1}^{r} \mathbf{x}_{\text{pvt}_i}^{(i)} \mathbf{y}_{\text{pvt}_i}^{(i)T} + A^{(r+1)}$$

$$= \begin{bmatrix} \mathbf{x}_{\text{pvt}_1}^{(1)} & \cdots & \mathbf{x}_{\text{pvt}_r}^{(r)} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{\text{pvt}_1}^{(1)T} \\ \vdots \\ \mathbf{y}_{\text{pvt}_r}^{(r)T} \end{bmatrix} + A^{(r+1)},$$

where

$$A^{(r+1)} \equiv \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & S^{(r+1)} \end{array}\right].$$

Remark 2. The above r-step partial triangularization breaks down if and only if there is a singular leading principal submatrix of order less than or equal to r; we shall assume that this is not so, i.e., the matrix is assumed to be *strongly nonsingular* or strongly regular. Various authors, including ourselves, however, have obtained results for the indefinite matrices with singular leading principal submatrix case as well (see [38]).

The above process can be summarized in the following algorithm, which we shall call a *generalized Schur algorithm*.

GENERALIZED SCHUR ALGORITHM
Input: A generator $\{X, Y\}$ of $A \in \mathbf{R}^{m\times n}$ with respect to $\{F^f, F^b\}$.
Output: (i) Partial triangular factors $L \in \mathbf{R}^{m\times r}$ and $U \in \mathbf{R}^{r\times n}$ of A.

(ii) A generator $\{X, Y\}$ of the Schur complement of the $r \times r$ leading principal submatrix of A,
Procedure:

for $k := 1$ **to** r **do begin**
  Find a proper generator of $A^{(k)}$;
  The kth column of $L := \mathbf{x}_{\text{pvt}}$; The kth row of $U := \mathbf{y}_{\text{pvt}}^T$;
  Replace $\mathbf{x}_{\text{pvt}}$ with $F^f\mathbf{x}_{\text{pvt}}$ and $\mathbf{y}_{\text{pvt}}$ with $F^b\mathbf{y}_{\text{pvt}}$ to get a generator of $A^{(k+1)}$;
end
return $(L, U, \{X, Y\})$;

Note that the above procedure needs $O(\alpha pr)$ operations, where $p = \max(m, n)$ and α is the length of the given generator, if the operation of making a generator proper takes $O(\alpha p)$ operations. We now show how to do this.

Construction of proper generators. This can be done in various ways. We shall describe a method using elementary matrices known as *spinors*; for methods using Householder matrices, see, e.g., [7], [12], [13], [16], [32]. A spinor $S_{(j|i)} \in \mathbf{R}^{\alpha\times\alpha}$ is defined as the identity matrix except for the following four entries,

$$[S_{(j|i)}]_{i,i} = c, \quad [S_{(j|i)}]_{i,j} = s_2, \quad [S_{(j|i)}]_{j,i} = -s_1, \quad [S_{(j|i)}]_{j,j} = c,$$

where $[A]_{i,j}$ denotes the (i, j)th element of the matrix A and $c^2 + s_1s_2 = 1$. The inverse of a spinor is also a spinor, viz., $S_{(j|i)}^{-1}$ is the identity matrix except for the following four entries,

$$[S_{(j|i)}^{-1}]_{i,i} = c, \quad [S_{(j|i)}^{-1}]_{i,j} = -s_2, \quad [S_{(j|i)}^{-1}]_{j,i} = s_1, \quad [S_{(j|i)}^{-1}]_{j,j} = c.$$

Let $\mathbf{x}^T \in \mathbf{R}^{1\times\alpha}$ and $\mathbf{y}^T \in \mathbf{R}^{1\times\alpha}$ be row vectors. Let c, s_1, and s_2 be chosen as

$$c = \left[\frac{x_i y_i}{x_i y_i + x_j y_j}\right]^{1/2}, \quad s_2 = -c\cdot\frac{x_j}{x_i}, \quad s_1 = -c\cdot\frac{y_j}{y_i},$$

and define $\mathbf{x}'$ and $\mathbf{y}'$ by

$$\mathbf{x}'^T \equiv \mathbf{x}^T S_{(j|i)}, \qquad \mathbf{y}'^T \equiv \mathbf{y}^T S_{(j|i)}^{-T}.$$

Then it is easy to check that $x_j' = y_j' = 0$ and $\mathbf{x}'^T\mathbf{y}' = \mathbf{x}^T\mathbf{y}$. We shall call the elements x_i and y_i *pivoting elements*. Therefore, by repeating this process we can *annihilate* all ele-

ments of $\mathbf{x}$ and $\mathbf{y}$ except the pivoting elements, resulting in

$$[0, \ldots, 0, x_i, 0, \ldots, 0] = \mathbf{x}^T \prod_{j \neq i}^{\alpha} S_{(j|i)}, \qquad [0, \ldots, 0, y_i, 0, \ldots, 0] = \mathbf{y}^T \prod_{j \neq i}^{\alpha} S_{(j|i)}^{-T}.$$

An arbitrary choice of pivoting element or an arbitrary *ordering* of annihilation might result in $[1 + (x_j y_i / x_i y_i)] \leqq 0$ for some j, for which real spinors do not exist. This issue is handled by the following lemma, whose proof along with other related results can be found in [7].

LEMMA 1. *Let* $\gamma_i = x_i y_i$ *and* $s = \sum_i \gamma_i > 0$ (<0). *If we choose a pivot element such that* $\gamma_i > 0$ (<0), *and if we annihilate all elements with* $\gamma_i > 0$ (<0) *before annihilating elements with* $\gamma_i < 0$ (>0), *then* $[1 + (x_j x_j / x_i y_i)] > 0$ *for all* $1 \leqq j \leqq \alpha$, $j \neq i$.

Some special cases. If we are given a symmetric generator of a symmetric matrix A, i.e., if $Y = X\Sigma$, then the updating of Y in the above procedure is redundant, because the updated $\{X', Y'\}$ after annihilating a row still remains symmetric. To see this, let

$$\mathbf{x}^T = \mathbf{y}^T = [x_{\text{pvt}}, x_j].$$

Then the spinor that annihilates x_j will reduce to a *Givens rotation*,

$$G_{(j|\text{pvt})} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \qquad c^2 + s^2 = 1.$$

On the other hand, if

$$\mathbf{x}^T = [x_{\text{pvt}}, x_j], \qquad \mathbf{y}^T = [x_{\text{pvt}}, -x_j],$$

the spinor will become a *hyperbolic rotation*,

$$H_{(j|\text{pvt})} = \begin{bmatrix} ch & -sh \\ -sh & ch \end{bmatrix}, \qquad ch^2 - sh^2 = 1.$$

Notice that Givens and hyperbolic rotations preserve the symmetry of the updated generator, i.e.,

$$YS^{-T} = Y' = X'\Sigma, \quad X' = XS, \quad S\text{: a Givens or hyperbolic rotation.}$$

As another special case of spinors, consider the two row vectors

$$\mathbf{x}^T = [x_{\text{pvt}}, x_j], \qquad \mathbf{y}^T = [y_{\text{pvt}}, 0].$$

For this case, the spinor that annihilates x_j will reduce to the usual *elimination* matrix,

$$E_{(j|\text{pvt})} = \begin{bmatrix} 1 & -\kappa \\ 0 & 1 \end{bmatrix}, \qquad \kappa = \frac{x_j}{x_{\text{pvt}}}. \tag{11}$$

We may mention that Ahmed, Delosme, and Morf [2] showed the significance of such elementary operations for efficient hardware implementation.

Remark 3. For a square Toeplitz-block array $A \in \mathbf{R}^{n \times n}$ with $T_{i,j} \in \mathbf{R}^{m_i \times n_j}$, we can obtain the LU factorization of A by completing the generalized Schur algorithm with $r = n$. Other authors have suggested first transforming A into a block-Toeplitz matrix by pre- and postmultiplication with permutation matrices and then applying an algorithm for square block-Toeplitz matrix to get a row- and column-permuted triangular factorization of A; there is clearly a difficulty with this approach when $m_i \neq m_j$. More importantly, if A is not positive definite, the permuted matrix is not necessarily strongly nonsingular, for which ordinary LU factorization does not exist. Our approach does not have this problem because it directly factorizes A without permutations.

Remark 4. For a square block-Toeplitz array $A \in \mathbf{R}^{n\times n}$, with square blocks $B_i \in \mathbf{R}^{r\times r}$, there exist several fast block triangular factorization algorithms such as the Bareiss algorithm [4], the multichannel Levinson algorithm [3], [35], and the Schur algorithm [1] and [18], all of which require matrix (of the block size $r \times r$) operations. Our approach treats block-Toeplitz matrices in essentially the same way as *scalar* Toeplitz matrices and in particular will use only elementary scalar operations. We remark that the absence of matrix operations such as inversion may simplify the design of dedicated hardware implementations.

4. Applications. By applying the generalized Schur algorithm in § 3 to judiciously chosen block matrices, we can obtain interesting results including fast *QR* factorizations and generalized Gohberg–Semencul formulas. Generators of the block matrices used in this section can be easily found by inspection (see Appendix). The *floating operation* (flop) counts given below are confined to the number of multiplications.

Simultaneous factorization of a symmetric Toeplitz matrix and its inverse. Let $T = (t_{i-j}) \in \mathbf{R}^{n\times n}$, $t_0 = 1$, be a strongly nonsingular symmetric Toeplitz matrix. The matrix

$$A = \begin{bmatrix} T & I \\ I & O \end{bmatrix} \tag{12}$$

has a symmetric generator $\{X, X\Sigma\}$ with respect to $Z_n \oplus Z_n$, where

$$X = \begin{bmatrix} 1 & t_1 & \cdot & t_{n-1} & 1 & 0 & \cdot & 0 \\ 0 & t_1 & \cdot & t_{n-1} & 1 & 0 & \cdot & 0 \end{bmatrix}^T, \qquad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

After performing n steps of partial triangular factorization using the generalized Schur algorithm, we shall have the factors L and U in

$$A = \begin{bmatrix} L \\ U \end{bmatrix} [L^T, U^T] + \begin{bmatrix} O & O \\ O & S \end{bmatrix}. \tag{13}$$

Now, one can check by comparing the entries of A in (12) and (13) that

$$T = LL^T, \quad T^{-1} = UU^T, \quad U\text{: upper triangular.}$$

Recall that the classical Schur algorithm gives only the factorization $T = LL^T$, whereas the Levinson algorithm gives the factorization $T^{-1} = UU^T$. Here we get both simultaneously in $4n^2 + O(n)$ flops (or $2n^2 + O(n)$ if one uses fast rotations; see, e.g., [9] and [25]). The computation only of $T = LL^T$ needs just one half of the above flop counts. If one only needs the factorization of T^{-1}, the above method is slower than the Levinson algorithm; however, the above (Schur) method does not require inner products and therefore is better suited to parallel implementation than the Levinson algorithm.

Orthogonalization of a fully windowed Toeplitz matrix [10]. Let $T = (t_{i-j}) \in \mathbf{R}^{m\times n}$, $m > n$ be a *fully windowed* Toeplitz matrix, i.e.,

$$t_{i-j} = 0, \quad \text{if } j > i, \quad \text{or} \quad i > m - n + j.$$

Then it is easy to check that $B = (b_{i-j}) \equiv T^T T$ is also an (unwindowed) Toeplitz matrix. Now assume that $t_0 \neq 0$ and $t_{m-n} \neq 0$, and consider the following matrix

$$A = \begin{bmatrix} T^T T & T^T \\ T & O \end{bmatrix}, \tag{14}$$

for which it can be checked that a symmetric generator with respect to $Z_n \oplus Z_m$ is

$$X = \begin{bmatrix} \sqrt{b_0} & b_1/\sqrt{b_0} & \cdot & b_{n-1}/\sqrt{b_0} & t_0 & t_1 & \cdot & t_{m-n} & 0 & \cdot & 0 \\ 0 & b_1/\sqrt{b_0} & \cdot & b_{n-1}/\sqrt{b_0} & t_0 & t_1 & \cdot & t_{m-n} & 0 & \cdot & 0 \end{bmatrix}^T, \qquad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

After performing n steps of partial triangular factorization using the generalized Schur algorithm, we shall have the factors R and Q in

$$A = \begin{bmatrix} R^T \\ Q \end{bmatrix} [R, Q^T] + \begin{bmatrix} O & O \\ O & S \end{bmatrix}. \tag{15}$$

By comparing (14) and (15), one can easily see that

$$T^T T = R^T R, \qquad T = QR,$$

so that Q is orthogonal because $R^T Q^T Q R = R^T R$. The computation of $T^T T$ needs $nm - \frac{3}{2}n^2 + O(n)$ flops and the partial triangularization needs additional $8m^2 - 4nm + O(m)$ flops (or $4m^2 - 2nm + O(m)$ flops with fast rotations).

Orthogonalization of a Toeplitz matrix. Let $B \in \mathbf{R}^{m \times n}$ be a Toeplitz, block-Toeplitz, or Toeplitz-block matrix of full-column rank, and let us define the block matrix

$$A \equiv \begin{bmatrix} -I & B & O \\ B^T & O & B^T \\ O & B & I \end{bmatrix}. \tag{16}$$

We can easily find a generator of A with respect to $Z_m \oplus Z_n \oplus Z_m$ by inspection. For example, if B is Toeplitz, a generator of A is given by

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \tilde{b}_1 & 0 & \tilde{b}_1 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \tilde{b}_{m-1} & 0 & \tilde{b}_{m-1} \\ -b_0 & 0 & 0 & b_0 \\ -b_{-1} & b_{-1} & b_{-1} & b_{-1} \\ \cdot & \cdot & \cdot & \cdot \\ -b_{1-n} & b_{-n+1} & b_{1-n} & b_{-n+1} \\ 0 & 1 & 1 & 1 \\ 0 & \tilde{b}_1 & 0 & \tilde{b}_1 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \tilde{b}_{m-1} & 0 & \tilde{b}_{m-1} \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} -1 & & & \\ & -1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix},$$

where $\tilde{b}_i = b_i / b_0$. If we apply the generalized Schur algorithm with the above generator, then after the mth step with $m^2 + 2nm + O(n) + O(m)$ flops, we shall have a generator of

$$A^{(m)} = \begin{bmatrix} B^T B & B^T \\ B & I \end{bmatrix}. \tag{17}$$

After another n steps of partial triangularization with $12mn + 6n^2 + O(m) + O(n)$ flops (or $6mn + 3n^2 + O(m) + O(n)$ flops with fast rotations), we shall have R and Q in

$$A^{(m)} = \begin{bmatrix} R^T \\ Q \end{bmatrix} [R, Q^T] + \begin{bmatrix} O & O \\ O & S \end{bmatrix},$$

such that $B = QR$. One can start with a generator of the block matrix (17), as in the fully windowed case. We note that a closed-form expression for a generator of (17) for block-Toeplitz and Toeplitz-block matrix B can be found in [7]. For a Toeplitz B, the closed-form expression for a generator of (17) can be evaluated in mn flops, and, therefore, it requires less computation to start with it. However, it would be necessary to work with a generator for (16) if B is non-Toeplitz and only its generator is given.

If one wishes to find R^{-1} directly, then one can perform the $(m + n)$ steps of partial triangularization with the matrix

$$A = \begin{bmatrix} -I & B & O \\ B^T & O & I \\ O & I & O \end{bmatrix}.$$

This is because

$$A^{(m)} = \begin{bmatrix} B^TB & I \\ I & O \end{bmatrix} = \begin{bmatrix} R^T \\ U \end{bmatrix} \cdot [R \quad U^T] + \begin{bmatrix} O & O \\ O & S \end{bmatrix},$$

and, therefore, $U = R^{-1}$ because $UR = I$.

Removing forward elimination in square systems. If one's primary interest in the factorization is in solving a square symmetric Toeplitz system of equations,

$$B\mathbf{x} = \mathbf{b}, \quad B = LL^T, \quad B \in \mathbf{R}^{n \times n}, \quad b_0 = 1, \tag{18}$$

then one might want to obtain the transformed right-side vector $\mathbf{y} \equiv L^{-1}\mathbf{b}$ during the course of the factorization process (see, e.g., [2], [25]). This can also be done using the generalized Schur algorithm by performing the following triangular factorization of the matrix A,

$$A \equiv [B \quad \mathbf{b}] = L \cdot [L^T, \mathbf{y}] \tag{19}$$

whence the solution to (18) can be obtained by solving the triangular system of equations

$$L^T\mathbf{x} = \mathbf{y}. \tag{20}$$

Note that the matrix A has displacement rank three with respect to $\{Z_n, Z_n \oplus Z_1\}$, $(Z_1 = 0)$, and a generator is given by

$$X = \begin{bmatrix} b_0 & 0 & \beta_0 \\ b_1 & b_1 & \beta_1 \\ \cdot & \cdot & \cdot \\ b_{n-1} & b_{n-1} & \beta_{n-1} \end{bmatrix}, \qquad Y = \begin{bmatrix} b_0 & 0 & 0 \\ b_1 & -b_1 & 0 \\ \cdot & \cdot & \cdot \\ b_{n-1} & -b_{n-1} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where

$$\mathbf{b} = [\beta_0, \beta_1, \ldots, \beta_{n-1}]^T.$$

The triangularization in (19) needs $2n^2 + (n^2/2) + O(n)$ flops (or $n^2 + (n^2/2) + O(n)$ flops with fast rotations). Note that there is no saving in computing $\mathbf{y} = L^{-1}\mathbf{b}$, as above, over the conventional forward elimination method.

Removing back substitution in square systems. From a hardware implementation point of view, the back-substitution step in (20) can be quite cumbersome [17]. This

back-substitution process can also be eliminated by performing the partial triangularization of the matrix [20]

$$A = \begin{bmatrix} B & -\mathbf{b} \\ I & 0 \end{bmatrix} = \begin{bmatrix} L \\ U \end{bmatrix} [L^T, -\mathbf{y}] + \begin{bmatrix} O & 0 \\ O & \mathbf{s} \end{bmatrix}. \tag{21}$$

Notice that the solution $B^{-1}\mathbf{b}$ is the Schur complement of B in A. For Toeplitz B ($b_0 = 1$), a generator of A in (21) with respect to $\{Z_n \oplus Z_n, Z_n \oplus 0\}$ is given by

$$X = \begin{bmatrix} b_0 & 0 & \beta_0 \\ b_1 & b_1 & \beta_1 \\ \cdot & \cdot & \cdot \\ b_{n-1} & b_{n-1} & \beta_{n-1} \\ \mathbf{e}_1 & \mathbf{e}_1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} b_0 & 0 & 0 \\ b_1 & -b_1 & 0 \\ \cdot & \cdot & \cdot \\ b_{n-1} & -b_{n-1} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\mathbf{e}_1 = [1, 0, \ldots, 0]^T$. After n steps of partial triangularization indicated in (21) with $5n^2 + O(n)$ flops (or $3n^2 + O(n)$ flops with fast rotations), we shall have a "generator of the solution vector," from which we can read out the solution after certain normalizations; see [29] for details.

Solving least-squares problems without back substitution. To solve the weighted least-squares problem of minimizing

$$\| B_2(B_1\mathbf{x} - \mathbf{b}) \|_2,$$

where B_1 and B_2 are full-rank block-Toeplitz or Toeplitz-block matrices, we form the matrix

$$A = \begin{bmatrix} -B_2 & B_1 & -\mathbf{b} \\ B_1^T & O & 0 \\ O & I & 0 \end{bmatrix}. \tag{22}$$

Now notice that the least-squares solution

$$\mathbf{x} = (B_1^T B_2^{-1} B_1)^{-1} B_2^T \mathbf{b} \tag{23}$$

is the Schur complement of the submatrix

$$\begin{bmatrix} -B_2 & B_1 \\ B_1^T & O \end{bmatrix}.$$

The displacement rank of the matrix A in (22) is five. After $m + n$ steps of the generalized Schur algorithm, we shall have the solution (23) [see [29] for a generator of (22)].

Regularization. If the given Toeplitz least-squares system is particularly ill conditioned, it is meaningless to compute the exact (least-squares) solution, because small perturbations of the matrix can cause very large perturbations in the solution. In such cases, we may solve the following regularized system [19], [36], [39], [44]

$$\begin{bmatrix} B \\ \eta L \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}, \quad B \in \mathbf{R}^{m \times n}, \quad L\text{: lower triangular banded Toeplitz matrix.}$$

This can be done by partial triangularization of the matrix

$$A = \left[\begin{array}{cc|cc} & & B & b \\ & -I & \eta L & 0 \\ \hline B^T & \eta L & O & 0 \\ O & & I & 0 \end{array}\right]. \tag{24}$$

The matrix A in (24) has displacement rank five. After $m + 2n$ steps of the generalized Schur algorithm, we shall have the solution. We may remark that this technique of regularization is known as the *leakage method* (adding white (if $L = I$) or colored (if L is banded) noise with variance η^2 to the data sample) in the signal processing literature.

Generalized Gohberg–Semencul formulas ([22], [24]). Generalized Gohberg–Semencul formulas for the matrices

$$(T^TT)^{-1}, \quad T_2^TT_1^{-1}T_2, \quad I - T(T^TT)^{-1}T^T, \quad (T^TT)^{-1}T^T,$$

can be obtained after partial triangularization of the (1,1) block of the matrices

$$\begin{bmatrix} T^TT & I \\ I & O \end{bmatrix}, \quad \begin{bmatrix} T_1 & T_2 \\ T_2^T & O \end{bmatrix}, \quad \begin{bmatrix} T^TT & T^T \\ T & I \end{bmatrix}, \quad \begin{bmatrix} T^TT & T^T \\ I & O \end{bmatrix}$$

using the generalized Schur algorithm (see [14], [26], [28], [31], [42] for related results).

5. Concluding remarks. We have generalized earlier definitions of the displacement for Toeplitz-like matrices and presented a correspondingly generalized Schur algorithm for obtaining their triangular factors and their displacement representations. Derived matrices obtained as products and inverses of Toeplitz matrices can be nicely handled by formulating them as Schur complements of entries in a suitably defined Toeplitz-block matrix. The extended definition allows us to efficiently handle block-Toeplitz and Toeplitz-block matrices and Schur complements with respect to the leading (block) entries of such matrices. Some interesting examples were given in § 4. Although the result that displacement rank is not increased under Schur complementation has been know for over a decade (see [5], [13], [37]), the failure to use a generalized definition of displacement made further analysis more cumbersome; similarly cumbersome were the efforts to find expressions for the generators of derived matrices such as $T_1 - T_2T_3^{-1}T_4$.

We may note that appropriate modifications of the above approach can be used to study Hankel, Vandermonde, Hilbert, and Cauchy matrices and derived matrices (see [8]). Among earlier studies of such matrices, we may mention [13], [23], [26], [32]–[34].

Finally, we remark that numerical stability issues are not examined here; studies are in progress on appropriate modifications that can improve the stability.

Appendix. Given the *displacement* of a matrix, we can obtain a generator of the matrix by representing each pair of nonzero columns and rows that cross at the main diagonal as a sum of two rank-one matrices. More precisely, the following procedure can be used to find a (possibly nonminimal) generator from the given displacement with $O(mn)$ flops.

Finding a generator
Input: The displacement $\nabla_{(F^f,F^b)}A$
Output: A generator $\{X, Y\}$ of A
Procedure:
  $X := \{\ \}$; $Y := \{\ \}$;
  while there is nonzero column or row
    for each pair of a column $\mathbf{u}$ and a row $\mathbf{v}^T$ that crosses in the ith position of the main diagonal of $\nabla_{(F^f,F^b)}A$
      if $u_i \neq 0$ **then**
        $\tilde{\mathbf{u}} := \mathbf{u}/u_i^{1/2}$; $\bar{\mathbf{u}} := \mathbf{u}$ except $\bar{u}_i = 0$;
        $\tilde{\mathbf{v}} := \mathbf{v}/u_i^{1/2}$; $\bar{\mathbf{v}} := \mathbf{v}$ except $\bar{v}_i = 0$;

else
$\tilde{\mathbf{u}} := \mathbf{u}$ except $\tilde{u}_i = 1/2$; $\bar{\mathbf{u}} := \mathbf{u}$ except $\bar{u}_i = -1/2$;
$\tilde{\mathbf{v}} := \mathbf{v}$ except $\tilde{v}_i = 1/2$; $\bar{\mathbf{v}} := \mathbf{v}$ except $\bar{v}_i = -1/2$;
end;
$X := [X, \tilde{\mathbf{u}}, \bar{\mathbf{u}}]$; $Y := [Y, \tilde{\mathbf{v}}, -\bar{\mathbf{v}}]$;
Remove **u** and **v**;
end;
for each an unpaired ith column **u**
$X := [X, \mathbf{u}]$; $Y := [y, \mathbf{e}_i]$;
Remove **u** and **v**;
end
for each an unpaired jth row $\mathbf{v}^T$
$X := [X, \mathbf{e}_j]$; $Y := [Y, \mathbf{v}]$;
Remove **u** and **v**;
end
return $\{X, Y\}$

REFERENCES

[1] R. ACKNER AND T. KAILATH, *The Schur algorithm for matrix valued meromorphic functions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 140–150.

[2] H. AHMED, J. DELOSME, AND M. MORF, *Highly concurrent computing structures for matrix arithmetic and signal processing*, Computer, (1982), pp. 65–82.

[3] H. AKAIKE, *Block Toeplitz matrix inversion*, SIAM J. Appl. Math., 24 (1973), pp. 234–241.

[4] E. BAREISS, *Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices*, Numer. Math., 13 (1969), pp. 404–424.

[5] R. BITMEAD AND B. D. O. ANDERSON, *Asymptotically fast solution of Toeplitz and related systems of linear equations*, Linear Algebra Appl., 34 (1980), pp. 103–116.

[6] A. BOJANCZYK, R. BRENT, AND F. DE HOOG, *QR factorization of Toeplitz matrices*, Numer. Math., 49 (1986), pp. 81–94.

[7] J. CHUN, *Fast Array Algorithms for Structured Matrices*, Ph.D. thesis, Stanford University, Stanford, CA, 1989.

[8] J. CHUN AND T. KAILATH, *Fast triangularization and orthogonalization of Hankel and Vandermonde matrices*, Linear Algebra Appl., 151 (1991), pp. 199–228.

[9] J. CHUN, T. KAILATH, AND H. LEV-ARI, *Fast parallel algorithms for QR and triangular factorization*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 899–913.

[10] G. CYBENKO, *A generalized orthogonalization technique with applications to time series analysis and signal processing*, Math. Comp., 40 (1983), pp. 323–336.

[11] ———, *Fast Toeplitz orthogonalization using inner products*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 734–740.

[12] G. CYBENKO AND M. BERRY, *Hyperbolic Householder algorithms for factoring structured matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 499–520.

[13] J. DELOSME, *Algorithms for Finite Shift-Rank Process*, Ph.D. thesis, Stanford University, Stanford, CA, 1982.

[14] J. DELOSME AND M. MORF, *Mixed and minimal representations for Toeplitz and related systems*, in Proc. 14th Asilomar Conf. on Circuits, Systems and Computers, Monterey, CA, 1980, pp. 19–24.

[15] ———, *Normalized doubling algorithms for finite shift-rank processes*, in Proc. 20th IEEE Conf. on Decision and Control, San Diego, CA, 1981, pp. 346–348.

[16] P. DELSARTE, Y. GENIN, AND Y. KAMP, *A polynomial approach to the generalized Levinson algorithm based on the Toeplitz distance*, IEEE Trans. Inform. Theory, IT-29 (1983), pp. 268–278.

[17] E. DEPRETTERE AND K. JAINANDUNSING, *On the design and partitioning of dedicated arrays for solving sets of linear equations without back-substitution*, Tech. Report, Dept. of Electrical Engineering, Delft University of Technology, the Netherlands, 1986.

[18] P. DEWILDE AND H. DYM, *Lossless chain scattering matrices and optimum prediction: the vector case*, Circuit Theory Appl., 9 (1981), pp. 135–175.

[19] L. ELDEN, *An algorithm for the regularization of ill-conditioned least squares problems*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 237–254.

[20] V. FADDEEVA, *Computational Methods of Linear Algebra*, Dover Publications Inc., New York, 1959.
[21] B. FRIEDLANDER, M. MORF, T. KAILATH, AND L. LJUNG, *New inversion formula for matrices classified in terms of their distance from Toeplitz matrices*, Linear Algebra Appl., 27 (1979), pp. 31–60.
[22] I. GOHBERG AND I. FEL'DMAN, *Convolution equations and projection methods for their solutions, Translations of Mathematical Monographs*, Vol. 41, Amer. Math. Soc., Providence, RI, 1974.
[23] I. GOHBERG, T. KAILATH, I. KOLTRACHT, AND P. LANCASTER, *Linear complexity parallel algorithms for linear systems of equations with recursive structure*, Linear Algebra Appl., 88 (1987), pp. 271–315.
[24] I. GOHBERG AND A. SEMENCUL, *On the inversion of finite Toeplitz matrices and their continuous analogs*, Mat. Issled., 2 (1972), pp. 201–233.
[25] G. GOLUB AND C. VAN LOAN, *Matrix Computation*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[26] G. HEINIG AND K. ROST, *Algebraic methods for Toeplitz-like matrices and operators*, Akademie-Verlag, Berlin, 1984.
[27] T. KAILATH, *Signal processing applications of some moment problems*, in Moments in Mathematics, Proc. Sympos. Appl. Math., San Antonio, TX, Amer. Math. Soc., 37 (1987), pp. 71–109.
[28] T. KAILATH AND J. CHUN, *Generalized Gohberg–Semencul formulas for matrix inversion*, in Proc. Sympos. Operator Theory and Applications, Calgary, Canada, 1988, pp. 231–246.
[29] T. KAILATH, J. CHUN, AND V. ROYCHOWDURY, *Systolic array for solving Toeplitz systems of equations*, in Spectral Analysis in One or Two Dimensions, S. Prasad and R. Kashyap, eds., Vedams Book Internat., New Delhi, India, 1990.
[30] T. KAILATH, S. KUNG, AND M. MORF, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407; also Bull. Amer. Math. Soc., 1 (1979), pp. 769–773.
[31] T. KAILATH, A. VIEIRA, AND M. MORF, *Inverses of Toeplitz operators, innovations, and orthogonal polynomials*, SIAM Rev., 20 (1978), pp. 106–119.
[32] H. LEV-ARI, *Nonstationary Lattice-Filter Modeling*, Ph.D. thesis, Stanford University, Stanford, CA, 1983.
[33] H. LEV-ARI AND T. KAILATH, *Lattice filter parameterizations and modeling of nonstationary process*, IEEE Trans. Inform. Theory, IT-30 (1984), pp. 2–16.
[34] ———, *Triangular factorization of structured Hermitian matrices*. I. Schur methods in operator theory and signal processing, in Oper. Theory: Adv. Appl., Vol. 18, Birkhäuser, Basel, Boston, 1986, pp. 301–324.
[35] N. LEVINSON, *The Wiener RMS error criterion in filter design and prediction*, J. Math. Phys., 25 (1947), pp. 261–278.
[36] S. LJUNG AND L. LJUNG, *Fast numerical solution of Fredholm integral equations with stationary kernels*, BIT, 22 (1982), pp. 54–72.
[37] M. MORF, *Doubling algorithms for Toeplitz and related equations*, Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing, Denver, CO, 1980, pp. 954–959.
[38] D. PAL, *Fast Triangular Factorization of Hermitian Toeplitz and Related Matrices with Arbitrary Rank Profile*, Ph.D. thesis, Stanford University, Stanford, CA, 1989.
[39] H. RUTISHAUSER, *Once again: The least squares problem*, Linear Algebra Appl., 1 (1968), pp. 471–478.
[40] I. SCHUR, *Über Potenzreihen, die im Innern des Einheitskreises beschrankt sind*, J. für die Reine und Angewandte Mathematik, 147 (1917), pp. 205–232. English translation appears in Operator Theory: Adv. Appl., Vol 18, Birkhäuser, Basel, Boston, 1986, pp. 31–60.
[41] D. SWEET, *Fast Toeplitz orthogonalization*, Numer. Math., 43 (1984), pp. 1–21.
[42] W. TRENCH, *An algorithm for inverse of finite Toeplitz matrices*, J. SIAM, 12.3 (1964), pp. 515–522.
[43] E. TYRTYSHNIKOV, *Fast algorithms for block Toeplitz matrices*, Soviet J. Numer. Anal. Math. Modelling, 1 (1985), pp. 121–139.
[44] V. VOEVODIN AND E. TYRTYSHNIKOV, *Numerical methods of solution of problems with Toeplitz type matrices*, Zh. Vychisl. Mat. i Mat. Fiz., 21 (1981), pp. 531–544.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 129–133, January 1994

009

ON THE CONTROLLABILITY OF MATRIX PAIRS (A, K) WITH K POSITIVE SEMIDEFINITE, II*

DAVID CARLSON†

Abstract. The subject of the previous article by David Carlson, B. N. Datta, and Hans Schneider [*SIAM Journal on Algebraic and Discrete Methods*, 5 (1984) pp. 346–350] is revisited to improve and clarify results given there and elsewhere.

Key words. controllability, Lyapunov matrix maps

AMS subject classifications. 15A24, 15A18

Introduction. For notation and terminology, we refer to the previous article [CDS]. In particular, A, H, and K are $n \times n$ complex matrices and H and K are hermitian. The following basic result, due independently to Chen [C] and Wimmer [Wi], has had many useful consequences (cf. [CD 1979a], [CD 1979b], [D]).

THEOREM A. *Suppose that* $K = AH + HA^* \geqq 0$. *If* (A, K) *is controllable, then* $\delta(A) = 0$ *and* H *is nonsingular* (*and* $\pi(H) = \pi(A)$, $\upsilon(H) = \upsilon(A)$).

In particular, a search for a converse to this theorem led to [CDS]. The principal result of that article, Theorem 4, is stated below as Theorem B.

If $\Delta(A) = \prod_{i,j=1}^{n} (\lambda_i + \bar{\lambda}_j)$, then the map $L_A(H) = AH + HA^*$ is one to one if and only if (iff) $\Delta(A) \neq 0$. Note that $\Delta(A) \neq 0$ implies that $\delta(A) = 0$ but not conversely.

We will often assume that A is a block-diagonal matrix,

$$A = \operatorname{diag}(A_{11}, \ldots, A_{pp}), \quad \text{with } A_{11}, \ldots, A_{pp} \text{ square.} \tag{1}$$

Under (1), we will assume that matrices $H = [H_{ij}]$ and $K = [K_{ij}]$ are partitioned conformably with A and will define $\hat{H} = \operatorname{diag}(H_{11}, \ldots, H_{pp})$ and $\hat{K} = \operatorname{diag}(K_{11}, \ldots, K_{pp})$. Observe that if $L_A(H) = K$, then also $L_A(\hat{H}) = \hat{K}$.

THEOREM B. *Let* A *be a block-diagonal matrix, as in* (1), *and suppose that* $\delta(A) = 0$ *and*

$$\sigma(A_{ii}) \cap \sigma(A_{jj}) = \varnothing, \qquad i, j = 1, \ldots, p, \ i \neq j. \tag{2}$$

Suppose that $K = AH + HA^* \geqq 0$. Then the following are equivalent:

(3a) (A, K) is controllable,
(3b) $(A, \hat{K})$ is controllable,
(4a) H is nonsingular and $(A^*, H^{-1}K)$ is controllable,
(4b) $\hat{H}$ is nonsingular,
(5a) $x^*Hx \neq 0$ for every eigenvector x of A^*,
(5b) $x^*\hat{H}x \neq 0$ for every eigenvector x of A^*.

We shall reprove and improve the lemmas in [CDS] that lead to Theorem B. We shall show that (3a), (3b), and (4a) are equivalent whenever $K = AH + HA^* \geqq 0$ (the assumptions that $\delta(A) = 0$ and that A is written in block-diagonal form are not necessary)

* Received by the editors December 10, 1990; accepted for publication (in revised form) March 21, 1992. This research was conducted at the Technion-Israel Institute of Technology, Haifa, Israel and was supported by the National Science Foundation grant DMS-8808237 and by the Technion.

† Mathematical Sciences Department, San Diego State University, San Diego, California 92182 (carlson@math.sdsu.edu).

and present an alternate form of Theorem B for block-diagonal A without the general assumption that $\delta(A) = 0$.

Given a matrix B, not necessarily square, the range of B is denoted in [CDS] by Im B and the controllability space of (A, B) (the smallest A-invariant space containing Im B) by $C(A, B)$. We shall emphasize here common aspects of these two concepts by denoting the range of B by $R(B)$ and the controllability space of (A, B) by $R_A(B)$. Note that always $R(B) \subseteq R_A(B)$.

The following result appears as Corollary II.2 of [CS]; we shall extend it to a bound on dim $R_A(K)$.

THEOREM C. *Suppose that* $K = AH + HA^* \geqq 0$. *Then*

$$\operatorname{rank}(K) \leqq \pi(A) + \nu(A) + \sum_{i=1}^{q} [\delta_i/2],$$

where $\delta_1, \ldots, \delta_q$ *are the degrees of the elementary divisors associated with imaginary eigenvalues of* A *and* $[x]$ *is the floor or greatest integer function.*

Finally, we shall improve the following result, which appeared as Theorem 4 of [CD 1979a].

THEOREM D. *Suppose that* $AH + HA^* = HBB^*H$ *and that*

(6) (A^*, B) *is controllable.*

Then the following are equivalent:

(7) H *is nonsingular,*

(8) (A, HB) *is controllable.*

Results on controllability spaces. In the proof of Lemma 1 of [CDS], the role of the critical assumption (2) is not made explicit. That role is explicit in the proof of our Proposition 1. It is closely related to Proposition 0.4 of [Wo] and provides a proof of Exercise 1.5 of [Wo].

PROPOSITION 1. *Let* A *be block diagonal as in* (1) *and suppose that* (2) *holds. Let* $B = (B_i)$ *be partitioned conformably with* A. *Then* $R_A(B) = \oplus_{i=1}^{p} R_{A_{ii}}(B_i)$.

Proof. It is clear that $R_A(B) \subseteq \oplus_{i=1}^{p} R_{A_{ii}}(B_i)$. To complete the proof, it is sufficient to show that for $i = 1, \ldots, p$,

$$0 \oplus \cdots \oplus 0 \oplus R_{A_{ii}}(B_i) \oplus 0 \oplus \cdots \oplus 0 \subseteq R_A(B).$$

By simultaneous permutation of the blocks of A and B, it is sufficient to show this for $i = 1$.

As $\sigma(A_{11}) \cap \sigma(\operatorname{diag}(A_{22}, \ldots, A_{pp})) = \varnothing$ by (2) there exists a polynomial $f(\lambda)$ over C for which $f(A_{11}) = I, f(\operatorname{diag}(A_{22}, \ldots, A_{pp})) = 0$. Thus,

$$f(A)B = \begin{bmatrix} I & & \\ & O & \\ & & \ddots \\ & & & 0 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{bmatrix} = \begin{bmatrix} B_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

so that $R(B_1 \oplus 0 \oplus \cdots \oplus 0) \subseteq R_A(B)$. Similarly, for $j = 1, 2, \ldots,$

$$f(A)A^jB = \begin{bmatrix} I & & \\ & O & \\ & & \ddots \\ & & & 0 \end{bmatrix} \begin{bmatrix} A_{11}^j B_1 \\ A_{22}^j B_2 \\ \vdots \\ A_{pp}^j B_p \end{bmatrix} = \begin{bmatrix} A_{11}^j B_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

so that $R(A_{11}^j B_1 \oplus 0 \oplus \cdots \oplus 0) \subseteq R_A(B)$. The result follows. □

It is clear from Proposition 1 that if A is block diagonal as in (1), $B = (B_i)$ and $C = (C_i)$ are partitioned conformably with A and $R(C_i) \subseteq R(B_i)$, $i = 1, \ldots, p$, then

$$R_A(C) = \bigoplus_{i=1}^{b} R_{A_{ii}}(C_i) \subseteq \bigoplus_{i=1}^{b} R_{A_{ii}}(B_i) = R_A(B).$$

This is true even if $R(C) \not\subseteq R(B)$, e.g., take

$$A = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & -1 \end{array}\right), \quad B = \left(\begin{array}{c|c} 1 & 1 \\ \hline 1 & 1 \end{array}\right), \quad C = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}\right).$$

It is known [A] that for $K = (K_{ij}) \geqq 0$,

$$R(K_{ii}) \supseteq R(K_{ij}), \quad i, j = 1, \ldots, p, \quad i \neq j.$$

Let $K_i = (K_{i1}, \ldots, K_{ii}, \ldots, K_{ip})$ and $\hat{K}_i = (0, \ldots, 0, K_{ii}, 0, \ldots, 0)$ for $i = 1, \ldots, p$; then

$$R(\hat{K}_i) = R(K_{ii}) = R(K_i),$$

so we have $R_A(\hat{K}) = R_A(K)$, thus completing the proof of Lemma 1, our version of Lemma 1 of [CDS].

LEMMA 1. *Let A be block diagonal as in* (1) *and suppose that* (2) *holds. For $K \geqq 0$, $R_A(\hat{K}) = R_A(K)$, and* (3a) *and* (3b) *are equivalent.*

We next show that the best possible bound on the rank of $K = AH + HA^* \geqq 0$ given in Theorem C extends to a bound on the dimension of $R_A(K)$. The proof of Theorem C in [CS] extends easily to show that the bound (9) is also best possible.

THEOREM 1. *Suppose that $K = AH + HA^* \geqq 0$. Then*

$$\text{rank } K \leqq \dim (R_A(K)) \leqq \pi(A) + \upsilon(A) + \sum_{i=1}^{q} [\delta_i/2]. \tag{9}$$

Proof. We follow and extend the proof of Corollary II.2 in [CS]. In this proof, $A = \text{diag}\,(A_{11}, \ldots, A_{q+1,q+1})$, where $A_{11}, \ldots, A_{q+1,q+1}$ are square, and, for $i = 1, \ldots, q$, A_{ii} is a single upper-triangular Jordan block of order δ_i associated with an imaginary eigenvalue of A. With $H = [H_{ij}]$ and $K = [K_{ij}]$ partitioned conformably, for $i = 1, \ldots, q$, K_{ii} has by Theorem II of [CS] at most $[\delta_i/2]$ nonzero rows; the bottom $\delta_i - [\delta_i/2]$ rows are zero.

Because $K \geqq 0$, this must also be true for $K_i = [K_{i1}, \ldots, K_{i,q+1}]$. And because for $j = 1, 2, \ldots$, A^j has conformable block-diagonal form, with upper-triangular diagonal blocks, this same statement about zero and possibly nonzero rows holds for $A^j K_i$. □

Suppose $K = AH + HA^* \geqq 0$ for some $H \neq 0$. As noted in the proof of Theorem IV of [CS], there exists a nonsingular S for which

$$SAS^{-1} = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}, \qquad SHS^* = \begin{pmatrix} H_{11} & 0 \\ 0 & 0 \end{pmatrix},$$

with H_{11} nonsingular, and then

$$SKS^* = (SAS^{-1})(SHS^*) + (SHS^*)(SAS^{-1})^* = \begin{pmatrix} A_{11}H_{11} + H_{11}A_{11}^* & 0 \\ 0 & 0 \end{pmatrix}.$$

It follows that (see also Lemma 3 of [CL])

$$R_A(K) \subseteq R(H), \tag{10}$$

which explains in a structural way the statement in Theorem A that H is nonsingular whenever (A, K) is controllable.

We next explore the case of equality in (10). Suppose first that A has a single eigenvalue, which is imaginary. Then by the argument leading to (10) and Theorem 1 applied to A_{11}, H_{11}, and K_{11}, if $K = AH + HA^* \geqq 0$, then whenever $H \neq 0$ we cannot have $R_A(K) = R(H)$. That is to say, $R_A(K) = R(H)$ iff $H = 0$.

We can now generalize Lemma 2 of [CDS]. A decomposition similar to that in our Lemma 2 appears in Theorem 4.7 of [HS].

LEMMA 2. *Let A be block diagonal as in* (1), *with* (2) *and*

(11) *either $\Delta(A_{ii}) \neq 0$, or A_{ii} has a single eigenvalue that is imaginary, $i = 1, \ldots, p$,*

holding. Suppose $K = AH + HA^* \geqq 0$.
Then $R_A(K) \subseteq R(\hat{H})$, with equality iff $H_{ii} = 0$ for all $i = 1, \ldots, p$ for which A_{ii} has a single imaginary eigenvalue.

Proof. Because $K = L_A(H) \geqq 0$, then also $\hat{K} = L_A(\hat{H}) \geqq 0$, so that

$$R_A(K) = R_A(\hat{K}) \subseteq R(\hat{H})$$

by Lemma 1 and (10). As

$$R_A(\hat{K}) = \bigoplus_{i=1}^{p} R_{A_{ii}}(K_{ii}) \quad \text{and} \quad R(\hat{H}) = \bigoplus_{i=1}^{p} R(H_{ii}),$$

the statement on equality follows from Corollary 2 of [CL] for A_{ii} with $\Delta(A_{ii}) \neq 0$ and from our remarks above for A_{ii} with a single imaginary eigenvalue. □

Note that every square complex matrix is similar to a block-diagonal matrix as in (1), with (2) and (11) holding.

Note also that Lemma 2 may be regarded as a more structural explanation of another part of Theorem A: if (A, K) is controllable, then so is $(A, \hat{K})$, and by (10) $\hat{H}$ is nonsingular, and A can have no imaginary eigenvalue.

Results on controllability. We first strengthen Theorem D.

THEOREM 2. *Suppose $AH + HA^* = HBB^*H$. Then* (8) *((A, HB) is controllable) iff* (6) *((A^*, B) is controllable) and* (7) *(H is nonsingular).*

Proof. That (6) and (7) together imply (8) is part of Theorem D. Suppose now that (8) holds, then also (A, HBB^*H) is controllable and $HBB^*H \geqq 0$.
It follows from Theorem A that H is nonsingular. Now

$$H^{-1}A + A^*H^{-1} = H^{-1}(HBB^*H)H^{-1} = BB^* \geqq 0$$

and, applying Theorem D again, $(A^*, H^{-1}(HB) = B)$ is controllable. □

Observe that (6) does not imply (7) or (8) (take $H = 0$) and that (7) does not imply (6) or (8) (take $A = 0$ and $B = 0$).

We may now reformulate and prove Theorem B without the assumption that $\delta(A) = 0$.

THEOREM 3. *Let A be block diagonal as in* (1), *with* (2) *and* (11) *holding. Suppose that $K = AH + HA^* \geqq 0$. The following are equivalent*: (3a), (3b), (4a), $\delta(A) = 0$ *and* (4b), $\delta(A) = 0$ *and* (5a), $\delta(A) = 0$ *and* (5b).

Proof. The proof in [CDS] that (3a) iff (5a) holds without the assumption that $\delta(A) = 0$.

Recall that (A, B) is controllable iff (A, BB^*) is. Now H is nonsingular if (3a) holds (by Theorem A) or if (4a) holds (by hypothesis). From $H^{-1}(AH + HA^* = K)H^{-1}$ we obtain

$$A^*H^{-1} + H^{-1}A = H^{-1}KH,$$

and now (3a) and (4a) are equivalent by Theorem 2 applied to A^*, H^{-1}, and $K = BB^*$.

The rest of the proof follows immediately from Theorems A and B. □

Acknowledgment. The author wishes to thank the referees, whose comments have significantly clarified the presentation.

REFERENCES

[A] A. ALBERT, *Conditions for positive and nonnegative definiteness in terms of pseudoinverses*, SIAM J. Appl. Math., 17 (1969), pp. 434–440.

[CD 1979a] D. CARLSON AND B. N. DATTA, *The Lyapunov matrix equation* $SA + A^*S = S^*B^*BS$, Linear Algebra Appl., 28 (1979), pp. 43–52.

[CD 1979b] ———, *On the effective computation of the inertia of a non-Hermitian matrix*, Numer. Math., 33 (1979), pp. 315–322.

[CDS] D. CARLSON, B. N. DATTA, AND H. SCHNEIDER, *On the controllability of matrix pairs* (A, K) *with K positive semidefinite*, SIAM J. Alg. Disc. Math., 5 (1984), pp. 346–350.

[CL] D. CARLSON AND R. LOEWY, *On ranges of Lyapunov transformations*, Linear Algebra Appl., 8 (1974), pp. 237–248.

[CS] D. CARLSON AND H. SCHNEIDER, *Inertia theorems for matrices. The semidefinite case*. J. Math. Anal. Appl., 6 (1963), pp. 430–446.

[C] C. T. CHEN, *A generalization of the inertia theorem*, SIAM J. Appl. Math., 25 (1973), pp. 158–161.

[D] B. N. DATTA, *On the Routh-Hurwitz-Fujiwara and the Schur-Cohn-Fujiwara theorems for the root-separation problem*, Linear Algebra Appl., 22 (1978), pp. 235–246.

[HS] D. HERSHKOWITZ AND H. SCHNEIDER, *Semistability factors and semifactors*, Contemp. Math., 47 (1985), pp. 203–216.

[Wi] H. WIMMER, *Inertia theorems for matrices, controllability, and linear vibrations*. Linear Algebra Appl., 8 (1974), pp. 337–343.

[Wo] W. M. WONHAM, *Linear Multivariable Control: Geometric Approach*, Second ed., Springer-Verlag, New York, 1979.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 134–139, January 1994

010

REDUCTION OF A TRANSFER FUNCTION VIA AN OBSERVABILITY MATRIX*

STEPHEN BARNETT†

Abstract. An algorithm is given for reduction of a scalar transfer function $g(s)$ to its lowest terms. The main step is to reduce the observability matrix for a controllable canonical form state-space realization of $g(s)$ to a block-triangular form by row operations. No polynomial manipulations are required and only a single rank computation is needed. As a byproduct, other properties of the numerator and denominator of $g(s)$ are obtained with little extra effort. The method can be extended to the case when a basis of orthogonal polynomials is used.

Key words. transfer function reduction, observability matrix

AMS subject classifications. 15, 93

1. Introduction.

Consider a given proper transfer function

$$g(s) = \frac{b(s)}{a(s)} \tag{1.1}$$

$$= \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^n + a_1 s^{n-1} + \cdots + a_n}, \tag{1.2}$$

where $b_0 \neq 0$ and $t = n - m \geq 1$. The state-space realization of (1.2) in controllable canonical form is

$$\dot{x} = Ax + du, \qquad y = cx, \tag{1.3}$$

where

$$d = [0, 0, \ldots, 0, 1]^T, \qquad c = [b_m, b_{m-1}, \ldots, b_0, 0, \ldots, 0], \tag{1.4}$$

and A is an $n \times n$ companion matrix associated with $a(s)$ in the form

$$A = \begin{bmatrix} 0 & & I_{n-1} \\ -a_n & \cdots & -a_1 \end{bmatrix}, \tag{1.5}$$

where I_{n-1} denotes the unit matrix of order $n - 1$. If the realization (1.3) is completely observable, then $g(s)$ is irreducible. If not, then $g(s)$ can be reduced to the form

$$\frac{\beta(s)}{\alpha(s)} = \frac{\beta_0 s^{m-k} + \beta_1 s^{m-k-1} + \cdots + \beta_{m-k}}{s^{n-k} + \alpha_1 s^{n-k-1} + \cdots + \alpha_{n-k}}, \tag{1.6}$$

where the greatest common divisor $d(s)$ of $a(s)$ and $b(s)$ has degree k. There are, of course, many ways of obtaining the reduced form (1.6): for example, determine $d(s)$ by constructing the Routh array associated with $a(s)$ and $b(s)$, and hence obtain $\alpha(s)$ and $\beta(s)$ by direct division; or, to avoid polynomial manipulations, use the Hankel matrix of Markov parameters [10]. A recent method that also avoids divisions has been suggested by Chui and Chen [11] and involves a Sylvester-type resultant matrix. This approach is interesting, because although it has long been known [2, p. 39] how to compute $d(s)$ from a Sylvester matrix, their algorithm produces the reduced form (1.6) directly by using appropriate row operations, without actually finding $d(s)$ itself. However, it seems to be true that for every algorithm involving a Sylvester matrix, there is a corresponding scheme based on using a companion matrix. The purpose of this article is to show how

* Received by the editors November 26, 1990; accepted for publication (in revised form) March 23, 1992.
† Department of Applied Mathematical Studies, University of Leeds, Leeds LS2 9JT, United Kingdom.

the problem of reducing $g(s)$ in (1.1) to the form (1.6) can also be solved in a "companionable" fashion.

2. The algorithm. There is no loss of generality in assuming from now on that $b_0 = 1$, $\beta_0 = 1$ in (1.2) and (1.6), respectively. The algorithm is as follows.

Step 1. Construct the observability matrix M having rows $c, cA, cA^2, \ldots, cA^{n-1}$, where A and c are defined in (1.3). Notice that the first t rows of M are obtained simply by performing repeated cyclic shifts on c, i.e.,

$$cA^i = [\underbrace{0, 0, \ldots, 0}_{i}, b_m, \ldots, b_1, 1, \underbrace{0, \ldots, 0}_{t-i-1}], \quad i = 0, 1, \ldots, t-1.$$

Step 2. Apply elementary row operations to the last m rows of the $n \times 2n$ matrix

$$X = [M, I_n] \tag{2.1}$$

so as to reduce it to $[X_1, X_2]$, where the $n \times n$ matrix X_1 has the block-triangular form

$$X_1 = \begin{matrix} \\ t \\ m \end{matrix}\!\!\begin{matrix} \begin{matrix} m & t \end{matrix} \\ \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & 0 \end{bmatrix} \end{matrix}. \tag{2.2}$$

In (2.2) the first t rows of X_1 are precisely the first t rows of M constructed in Step 1, so in particular X_{12} is lower triangular and X_{21} is upper triangular relative to its secondary (northeast to southwest) diagonal.

Notice that when $t = 1$, then X_1 is itself upper triangular in this latter sense.

Step 3. The matrix X_{21} is nonsingular if and only if $g(s)$ in (1.1) is irreducible, so if X_{21} in (2.2) has no zero rows, then no reduction of $g(s)$ is possible. Otherwise, make the last k rows of X_{21} zero, and row $n-k+1$ of X_2 is then by Corollary 1.2 of [3]

$$[\alpha_{n-k}, \alpha_{n-k-1}, \ldots, \alpha_1, 1, 0, \ldots, 0],$$

which gives the required coefficients of the denominator in (1.6).

Step 4. Construct the triangular Hankel matrix

$$W = \begin{bmatrix} 0 & & & 0 & 1 \\ & & \iddots & 1 & w_1 \\ & \iddots & \iddots & \iddots & \vdots \\ 0 & 1 & w_1 & \cdots & w_{m-k-1} \\ 1 & w_1 & w_2 & \cdots & w_{m-k} \end{bmatrix} \tag{2.3}$$

of order $m-k+1$, where

$$w_j = -\sum_{i=1}^{j} a_i w_{j-i}, \quad w_0 = 1; \quad j = 1, 2, \ldots, m-k. \tag{2.4}$$

Step 5. The required coefficients of the numerator in (1.6) are given by

$$[\beta_{m-k}, \beta_{m-k-1}, \ldots, \beta_1, 1] = [\alpha_{m-k}, \alpha_{m-k-1}, \ldots, \alpha_1, 1]WT, \tag{2.5}$$

where T is the triangular Hankel matrix

$$T = \begin{bmatrix} b_{m-k} & b_{m-k-1} & \cdots & & b_1 & 1 \\ b_{m-k-1} & b_{m-k-2} & \cdots & b_1 & 1 & 0 \\ \vdots & \vdots & \iddots & \iddots & \iddots & \vdots \\ b_1 & 1 & \iddots & & 0 & \\ 1 & 0 & & & & \end{bmatrix}. \tag{2.6}$$

Proof. Steps 1, 2, and 3 are derived in [3] in a more general setting. The theory behind Steps 4 and 5 is developed in detail in [7] and relies on establishing a relationship between $b(A)$ and $a(B)$, where B is a companion matrix for $b(s)$. □

The procedure can in fact still be applied if the degrees of $b(s)$ and $a(s)$ are equal: simply replace the numerator in (1.1) by $b(s) - a(s)$, so that the vector c in (1.4) used in Step 1 becomes

$$[b_n - a_n, b_{n-1} - a_{n-1}, \ldots, b_1 - a_1] \tag{2.7}$$

and select the appropriate value of t in (2.2).

3. Illustrative examples.

Example 1. Consider the example used in [11] with $n = 4$, namely,

$$g(s) = \frac{s^4 + \frac{3}{2}s^3 + 2s^2 + s + \frac{1}{2}}{s^4 - \frac{1}{2}s^3 + 2s^2 + \frac{1}{2}s + 1}. \tag{3.1}$$

From (2.7) we have

$$c = [-\tfrac{1}{2}, \tfrac{1}{2}, 0, 2],$$

and from (1.5) the companion matrix for the denominator in (3.1) is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -\frac{1}{2} & -2 & \frac{1}{2} \end{bmatrix}.$$

After constructing the observability matrix M in Step 1, the matrix in (2.1) becomes

$$X = \left[\begin{array}{cccc:c} -\frac{1}{2} & \frac{1}{2} & 0 & 2 & \\ -2 & -\frac{3}{2} & -\frac{7}{2} & 1 & \\ -1 & -\frac{5}{2} & -\frac{7}{2} & -3 & I_4 \\ 3 & \frac{1}{2} & \frac{7}{2} & -5 & \end{array}\right].$$

M

Using appropriate row operations, this is easily reduced to the stated form in Step 2:

$$\left[\begin{array}{cccc:cccc} -\frac{1}{2} & \frac{1}{2} & 0 & 2 & 1 & 0 & 0 & 0 \\ -\frac{7}{4} & -\frac{7}{4} & -\frac{7}{2} & 0 & -\frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 \end{array}\right]. \tag{3.2}$$

X_1 $\qquad$ X_2

Because the last two rows of X_1 in (3.2) are zero, it follows from Step 3 that $k = 2$, and hence row $n - k + 1 = 3$ of X_2 gives the denominator in (1.6) as $\alpha(s) = s^2 - s + 2$.

Using (2.4), the matrix in (2.3) is easily found to be

$$W = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -\frac{7}{4} \end{bmatrix}$$

and from (2.6)

$$T = \begin{bmatrix} 2 & \frac{3}{2} & 1 \\ \frac{3}{2} & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Finally, from (2.5) we have

$$[\beta_2, \beta_1, 1] = [2, -1, 1]WT$$
$$= [1, 1, 1]$$

so that the numerator in (1.6) is $\beta(s) = s^2 + s + 1$.

It should be noted that determination of k from X_1 is equivalent to finding the rank of M; in the solution of the reduction problem in [11], three separate rank calculations are necessary.

Example 2. Consider

$$g(s) = \frac{s^3 + 8s^2 + s - 42}{s^5 + 10s^4 + 22s^3 + 4s^2 - 23s - 14}$$

for which $n = 5$ and $t = n - m = 2$. Using A in (1.5) and Step 1, the matrix in (2.1) is found to be

$$X = \left[\begin{array}{ccccc:c} -42 & 1 & 8 & 1 & 0 & \\ 0 & -42 & 1 & 8 & 1 & \\ 14 & 23 & -46 & -21 & -2 & I_5 \\ -28 & -32 & 31 & -2 & -1 & \\ -14 & -51 & -28 & 53 & 8 & \end{array}\right].$$
$$M$$

This is reduced by row operations to the required form in Step 2 with

$$X_1 = \left[\begin{array}{ccc:cc} -42 & 1 & 8 & 1 & 0 \\ 0 & -42 & 1 & 8 & 1 \\ \hline -196 & -56 & -4 & 0 & 0 \\ 1008 & 144 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{3.3}$$
$$X_{21}$$

$$X_2 = \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 5 & 2 & 1 & 0 & 0 \\ -26 & -7 & -4 & 1 & 0 \\ -2 & -3 & 1 & 3 & 1 \end{array}\right]. \tag{3.4}$$

It follows from Step 3 by inspection of X_1 that $k = 1$, and from row $n - k + 1 = 5$ of X_2 the denominator in (1.6) is

$$\alpha(s) = s^4 + 3s^3 + s^2 - 3s - 2.$$

The above part of the example is essentially as worked out in [3]. To determine $\beta(s)$, from (2.3), (2.4), and (2.6) we have

$$W = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -10 \\ 1 & -10 & 78 \end{bmatrix}, \qquad T = \begin{bmatrix} 1 & 8 & 1 \\ 8 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

so that in (2.5)

$$[\beta_2, \beta_1, 1] = [1, 3, 1]WT = [-6, 1, 1],$$

whence the numerator in (1.6) is $\beta(s) = s^2 + s - 6$.

For the purposes of this article, only row $n - k + 1$ of X_2 needs to be recorded. However, it should be pointed out that the other rows of X_1 do in fact give the coefficients of the set of Euclidean remainders associated with $a(s)$ and $b(s)$, as described in [6]. In particular, a greatest common divisor $d(s)$ of $a(s)$ and $b(s)$, if required, is given by the last nonzero row in X_1. Thus, in Example 1, from (3.2) we have

$$d(s) = -\tfrac{7}{2}s^2 - \tfrac{7}{4}s - \tfrac{7}{4},$$

and, for Example 2, (3.3) gives

$$d(s) = 144s + 1008. \tag{3.5}$$

Furthermore, without additional effort, the solution $y(s)$ of the diophantine equation

$$a(s)x(s) + b(s)y(s) = d(s) \tag{3.6}$$

can be read off from row $n - k$ of X_2 [4]. Thus, in Example 2, the fourth row of X_2 in (3.4) gives

$$y(s) = s^3 - 4s^2 - 7s - 26,$$

where $d(s)$ is given in (3.5). Finally, following [7], the coefficients of $-x(s)$ in (3.6) are obtained by multiplying the elements in columns $n - m + 1$ to $n - k + 1$ of row $n - k$ of X_2 by WT. In Example 2, this gives

$$[-4, 1, 0]WT = [-6, 1, 0],$$

showing that $x(s) = -s + 6$.

4. Discussion and conclusions. It has been shown how a given transfer function $g(s)$ can be reduced to its lowest terms by performing row operations on the observability matrix of a controllable canonical form realization so as to reduce it to the block-triangular form (2.2). The reduction of the denominator of $g(s)$ was given in [3], but the complete algorithm is detailed above for the first time.

Like the scheme proposed in [11], there are no polynomial manipulations, but the algorithm in § 2 seems to have several advantages.

(i) There is only a single computation of rank, that of the observability matrix M in Step 3, whereas in [11] $k + 1$ separate calculations of rank are needed.

(ii) It is more natural in a control context to use an observability matrix rather than a Sylvester-type matrix. For example, if a minimal realization of $g(s)$ in state-space form is required, then a standard method [8] is to extract the completely observable part of the realization $\{A, d, c\}$ in (1.3) using a similarity transformation. However, recovering the reduced transfer function then requires inversion of a characteristic matrix, and this is difficult in general.

(iii) The procedure described in this article also produces the greatest common divisor between the numerator and denominator of $g(s)$, as well as the solution of a diophantine equation, and the associated Euclidean remainders, with little extra computational effort. It is interesting that the method still gives the greatest common divisor directly, even in cases where the Routh array applies to $\alpha(s)$ and $\beta(s)$ requires modifications. For example, if

$$g(s)0\,\frac{s^3 - s^2 + s - 1}{s^4 - 2s^3 + 2s^2 - s},$$

then the algorithm gives

$$X_1 = \left[\begin{array}{cccc} -1 & 1 & -1 & 1 \\ \hline 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right], \qquad X_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 \end{bmatrix}.$$

From the form of X_{21} we have $k = 1$, and $n - k + 1 = 4$ of X_2 gives the denominator of the reduced form of $g(s)$ as $s - s^2 + s^3$. The last nonzero row of X_1 shows that the greatest common divisor of the numerator and denominator of $g(s)$ is $s - 1$. The reader can easily check that the first column element in the fourth row of the corresponding Routh array is zero.

(iv) If the numerator and denominator of $g(s)$ are expressed relative to a basis of orthogonal polynomials, then the method described in § 2 for finding $\alpha(s)$ carries over with little modification, provided that the companion matrix is replaced by the comrade matrix [2, p. 372]. However, to find $\beta(s)$, it is necessary to reverse the roles of $a(s)$ and $b(s)$, since there is no analogue of Steps 4 and 5 (for details, see [3] and [5]). The method is therefore applicable to the model-reduction problem when, for example, a transfer function is represented as a ratio of Chebyshev polynomial series [1], [9]. There seems to be no corresponding generalization of the Sylvester-type matrix.

REFERENCES

[1] G. A. BAKER, JR AND P. R. GRAVES-MORRIS, *Padé Approximants, Part II: Extensions and Applications*, Addison-Wesley, Reading, MA, 1981, p. 56.

[2] S. BARNETT, *Polynomials and Linear Control Systems*, Marcel Dekker, New York, 1983.

[3] ———, *Division of generalized polynomials using the comrade matrix*, Linear Algebra Appl., 60 (1984), pp. 159–175.

[4] ———, *Solution of ax + by = d for generalized polynomials*, in Proc. 23rd IEEE Conference on Decision and Control, Las Vegas, 1984, pp. 1766–1767.

[5] ———, *Matrix methods for performing algebraic operations on generalized polynomials*, in Current Trends in Matrix Theory, F. Uhlig and R. Grone, eds., North Holland, New York, 1987, pp. 41–44.

[6] ———, *Euclidean remainders and the Routh array from an observability matrix*, Preprints 10th World IFAC Congress on Automatic Control, Vol. 9, Munich, 1987, pp. 34–39.

[7] ———, *On a relationship between two matrix polynomials, with applications*, Mathematical Sciences Report TAM 87-44, University of Bradford, United Kingdom, 1987.

[8] S. BARNETT AND R. G. CAMERON, *Introduction to Mathematical Control Theory*, Second ed., Oxford University Press, Oxford, 1985, p. 116.

[9] 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.

[10] C.-T. CHEN, *Linear System Theory and Design*, Holt, Rinehart and Winston, New York, 1984, p. 245.

[11] C. K. CHUI AND G. CHEN, *An efficient algorithm for order reduction of transfer functions*, in Proc. SIAM Conference on Linear Algebra in Signals, Systems and Control, Boston, 1986.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 140–150, January 1994

011

THE SCHUR ALGORITHM FOR MATRIX-VALUED MEROMORPHIC FUNCTIONS*

REUVEN ACKNER†, HANOCH LEV-ARI‡, AND THOMAS KAILATH†

Dedicated to Gene Golub on the occasion of his 60th birthday, with admiration and appreciation.

Abstract. This article extends the classical Schur algorithm to matrix-valued functions that are bounded on the unit circle and have a finite number of Smith–McMillan poles inside the unit disc. With each such function this article associates two infinite sequences: one is the well-known sequence of reflection coefficients (all less than one in magnitude), whereas the other is a sequence of signs. Under certain assumptions, the number of negative signs equals the number of poles within the unit disc. This article shows how to solve tangential interpolation problems using the algorithm and gives a simple proof for the connection between the number of poles inside the unit disc of each solution to the inertia of a certain Pick matrix. Also described is a numerically efficient procedure for carrying out the algorithm that involves only scalar operations.

Key words. Schur algorithm, tangential interpolation, fast algorithms

AMS subject classifications. 30C15, 30D30, 12D10

1. Introduction.

In 1917, Schur [1] introduced a recursive algorithm for parametrization of scalar functions that are analytic and bounded within the unit disc. Since then, the algorithm has been applied in many fields of engineering and mathematics. Among others, applications include estimation and modeling of stochastic processes [2], stability checking [3], [4], filter design [5], [6], fast algorithms for signal processing [7], and H^∞ control [8]. The survey article [7] contains a more detailed discussion of several of these applications. Since its introduction, Schur's algorithm, has been extended in many directions. One such extension is the Nevanlinna algorithm [9], [10]; another is the *modified* Schur algorithm [11]–[13] for meromorphic functions that have a finite number of poles inside the unit disc.

Yet another extension by Delsarte, Genin, and Kamp [14] is the block Schur algorithm for *analytic* matrix-valued functions. This algorithm cannot be applied to functions with poles, because it requires square roots and some of the matrices in the recursion may become indefinite (see also [4], [15]). Furthermore, this recursion is computationally expensive because it requires matrix operations. A first step in a simplification of the matrix algorithm was noted by Fedčina [16], who introduced the *tangential* or *directional* Schur algorithm; in this version the operations are performed in only one "direction" at a time, for example, row after row. This procedure is more attractive because it saves

* Received by the editors September 3, 1991; accepted for publication April 8, 1992. This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command contract AF88-0327, the U.S. Army Research Office contract DAAL03-89-K-0109, and the National Science Foundation grant MIP86-19169A2. This manuscript is submitted for publication with the understanding that the U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

† Information Systems Laboratory, Stanford University, Stanford, California 94305-4055 (ackner@isl.stanford.edu, levari@neung.coe.northeastern.e, tk@isl.stanford.edu).

‡ Present address, the Department of Electrical and Computing Engineering, Northeastern University, 360 Huntington Avenue, Boston, Massachusetts 02115.

computations. We also note that the tangential Schur algorithm implicitly appeared in the work of Dewilde and Dym [17]. One contribution of our article is in further reducing the operations in each direction to a set of elementary scalar operations. This process of "scalarization" of the algorithm avoids matrix or vector arithmetic, reduces the computation complexity, and provides the maximum degree of freedom in the implementation.

The main contribution of this article is the extension of the Schur algorithm to functions that are both matrix valued and meromorphic, with a finite number of Smith–McMillan poles inside the unit disc. We do so by combining the results of [17] on the matrix Schur algorithm with the earlier work on scalar meromorphic functions [11]–[13]. In the scalar case, functions with poles, or equivalently with reflection coefficients greater than one in magnitude, are handled by "switching" and performing the classical Schur algorithm on the reciprocal function $1/f_i(z)$ instead of on the original function $f_i(z)$, whenever (the ith reflection coefficient) $f_i(0)$ exceeds one in magnitude. This idea does not apply directly in the matrix case because the matrices involved might not even be square; however, scalarization enables us to use the same idea at each scalar step of the algorithm.

Our version of the algorithm produces a sequence of reflection coefficients $\{k_i\}_{i=0}^{\infty}$ with $|k_i| < 1$ and a sequence of signs $\{\varepsilon_i\}_{i=0}^{\infty}$. The latter contains the information on the number of (Smith–McMillan) poles inside the unit disc of the given function. Whenever the sign in the recursion is positive ($\varepsilon_i = 1$), this number does not change; each time the sign is negative ($\varepsilon_i = -1$), the number of poles decreases by one. As a result, the total number of poles of a function is greater than or equal to the number of negative signs in the sequence $\{\varepsilon_i\}_{i=0}^{\infty}$; equality of these two numbers holds under certain additional conditions (see § 2).

As an application, we show how to solve certain tangential interpolation problems [16], [18, ch. 18] using the Schur algorithm. We also give a simple proof of the fact that the minimum number of poles inside the unit disc of each interpolating function is equal to the number of negative eigenvalues of a Pick matrix that is determined by the given data.

Finally, for computational purposes we describe an array formulation of the algorithm, using coefficients of matrix Taylor expansions. The operations applied to the array are (orthogonal and hyperbolic) rotations and shifts.

As in [13], we shall assume regularity in the sense that no reflection coefficient has unit magnitude; nonregular cases have to be treated by a different approach.

2. Modified Schur algorithm. We shall base our discussion on [17] and extend the results therein to the meromorphic case.

Let $F(z)$ be a $p \times q$ matrix of rational functions with

$$\|F\|_{\infty} = \sup_{0 \leqq \theta < 2\pi} \sigma_{\max}(F(e^{j\theta})) \leqq 1,$$

where $\sigma_{\max}(\cdot)$ denotes the maximum singular value of the matrix. We can always write $F(z)$ as a ratio $F(z) = U_0^{-1}(z)V_0(z)$, where $U_0(z)$ and $V_0(z)$ are left coprime matrix polynomials of sizes $p \times p$ and $p \times q$, respectively [19, ch. 6]. We note that this representation is not unique; however, every such representation has the property that the poles of $F(z)$ are exactly the zeros of $U_0(z)$ (including multiplicity). Define the $p \times (p + q)$ generator matrix of $F(z)$ as

$$G_0(z) = [U_0(z) \quad V_0(z)].$$

Also, choose a sequence of constant $1 \times p$ vectors η_i and a sequence $\{z_i\}_0^\infty$ of extraction points inside the unit disc. Then the first step in the ith stage of the Schur recursion of [17] is

$$\bar{G}_i(z) = G_i(z)\Psi_i(z), \qquad i = 0, 1, \ldots, \tag{1a}$$

where

$$G_i(z) = [U_i(z) \quad V_i(z)], \tag{1b}$$

$$\bar{G}_i(z) = [\bar{U}_i(z) \quad \bar{V}_i(z)], \tag{1c}$$

$$\Psi_i(z) = [I_{p+q} + (B_i(z) - 1)J_{pq}\xi_i^*(\xi_i J_{pq}\xi_i^*)^{-1}\xi_i]W_i, \tag{1d}$$

$$B_i(z) = (z - z_i)/(1 - z_i^* z), \tag{1e}$$

$$\xi_i = \eta_i G_i(z_i), \tag{1f}$$

and W_i is any J_{pq} unitary matrix. Because $\Psi_i(z)$ is clearly analytic inside the unit disc, the first step of each stage of the algorithm preserves the analyticity of the generator. It can be also verified by a direct calculation that

$$\eta_i \bar{G}_i(z_i) = 0. \tag{1g}$$

The second step of the ith stage of the Schur algorithm is given by

$$G_{i+1}(z) = \Phi_i^{-1}(z)\bar{G}_i(z), \tag{2a}$$

where $\Phi_i(z)$ is a $p \times p$ Blaschke factor defined as

$$\Phi_i(z) = I_p + (B_i(z) - 1)\eta_i^*(\eta_i\eta_i^*)^{-1}\eta_i. \tag{2b}$$

The matrix

$$\Phi_i^{-1}(z) = I_p + (B_i^{-1}(z) - 1)\eta_i^*(\eta_i\eta_i^*)^{-1}\eta_i$$

is not analytic inside the unit disc because of the simple pole at $z = z_i$. Nevertheless, the resulting generator $G_{i+1}(z)$ in (2a) is analytic at $z = z_i$. To prove this, we rewrite (2a) in the form

$$G_{i+1}(z) = [I_p - \eta_i^*(\eta_i\eta_i^*)^{-1}\eta_i]\bar{G}_i(z) + \eta_i^*(\eta_i\eta_i^*)^{-1}[B_i^{-1}(z)\eta_i\bar{G}_i(z)].$$

Now, using (1g), we can easily see that $G_{i+1}(z)$ is finite at $z = z_i$.

Another property that is preserved by the algorithm is the boundedness of the norm of $F_i(z) = U_i^{-1}(z)V_i(z)$. This follows from the fact that the matrix $\Psi_i(z)$ is J_{pq} paraunitary; hence, $\|U_i^{-1}V_i\|_\infty \leqq 1$ implies that $\|\bar{U}_i^{-1}\bar{V}_i\|_\infty \leqq 1$. Now because

$$F_{i+1}(z) = U_{i+1}^{-1}(z)V_{i+1}(z) = [\Phi_i^{-1}(z)\bar{U}_i(z)]^{-1}[\Phi_i^{-1}(z)\bar{V}_i(z)] = \bar{U}_i^{-1}(z)\bar{V}_i(z),$$

we obtain that if $\|F_i\|_\infty \leqq 1$, then $\|F_{i+1}\|_\infty \leqq 1$.

Steps (1) and (2), which map $G_i(z)$ to $G_{i+1}(z)$, form the Schur algorithm for the matrix case. We remark that the algorithm in [17] contains a third step, which is irrelevant to our discussion, of producing a zero in $F_{i+1}(z)$ at some fixed point.

It is clear from the above presentation that the algorithm preserves the following two properties:

(i) $G_i(z)$ is analytic inside the unit disc.

(ii) $\|F_i\|_\infty = \|U_i^{-1}V_i\|_\infty \leqq 1$.

That is, if the above two properties are satisfied for the generator $G_i(z)$, then they are also satisfied for the generator $G_{i+1}(z)$.

Besides the analyticity of $G_i(z)$ in the unit disc and the property $\|F_i\|_\infty \leqq 1$, a third property is also preserved when $F_i(z)$ is analytic: $U_i(z)$ is nonsingular for every z inside the unit disc. This property is equivalent to $\nu_p(F_i) = 0$, where $\nu_p(\cdot)$ denotes the number of poles of the function inside the unit disc. A matrix $G_i(z)$ that has these three properties is said to be *admissible* in the language of Dewilde and Dym [17]. Thus, the Schur algorithm preserves the admissibility of $G_i(z)$ when $F_i(z)$ is analytic. We shall show below how to extend the third property to the case of meromorphic $F_i(z)$.

We now present a different formulation of the algorithm in [17] by introducing

$$\mu_i = \xi_i W_i,$$

which allows us to write (1a) in the form

$$\bar{G}_i(z) = G_i(z)W_i\left[I_{pq} + (B_i(z) - 1)\frac{J_{pq}\mu_i^*\mu_i}{\mu_i J_{pq}\mu_i^*}\right]. \tag{3}$$

Note that we still have complete freedom in choosing W_i. Let us now define

$$\varepsilon_i = \operatorname{sgn}(\mu_i J_{pq}\mu_i^*) = \operatorname{sgn}(\xi_i J_{pq}\xi_i^*) = \operatorname{sgn}[\eta_i G_i(z_i)J_{pq}G_i^*(z_i)\eta_i^*]. \tag{4}$$

It is convenient to consider separately the cases of $\varepsilon_i = \pm 1$ (the case $\varepsilon_i = 0$ leads to a singularity in the recursion and must be handled by different methods).

Case $\varepsilon_i = 1$. This case occurs, for example, if $F_i(z)$ is analytic and bounded by unity inside the unit disc (see [17]). It is well known that every two vectors with the same Euclidean norm can be (nonuniquely) related by an orthogonal rotation. Similarly, every two vectors with the same J norm can be related by a J-unitary matrix. Because ξ_i has positive J norm, there exists a J-unitary matrix W_i such that μ_i is collinear with $e_1 = [1, 0, \ldots]$, i.e.,

$$\xi_i W_i = (\xi_i J_{pq}\xi_i^*)^{1/2}e_1. \tag{5}$$

This particular choice of $\mu_i = \xi_i W_i$ reduces (1) to the simple form

$$\bar{G}_i(z) = G_i(z)W_i(B_i(z) \oplus I_{p+q-1}). \tag{6a}$$

Using this representation, it is easy to see that the first step in the algorithm consists of a multiplication of $G_i(z)$ from the right by a J_{pq} paraunitary matrix, which is a product of a constant part W_i, and a "dynamic part" $B_i(z) \oplus I_{p+q-1}$.

A similar transformation can also be applied to the second step of the algorithm. We choose a unitary matrix T_i (i.e., $T_i T_i^* = I$), such that

$$\eta_i T_i = \|\eta_i\| e_1.$$

Then, (2) can be written in the form

$$G_{i+1}(z) = T_i(B_i^{-1}(z) \oplus I_{p-1})T_i^*\bar{G}_i(z). \tag{6b}$$

We note that the choice to rotate in the direction of e_1 was for simplification purposes only and that one can choose other directions of rotation as well.

So far the matrix rotation W_i is subject to only one constraint (5). We present here one convenient way of choosing this matrix. First, we use the first entry of the row vector ξ_i as a pivot element and perform $p - 1$ elementary orthogonal rotations that annihilate the elements in positions 2 to p. Next, we use the last entry as a pivot element and annihilate the entries in positions from $p + 1$ to $p + q - 1$ using $q - 1$ elementary orthogonal rotations. The combined effect of these operations is to multiply $G_i(z)$ by a matrix of the form $P_i \oplus Q_i$, where P_i and Q_i are unitary. Because the matrix $P_i \oplus Q_i$ is

J_{pq} unitary, the resulting vector has the form $[x_i, 0, \ldots, 0 | 0, \ldots, 0, y_i]$ with $\xi_i J_{pq} \xi_i^* = |x_i|^2 - |y_i|^2 > 0$. As a result it follows that $|y_i/x_i| < 1$. Finally, we do a single elementary hyperbolic rotation that annihilates the last element of the vector. This last operation can be described as a multiplication by the matrix Θ_i, which is defined as

$$\Theta_i = \begin{pmatrix} L_i & -K_i \\ -K_i^* & M_i \end{pmatrix}, \tag{7a}$$

where

$$L_i = k_i^{-c} \oplus I_{p-1}, \quad M_i = I_{q-1} \oplus k_i^{-c}, \quad K_i = k_i k_i^{-c} e_1^* e_q \tag{7b}$$

and

$$k_i = y_i/x_i, \quad k_i^c = \sqrt{1 - |k_i|^2}, \quad k_i^{-c} = 1/k_i^c. \tag{7c}$$

We note that the condition $\varepsilon_i = 1$ enables us to do the last operation because it guarantees that $|k_i| < 1$. The three different operations can be represented as follows:

$$\xi_i \overset{P_i \oplus Q_i}{\longmapsto} [x_i, 0, \ldots, 0 | 0, \ldots, 0, y_i] \overset{\Theta_i}{\mapsto} [z_i, 0, \ldots, 0 | 0, \ldots, 0]. \tag{8}$$

In conclusion, our matrix W_i has the form

$$W_i = \begin{pmatrix} P_i & 0 \\ 0 & Q_i \end{pmatrix} \begin{pmatrix} L_i & -K_i \\ -K_i^* & M_i \end{pmatrix}. \tag{9}$$

This representation of the Schur algorithm makes it possible to extend the results proven in the scalar case [11]–[13] to matrix-valued functions.

LEMMA 1. *Let $F_i(z) = U_i^{-1}(z) V_i(z)$ be a ratio of two left coprime matrix polynomials and suppose that $\|F_i\|_\infty \leqq 1$ and $\varepsilon_i = 1$ (where ε_i is defined in (4)). Let $F_{i+1}(z) = U_{i+1}^{-1}(z) V_{i+1}(z)$, where $G_{i+1}(z) = [U_{i+1}(z) \quad V_{i+1}(z)]$ is defined by (6). Then*

(i) $\|F_{i+1}\|_\infty \leqq 1$.

(ii) $\nu_p(F_{i+1}) = \nu_p(F_i)$, *where $\nu_p(F)$ denotes the number of poles of $F(z)$ inside the unit disc. The poles here are defined using the Smith–McMillan form (see* [19]).

Proof. (i) Was already proved.

(ii) From (6)–(9) we obtain

$$U_{i+1}(z) = T_i [B_i^{-1}(z) \oplus I_{p-1}] T_i^* [U_i(z) P_i L_i - V_i(z) Q_i K_i^*] [B_i(z) \oplus I_{p-1}]. \tag{10}$$

Consequently,

$$\begin{aligned} \det U_{i+1} &= \det [U_i(z) P_i L_i - V_i(z) Q_i K_i^*] \\ &= \det [(U_i(z) - V_i(z) Q_i K_i^* L_i^{-1} P_i^*) P_i L_i] \\ &= k_i^{-c} \det [U_i(z) - V_i(z) Q_i K_i^* L_i^{-1} P_i^*] \end{aligned}$$

so that

$$\nu_z(U_{i+1}) = \nu_z(U_i - V_i Q_i K_i^* L_i^{-1} P_i^*), \tag{11}$$

where $\nu_z(\cdot)$ denotes the number of zeros of the function inside the unit disc. Now we observe that we can write

$$U_i(z) - V_i(z) Q_i K_i^* L_i^{-1} P_i^* = U_i(z) [I - U_i^{-1}(z) V_i(z) Q_i K_i^* L_i^{-1} P_i^*],$$

where

$$\|U_i^{-1} V_i Q_i K_i^* L_i^{-1} P_i^*\|_\infty \leqq \|U_i^{-1} V_i\|_\infty \|Q_i\|_\infty \|K_i^* L_i^{-1}\|_\infty \|P_i^*\|_\infty \leqq |k_i| < 1.$$

Thus, it follows from the matrix version of the Rouché theorem [15], [20] that

$$\nu_z(U_i - V_iQ_iK_i^*L_i^{-1}P_i^*) = \nu_z(U_i). \tag{12}$$

The last equation together with (11) imply

$$\nu_z(U_{i+1}) = \nu_z(U_i),$$

which is equivalent to (ii). Actually, because $K_i^*L_i^{-1} = k_ie_q^*e_1$ is a rank 1 matrix, we also observe that

$$\det[I - U_i^{-1}(z)V_i(z)Q_ik_ie_q^*e_1P_i^*] = 1 - e_1k_iP_i^*F_i(z)Q_ie_q^*.$$

Because $\|F_i\|_\infty \leqq 1$ and $|k_i| < 1$, it follows from the *scalar* Rouché theorem that

$$\nu_z[1 - e_1k_iP_i^*F_i(z)Q_ie_q^*] = \nu_p[1 - e_1k_iP_i^*F_i(z)Q_ie_q^*],$$

which also implies (12). □

For the analytic case, Lemma 1 provides a proof for the third property in the Dewilde and Dym definition of admissibility. Because an analytic function $F_i(z)$ satisfies $\nu_z(U_i) = 0$ and $\varepsilon_i = 1$, Lemma 1 implies that $\nu_z(U_{i+1}) = 0$; hence, $F_{i+1}(z)$ is also analytic.

Case $\varepsilon_i = -1$. Because in this case $\xi_i = \eta_iG_i(z_i)$ has *negative J* norm, we can rotate it so that μ_i will be collinear with $e_p = [0, \ldots, 0, 1]$. As a result, we find that (6a) is replaced by

$$\bar{G}_i(z) = G_i(z)W_i(I_{p+q-1} \oplus B_i(z)) \tag{13}$$

in complete analogy with the scalar case. Similarly, the form of W_i remains unchanged, but now k_i is selected to eliminate the first element of the vector ξ_i, i.e.,

$$\xi_i \overset{P_i \oplus Q_i}{\longmapsto} [x_i, 0, \ldots, 0 | 0, \ldots, 0, y_i] \overset{\Theta_i}{\mapsto} [0, \ldots, 0 | 0, \ldots, 0, z_i] \tag{14}$$

with $k_i = x_i^*/y_i^*$. We note that the second step in the recursion remains also unchanged and is given by (6b). Finally, when $\varepsilon_i = -1$, the number of poles of $F_i(z)$ decreases by one, as will be proved in the following lemma.

LEMMA 2. *Let* $F_i(z) = U_i^{-1}(z)V_i(z)$ *be a ratio of two left coprime matrix polynomials and suppose* $\|F_i\|_\infty \leqq 1$ *and* $\varepsilon_i = -1$ (*where* ε_i *is defined in* (4)). *Let* $F_{i+1}(z) = U_{i+1}^{-1}(z)V_{i+1}(z)$, *where* $G_{i+1}(z) = [U_{i+1}(z) \quad V_{i+1}(z)]$ *is defined by* (13) *and* (6b). *Then*

(i) $\|F_{i+1}\|_\infty \leqq 1$.

(ii) $\nu_p(F_{i+1}) = \nu_p(F_i) - 1$.

Proof. (i) The proof is the same as in Lemma 1.

(ii) In this case

$$U_{i+1}(z) = T_i[B_i^{-1}(z) \oplus I_{p-1}]T_i^*[U_i - V_i(z)Q_iK_i^*L_i^{-1}P_i^*]$$

(compare with (10)). Thus,

$$\nu_z(U_{i+1}) = \nu_z(U_i - V_iQ_iK_i^*L_i^{-1}P_i^*) - 1.$$

But the last equation and (12) from Lemma 1, which is also valid in this case, imply

$$\nu_z(U_{i+1}) = \nu_z(U_i) - 1,$$

which is equivalent to (ii). □

The following theorem summarizes the results of Lemmas 1 and 2.

THEOREM 1 (Schur algorithm for matrix-valued meromorphic functions). *Let* $F_i(z) = U_i^{-1}(z)V_i(z)$ *be a ratio of two left coprime matrix polynomials with* $\|F_i\|_\infty \leqq 1$

and let η_i *be any vector of dimension* $1 \times p$. *Define* $F_{i+1}(z) = U_{i+1}^{-1}(z)V_{i+1}(z)$ *by the following recursion.*

$$G_{i+1}(z) = \begin{cases} T_i(B_i^{-1}(z) \oplus I_{p-1})T_i^* G_i(z)W_i(B_i(z) \oplus I_{p+q-1}) & \text{if } \varepsilon_i = 1, \\ T_i(B_i^{-1}(z) \oplus I_{p-1})T_i^* G_i(z)W_i(I_{p+q-1} \oplus B_i(z)) & \text{if } \varepsilon_i = -1, \end{cases}$$

where ε_i is defined in (4), W_i is a J_{pq} unitary matrix defined in (9) and satisfies

$$\eta_i G_i(z_i)W_i = \begin{cases} [\eta_i G_i(z_i)J_{pq}G_i^*(z_i)\eta_i^*]^{1/2}[1, 0, \ldots, 0] & \text{if } \varepsilon_i = 1, \\ [-\eta_i G_i(z_i)J_{pq}G_i^*(z_i)\eta_i^*]^{1/2}[0, \ldots, 0, 1] & \text{if } \varepsilon_i = -1, \end{cases}$$

and T_i is a unitary matrix that satisfies

$$\eta_i T_i = \|\eta_i\|[1, 0, \ldots, 0].$$

Then,

(i) $\|F_{i+1}\|_\infty \leqq 1$.

(ii) $\nu_p(F_{i+1}) = \begin{cases} \nu_p(F_i) & \text{if } \varepsilon_i = 1, \\ \nu_p(F_i) - 1 & \text{if } \varepsilon_i = -1. \end{cases}$

COROLLARY. *The number of Smith–McMillan poles inside the unit disc is greater or equal to the number of negative signs in the sequence* $\{\varepsilon_i\}_{i=0}^{\infty}$.

Remarks. (1) Unlike the scalar case, equality does not always hold in the previous corollary. The reason for this is the poor choice of extraction directions η_i, which may not cover all possible directions of the zeros. Equality occurs when at some stage of the algorithm the generator becomes admissible, i.e., $U_i(z)$ does not have zeros inside the unit disc. Such a case happens, for example, when the directions of extraction are chosen to be the standard unit vectors in a cyclic order (see § 4). Another important case for which equality holds is in interpolation problems where the algorithm terminates after a finite number of steps with a constant or analytic (load) function. Yet, another case of equality is discussed in the next remark.

(2) When $U_i(z)$ has a zero inside the unit disc it is possible, by choosing an appropriate extraction point z_i and extraction direction η_i, to make $\varepsilon_i = -1$ and to extract this zero. Thus, by a proper choice of the first extraction points and extraction directions we can extract all the poles of $U_0(z)$ that are inside the unit disc to get an admissible generator.

(3) Although the proofs of Lemmas 1 and 2 use the special choice for the matrix W_i as in (9), Theorem 1 is valid for any matrix W_i that is J unitary. This follows from the fact that multiplication of the generator $G_i(z)$ by a constant J-unitary matrix does not change the number of zeros of $U_i(z)$ inside the unit disc.

(4) Limebeer and Green [21] obtained a similar result to Theorem 1 for the number of poles of the interpolating function arising in model reduction problems. Their derivation, however, is for functions that are meromorphic in a half plane and not in the unit disc and it uses a different method than ours.

(5) Theorem 1 is equivalent to a certain result of Alpay and Dym [22] on the dimension of reproducing kernel spaces associated with the Schur algorithm.

3. Interpolation problems. As an application of the Schur algorithm we describe how to solve the tangential Schur–Takagi problem [16], [18], which is defined as follows. Given a set of points z_i, $0 \leqq i \leqq n-1$ inside the unit disc and a set of vectors x_i, y_i,

$0 \leqq i \leqq n-1$ of dimension $1 \times p$ and $1 \times q$, respectively, find a matrix-valued function $F(z)$ of dimension $p \times q$ such that

(1) $\|F\|_\infty \leqq 1$.

(2) $x_i F(z_i) = y_i$, $0 \leqq i \leqq n-1$.

(3) $F(z)$ has k Smith–McMillan poles inside the unit disc ($k \geqq 0$).

Remarks. For simplicity of discussion we assume that the points z_i are distinct.

One solution of the problem that is described in [18, ch. 18] is as follows. We first find a matrix-valued function

$$M_n(z) = \begin{pmatrix} A_n(z) & B_n(z) \\ C_n(z) & D_n(z) \end{pmatrix}$$

that is J unitary on $|z| = 1$ and satisfies

$$[x_i\, y_i] M_n(z_i) = 0 \quad \text{for } i = 0, \ldots, n-1.$$

Then a parametrization of all the solutions to the tangential interpolation problem is given by

$$F(z) = (A_n(z)F_L(z) - B_n(z))(D_n(z) - C_n(z)F_L(z))^{-1}, \tag{15}$$

where $F_L(z)$ is such that $\|F_L\|_\infty \leqq 1$ and $(D_n(z) - C_n(z)F_L(z))^{-1}$ exists at the points z_i. To show that $F(z)$ satisfies the interpolation condition (2), we write

$$\begin{pmatrix} -F(z) \\ I \end{pmatrix} = M_n(z) \begin{pmatrix} -F_L(z) \\ I \end{pmatrix} (D_n(z) - C_n(z)F_L(z))^{-1},$$

and using the fact that

$$[x_i\, y_i] M_n(z_i) = 0,$$

we obtain

$$[x_i\, y_i] \begin{pmatrix} -F(z_i) \\ I \end{pmatrix} = 0,$$

which is condition (2).

Our results show that the matrix $M_n(z)$ can be found using the Schur algorithm in the following way. We define

$$M_n(z) = \Psi_0(z) \cdots \Psi_{n-1}(z),$$

where

$$\Psi_i = I_{p+q} + (B_i(z) - 1) J_{pq} \xi_i^* (\xi_i J_{pq} \xi_i^*)^{-1} \xi_i$$

(compare with (1)) and

$$\xi_i = [x_i\, y_i] \Psi_0(z_i) \cdots \Psi_{i-1}(z_i).$$

Then a simple calculation gives the desired condition

$$[x_i\, y_i] M_n(z_i) = \xi_i \Psi_i(z_i) \Psi_{i+1}(z_i) \cdots = 0.$$

We remark that the vectors $\xi_0, \ldots, \xi_{n-1}$ can be efficiently calculated in $O(n^2)$ operations.

From Theorem 1 we also obtain information on the number k of Smith–McMillan poles of $F(z)$ inside the unit disc:

$$k = \nu_p(F) = N_-(\varepsilon_0, \ldots, \varepsilon_{n-1}) + \nu_p(F_L),$$

where $N_-(\varepsilon_0, \ldots, \varepsilon_{n-1})$ is the number of negative elements in the sequence $\varepsilon_0, \ldots, \varepsilon_{n-1}$ (ε_i was defined in (4)). Note that a solution with the minimum number of poles ($k = N_-(\varepsilon_0, \ldots, \varepsilon_{n-1})$) is obtained by choosing a function $F_L(z)$ that is analytic.

An alternative way to obtain the number $N_-(\varepsilon_0, \ldots, \varepsilon_{n-1})$ is from the inertia of the Pick matrix

$$P = P_0 = \left(\frac{x_i x_j^* - y_i y_j^*}{1 - z_i z_j^*} \right)_{0 \leq i,j \leq n-1}. \tag{16}$$

One can easily check that P_0 is the solution of the following matrix equation

$$P_0 - F P_0 F^* = G_0 J G_0^*,$$

where $F = \operatorname{diag}(z_0, \ldots, z_{n-1})$ and

$$G_0 = \begin{pmatrix} x_0 & y_0 \\ \vdots & \vdots \\ x_{n-1} & Y_{n-1} \end{pmatrix}.$$

Similarly, we define the matrix P_i, $1 \leq i \leq n$ to be the solution of the equation

$$P_i - F P_i F^* = G_i J G_i^*$$

with

$$G_i = \begin{pmatrix} G_i^0 \\ \vdots \\ G_i^{n-1} \end{pmatrix},$$

where

$$G_i^j = [x_j y_j] \Psi_0(z_j) \Psi_1(z_j) \cdots \Psi_{i-1}(z_j).$$

We note that

$$(P_i)_{l,m} = \frac{1}{1 - z_l z_m^*} G_i^l J (G_i^m)^*,$$

and consequently

$$\begin{aligned}
(P_i)_{l,m} - (P_{i+1})_{l,m} &= \frac{1}{1 - z_l z_m^*} G_i^l J (G_i^m)^* - \frac{1}{1 - z_l z_m^*} G_{i+1}^l J (G_{i+1}^m)^* \\
&= \frac{1}{1 - z_l z_m^*} [G_i^l J (G_i^m)^* - G_i^l \Psi_i(z_l) J \Psi_i^*(z_m) (G_i^m)^*] \\
&= \frac{1}{1 - z_l z_m^*} G_i^l J [J - J \Psi_i(z_l) J \Psi_i^*(z_m) J] J (G_i^m)^* \\
&= \frac{1}{1 - z_l z_m^*} G_i^l J \left[\frac{(1 - |z_i|^2)(1 - z_l z_m^*)}{(1 - z_i^* z_l)(1 - z_i z_m^*)} \xi_i^* (\xi_i J \xi_i^*)^{-1} \xi_i \right] J (G_i^m)^* \\
&= \frac{1 - |z_i|^2}{\xi_i J \xi_i^*} \frac{1}{1 - z_i^* z_l} G_i^l J \xi_i^* \xi_i J (G_i^m)^* \frac{1}{1 - z_i z_m^*}.
\end{aligned}$$

Thus, we can write

$$P_i - P_{i+1} = \varepsilon_i u_i^* u_i,$$

where

$$u_i^* = \sqrt{\frac{1 - |z_i|^2}{|\xi_i J \xi_i^*|}} \begin{pmatrix} (1 - z_0 z_i^*)^{-1} G_i^0 J \xi_i^* \\ \vdots \\ (1 - z_{n-1} z_i^*)^{-1} G_i^{n-1} J \xi_i^* \end{pmatrix}.$$

Using the fact that $P_n = 0$, repeated application of the last equation gives the decomposition of the Pick matrix as

$$P = \varepsilon_0 u_0^* u_0 + \cdots \varepsilon_{n-1} u_{n-1}^* u_{n-1}.$$

Assuming that P is nonsingular, we obtain that $N_-(\varepsilon_0, \ldots, \varepsilon_{n-1})$ is equal to the number of negative eigenvalues of P.

Remark. The same technique of factorization of the Pick matrix was used in [23]. A similar factorization in the half-plane case was obtained in [21], [24].

4. Array formulation. For computational purposes it is more convenient to work with an array of coefficients rather than with functions. Here we describe the array formulation for the special case that all the extraction points are at the origin, i.e., $z_i = 0$ for all i. The array $\underline{G}_i = [\underline{U}_i \quad \underline{V}_i]$ is built of blocks of size $p \times (p + q)$, where each block is a coefficient of a (matrix) Taylor expansion

$$[I_p, zI_p, z^2 I_p \cdots] \underline{G}_i = \Phi_0(z) \cdots \Phi_{i-1}(z) G_i(z) = G_0(z) \Psi_0(z) \cdots \Psi_{i-1}(z).$$

Translating the modified Schur recursion in Theorem 1 to the array, we obtain

$$[I_p, zI_p, z^2 I_p \cdots] \underline{G}_{i+1} = [I_p, zI_p, z^2 I_p \cdots] \underline{G}_i W_i \begin{Bmatrix} z \oplus I_{p+q-1} \\ I_{p+q-1} \oplus z \end{Bmatrix}.$$

Consequently, one step of the algorithm is equivalent to a multiplication of the array from the right by the matrix W_i and then a downward shift of p places of the first or last column depending if $\varepsilon_i = 1$ or $\varepsilon_i = -1$, respectively.

Translating the property $\eta_i \bar{G}_i(0) = 0$, which we quoted as (1g), to the array, we can verify that the array $\underline{G}_i$ has i left null vectors

$$\{\eta_j \Phi_{j-1}(z^{-1}) \cdots \Phi_0(z^{-1}) [I_p, zI_p, z^2 I_p \cdots]\}|_{z=0} \underline{G}_i = 0 \quad \text{for } j = 0, \ldots, i-1.$$

Moreover, each additional step of the algorithm produces one more null vector. Thus, in the array domain the algorithm finds recursively an array that has a given set of left null vectors.

We now restrict our discussion to the case where each vector η_i is chosen to be one of the standard unit vectors $\{e_j\}_{j=1}^p$. In particular, we choose the vectors $\{\eta_i\}$ in a cyclic order: $\eta_0 = e_1, \ldots, \eta_{p-1} = e_p, \eta_p = e_1, \ldots$, and so on. For the choice $\eta_i = e_j$, where $j = i - 1 \bmod p$, the corresponding ξ_i, which is equal to the jth row of $G_i(0)$, is equal to the ith row of the array $\underline{G}_i$. Thus, for this case, the multiplication by W_i performs a special operation on the ith row of the array. Depending on whether $\varepsilon_i = 1$ or $\varepsilon_i = -1$, the multiplication by W_i annihilates all the elements in positions from 2 to $p + q$ or 1 to $p + q - 1$ on the ith row. Then, a downward shift of p places is done on the first or last column, respectively. As a result, the first i rows of the array $\underline{G}_i$ are equal to zero, and the corresponding i null vectors in this case are the first i standard unit vectors. Moreover, each cycle of p steps annihilates one complete block in the array and, therefore, corresponds to one step of the algorithm in block form [14].

5. Concluding remarks. In this article we present a new version of the classical Schur algorithm that is adapted to handle meromorphic matrix-valued functions. The new version leads to a new parametrization of matrix-valued meromorphic functions

that reflects the number of Smith–McMillan poles inside the unit disc in a very explicit way; it is also numerically efficient because it requires only scalar operations in contrast to the matrix and vector operations in previous versions. We also show how to solve tangential interpolation problems using this algorithm and we give a simple proof for the connection between the inertia of a certain Pick matrix and the number of poles of any interpolating function.

REFERENCES

[1] I. SCHUR, *Über potenzreihen die im innern des Einheitskreises beschränkt sind*, Journal für die Reine Angewandte Mathematik, 147 (1917), pp. 202–232. English translation in I. Schur, *Methods in operator theory and signal processing*, in Operator Theory: Advances and Applications, I. Gohberg, ed., Vol. 18, Birkhäuser-Verlag, Basel, 1986, pp. 31–88.

[2] P. DEWILDE AND H. DYM, *Schur recursion, error formulas, and convergence of rational estimators for stationary stochastic sequences*, IEEE Trans. Inform. Theory, IT-27 (1981), pp. 446–461.

[3] E. I. JURY, *Theory and Application of the z-Transform Method*, John Wiley & Sons, Inc., New York, 1964.

[4] P. P. VAIDYANATHAN AND S. K. MITRA, *A unified structural interpretation of some well-known stability test procedures for linear systems*, Proc. IEEE, 75 (1987), pp. 478–497.

[5] A. H. GRAY AND J. D. MARKEL, *Digital lattice and ladder filter synthesis*, Trans. Audio Electroacoust., AU-21 (1973), pp. 491–500.

[6] S. K. RAO AND T. KAILATH, *Orthogonal digital filters for* VLSI *implementation*, IEEE Trans. Circuits and Systems, CAS-31 (1984), pp. 933–945.

[7] T. KAILATH, *Signal processing applications of some moment problems*, in Moments in Mathematics, H. J. Landau, ed., pp. 71–109; Proc. Symposia in Applied Mathematics, 37, Amer. Math. Soc., Providence, Rhode Island, 1987.

[8] T. CONSTANTINESCU, A. H. SAYED, AND T. KAILATH, *A Schur-based approach to the four-block problem*, Proc. American Control Conf., Chicago, IL, June 1992.

[9] R. NEVANLINNA, *Über Beschränkte analytische funktionen*, Ann. Acad. Sci. Fenn., A32 (1929), pp. 1–75.

[10] P. DELSARTE, Y. GENIN, AND Y. KAMP, *On the role of the Nevanlinna-Pick problem in circuit and system theory*, Circuit Theory Appl., 9 (1981), pp. 177–187.

[11] C. CHAMFY, *Functions méromorphes dans le cercle-unité et leurs series de Taylor*, Ann. Inst. Fourier, 8 (1958), pp. 211–251.

[12] P. DELSARTE, Y. GENIN, AND Y. KAMP, *Pseudo-Carathéodory functions and Hermitian Toeplitz matrices*, Philips J. Res., 41 (1986), pp. 1–54.

[13] R. ACKNER, H. LEV-ARI, AND T. KAILATH, *Transmission-line models for the modified Schur algorithm*, IEEE Trans. Circuits and Systems, CAS-39, 4 (1992), pp. 290–296.

[14] P. DELSARTE, Y. GENIN, AND Y. KAMP, *Schur parametrization of positive definite block-Toeplitz systems*, SIAM J. Appl. Math., 36 (1979), pp. 47–61.

[15] Y. MONDEN AND S. ARIMOTO, *Generalized Rouché theorem and its application to multivariate autoregressions*, IEEE Trans. ASSP, ASSP-28 (1980), pp. 733–738.

[16] I. P. FEDČINA, *A description of the solution of the Nevanlinna-Pick tangent problem*, Akad. Nauk Armjan. SSR Dokl., 60 (1975), pp. 37–42.

[17] P. DEWILDE AND H. DYM, *Lossless chain scattering matrices and optimum linear prediction: the vector case*, Circuit Theory Appl., 9 (1981), pp. 135–175.

[18] J. A. BALL, I. GOHBERG, AND L. RODMAN, *Interpolation of Rational Matrix Functions*, Birkhäuser-Verlag, Basel, Boston, Berlin, 1990.

[19] T. KAILATH, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[20] I. C. GOHBERG AND E. I. SIGAL, *An operator generalization of the logarithmic residue theorem and the theorem of Rouché*, Math. USSR Sbornik, 13 (1971), pp. 603–625.

[21] D. J. N. LIMEBEER AND M. GREEN, *Parametric interpolation, H^∞-control and model reduction*, Int. J. Control,, 52 (1990), pp. 293–318.

[22] D. ALPAY AND H. DYM, *On applications of reproducing kernel spaces to Schur algorithm and rational J unitary factorization*, Oper. Theory: Adv. Appl., 18 (1986), pp. 89–159.

[23] R. ACKNER AND T. KAILATH, *On the Ptak-Young generalization of the Schur-Cohn theorem*, IEEE Trans. Automat. Control, 37 (1992), pp. 1601–1603.

[24] H. KIMURA, *Directional interpolation approach to H^∞ optimization and robust stabilization*, IEEE Trans. Automat. Control, 32 (1987), pp. 1085–1093.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 151–161, January 1994

012

REDUCIBILITY CONDITION OF A CLASS OF RATIONAL FUNCTION MATRICES*

KAI SHENG LU† AND JIA NING WEI‡

Abstract. The reducibility condition of a class of rational function matrices is derived. It is pointed out that the coefficient matrix of any resistor-inductor-capacitor (RLC) network is such a rational function matrix, which implies that the results obtained here can be applied to RLC networks and this paper has the effects on connecting matrix algebra and electrical network theory.

Key words. rational function field, rational function matrix, reducibility, network

AMS subject classifications. 15A15, 15A04, 93A10, 94C05

1. Introduction. Since the concepts of controllability and observability were first introduced by Kalman [1], much has been written on the subject [2]–[7] of linear systems theory. The linear systems over the field of real numbers were heavily studied. A linear system is said to be the one over the field of real numbers if each entry of each coefficient matrix of the system is a real number. However, uncontrollability (unobservability) is a "singular" condition in the sense that if the system $\dot{X} = AX + BU$, $Y = CX + DU$ is uncontrollable (unobservable), then almost any small perturbation of the elements of A and B (A and C) will cause it to become controllable (observable) [8], where the elements of A, B, and C are considered to be the independent parameters. Lin [9] first proposed the concept and condition of structural controllability to analyze these issues. Shields and Pearson [10], Glover and Silverman [11], Davison [12], Hosoe and Matsumoto [13], and Mayeda [14] extended them to multivariable linear systems. In [9]–[14] a matrix is said to be a structured matrix (SM) if each entry in the matrix is either fixed zero or free nonzero. Corfmat and Morse [15], Anderson and Hong [16], and Willems [17] permit some dependent relationships among nonfixed entries that are one-degree polynomials of independently variable parameters (simply, such a matrix is called a one-degree polynomial matrix) for physical reasons. Murota [18]–[20] first defined and studied the mixed matrices. A matrix A of the form $A = Q + T$ is called a mixed matrix if the nonzero entries of T are algebraically independent over the field to which the entries of Q belong. The irreducibility condition of mixed matrices was derived in [19]. Yamada and Luenberger [21] investigated the properties of the matrices called column-structured matrices (CSMs), which lie between SMs and the rational function matrices (RFMs).

Let F_ξ denote the field of all rational functions with real coefficients in q independently variable parameters $\xi = (\xi_1, \xi_2, \ldots, \xi_q) \in R^q$. R^q is said to be a parameter space. Let the matrix $M = M(\xi)$. M is called an RFM or a matrix in the field F_ξ if each entry in M is a member in F_ξ.

First, we establish some results on reducibility of RFMs of the form $A = (C + V)^{-1}U$ and $G = C + D$ with $C = \operatorname{diag}(\xi_1, \ldots, \xi_n)$, where $\xi_1, \ldots, \xi_n$ are n independently variable parameters and the $n \times n$ matrices D, V, and U do not contain $\xi_1, \ldots, \xi_n$. Obviously, A and G are not the matrices over the field of real numbers, SMs

* Received by the editor August 14, 1991; accepted for publication April 29, 1992.

† Section of Automatics, Department of Power Engineering, Wuhan University of Water Transportation Engineering, Wuhan, P. R. China.

‡ Section of Mathematics, Department of Basic Courses, Wuhan University of Water Transportation Engineering, Wuhan, P. R. China.

or CSMs. Also, A is not a one-degree polynomial-matrix or a mixed matrix. But when D is a constant matrix, G can be considered to be a matrix defined in [15]–[17]. Unfortunately, the reducibility problem was not studied in [15]–[17]. G is a mixed matrix and so the criterion of [19] is used to prove some important results of this paper.

Second, we present the following two properties.

Let $M(\xi)$ be an $n \times n$ RFM. $M(\xi)$ is said to be of property 2 if $\det(\lambda_0 I - M(\xi)) \neq 0$ over F_ξ, where λ_0 is an arbitrary nonzero constant; $M(\xi)$ is said to be of property 1 if its characteristic polynomial $\det(\lambda I - M(\xi))$ in $F_\xi[\lambda]$ has no nonzero multiple roots [22].

SMs and CSMs are of property 1, and it is not difficult to prove that they are of property 2. The two properties have an important application to the problem of structural controllability and observability [22]. The fact that the matrices A and G are also of properties 1 and 2 is pointed out here.

2. Lemmas and definitions. Consider a linear time-invariant structured system

$$\dot{X} = AX + Be, \tag{2.1}$$

where the $n \times n$ matrix $A = (C + V)^{-1}U$ has $n + m$ independent parameters $\xi = (\xi_1, \xi_2, \ldots, \xi_n; \xi_{n+1}, \ldots, \xi_{n+m}) \in R^{n+m}$, R^{n+m} is called the $(n + m)$-dimensional parameter space. F_ξ denotes the field of all rational functions of ξ. $C = \operatorname{diag}[\xi_1, \xi_2, \ldots, \xi_n]$. Each element in V and U is a rational function of only $\xi_{n+1}, \ldots, \xi_{n+m}$. So matrix A is an RFM. When $\xi_{n+1}, \ldots, \xi_{n+m}$ are fixed, V and U become two constant matrices.

Let R denote the real field and $R[x_1, x_2, \ldots, x_n]$ denote the ring of all real coefficient polynomials of n indeterminates $x_1, x_2, \ldots, x_n$. $R[x_1, \ldots, x_n]$ can be simply written as R_x or $R[X]$, $X = (x_1, x_2, \ldots, x_n)$. Let F_x denote the quotient field of R_x. The following lemma is a conclusion in algebraic theory [23].

LEMMA 1. *If a polynomial $f(\lambda)$ in ring $R_x[\lambda]$ can be decomposed in ring $F_x[\lambda]$, then $f(\lambda)$ can be also decomposed in ring $R_x[\lambda]$.*

When $f(\lambda) \in R_x[\lambda]$, the reducibility of $f(\lambda)$ in $R_x[\lambda]$ is equivalent to the reducibility for $f(\lambda)$ in $F_x[\lambda]$ by Lemma 1.

LEMMA 2. *If $f(\lambda) = a_0\lambda^n + a_1\lambda^{n-1} + \cdots + a_{n-1}\lambda + a_n$ is an n-degree polynomial in $R_x[\lambda]$ and $a_0 \neq 0$, $a_n \neq 0$, then $f(\lambda)$ is reducible in $R_x[\lambda]$ if and only if (iff) $g(\lambda) = a_0 + a_1\lambda + \cdots + a_{n-1}\lambda^{n-1} + a_n\lambda^n$ is reducible in $R_x[\lambda]$.*

Proof. This proof is obvious.

LEMMA 3. *Assume that $f(x_1, \ldots, x_n)$ is a polynomial in R_x and the highest degree term is $x_1 \cdots x_n$ (the degree of each of the other terms is less than n). Then $f(x_1, \ldots, x_n)$ is a reducible polynomial in R_x iff $f(x_1 - \lambda, \ldots, x_n - \lambda)$ is a reducible polynomial in $R_x[\lambda]$.*

Proof. This proof is obvious.

DEFINITION 1. If an $n \times n$ matrix exists

$$Q = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 0 & \cdots & 1 & \\ & & & \vdots & 1 & \vdots & \\ & & & & \ddots & & \\ & & & 1 & \cdots & 0 & \\ & & & & & & 1 \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{pmatrix}, \tag{2.2}$$

then Q is called a type 1 elementary matrix. An $n \times n$ matrix P is said to be a permutation matrix if P is a product of some type 1 elementary matrices. PAP^{-1} is said to be a permutation transformation of the matrix A. Clearly, $P^{-1} = P'$.

DEFINITION 2. Let M be an $n \times n$ matrix over F_ξ. M is then said to be reducible under TMT^{-1} or simply to be reducible if there exists some nonsingular matrix T over F_ξ such that

$$TMT^{-1} = \begin{pmatrix} M_1 & 0 \\ M_{21} & M_2 \end{pmatrix}, \tag{2.3}$$

where M_1 is an $n_1 \times n_1$ matrix, $1 \leqq n_1 < n$; otherwise, M is irreducible (under TMT^{-1}). M is said to be reducible under QMP if there exists some nonsingular matrix Q over R and some permutation matrix P such that

$$QMP = \begin{pmatrix} M_1 & 0 \\ M_{21} & M_2 \end{pmatrix}, \tag{2.4}$$

where M_1 is an $n_1 \times n_1$ matrix, $1 \leqq n_1 < n$; otherwise, M is irreducible under QMP (see p. 287, [19]). M is said to be reducible under PMP' if there exists some permutation matrix P such that

$$PMP' = \begin{pmatrix} M_1 & 0 \\ M_{21} & M_2 \end{pmatrix}, \tag{2.5}$$

where M_1 is an $n_1 \times n_1$ matrix, $1 \leqq n_1 < n$; otherwise, M is irreducible under PMP'.

3. Reducibility condition.

THEOREM 1. *Let $G = C + D$, where $C = \operatorname{diag}[\xi_1, \ldots, \xi_n]$, $\xi_1, \ldots, \xi_n$ are algebraically independent over R, and D is an $n \times n$ matrix over R. G is reducible under PGP' if and only if it is reducible under QGP', where P is a permutation matrix and Q is a nonsingular matrix over R.*

Proof. The necessity of the proof is obvious.

For sufficiency, assume that G is irreducible under PGP'. Let

$$A = PGP' = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \qquad J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix}, \tag{3.1}$$

where P is any permutation matrix, J is any nonsingular matrix over R, J_{11}, and A_{11} are two $n_1 \times n_1$ matrices, $1 \leqq n_1 < n$. $A_{12} \neq 0$ (since G is irreducible under PGP') and it is a matrix over R because of $\xi_1, \ldots, \xi_n$ on the diagonal of A. Then,

$$\tilde{J} \triangleq JA = \begin{pmatrix} \tilde{J}_{11} & \tilde{J}_{12} \\ \tilde{J}_{21} & \tilde{J}_{22} \end{pmatrix}, \tag{3.2}$$

where

$$\tilde{J}_{12} = J_{11}A_{12} + J_{12}A_{22}. \tag{3.3}$$

We now prove $\tilde{J}_{12} \neq 0$. Conversely, suppose $\tilde{J}_{12} = 0$. Since $A_{12} \neq 0$ is a matrix over R, $J_{11}A_{12}$ is also a matrix over R.

(i) If $J_{11}A_{12} \neq 0$, then $J_{12}A_{22} = -J_{11}A_{12} \neq 0$ is a matrix over R. If $J_{12} \neq 0$, $J_{12}A_{22}$ is a matrix over F_ξ not over R and so $\tilde{J}_{12} \neq 0$. If $J_{12} = 0$, then $\tilde{J}_{12} = J_{11}A_{12} \neq 0$.

(ii) If $J_{11}A_{12} = 0$, then $J_{12}A_{22} = 0$. Since A_{22} has full rank, $J_{12} = 0$, which yields that J_{11} and J_{22} are invertible. Thus, $A_{12} = 0$ by $J_{11}A_{12} = 0$, which is a contradiction.

Hence, $\tilde{J}_{12} \neq 0$ from (i) and (ii). Let $J = QP'$, where Q is any nonsingular matrix over R. Then $\tilde{J} = JA = QGP'$ is not a block-triangular matrix, which means that G is irreducible under QGP'. □

THEOREM 2. *Let $G = C + D$, where $C = \operatorname{diag}[\xi_1, \ldots, \xi_n]$, $\xi_1, \ldots, \xi_n$ are algebraically independent over R, and D is an $n \times n$ matrix over R. G is reducible (under TGT^{-1}) iff G is reducible under PGP'.*

Proof. It is only necessary to prove the necessity.

It can be known by Lemma 3 that the reducibility of $\det(\lambda I - G)$ and the reducibility of $\det G$ are equivalent. Thus, when G is reducible, $\det G$ is a reducible polynomial in $R[\xi]$. Obviously, G is a nonsingular mixed matrix with respect to the field R. By Theorem 6.2 in [19], G is reducible under QGP'. Then G is reducible under PGP' by Theorem 1. □

COROLLARY 1. *Let $K \subseteq F$ be fields, $G = C + D$, where $C = \operatorname{diag}[\xi_1, \ldots, \xi_n]$ is a matrix over F such that $\xi_1, \ldots, \xi_n$ are algebraically independent over the field K, and D is an $n \times n$ matrix over K. Then the following propositions are equivalent.*

(i) G is reducible under TGT^{-1}, where T is a nonsingular matrix over F.

(ii) G is reducible under QGP', where Q is a nonsingular matrix over K and P is a permutation matrix.

(iii) G is reducible under PGP', where P is a permutation matrix.

COROLLARY 2. *Let $G = C + D$, where $C = \operatorname{diag}[\xi_1, \ldots, \xi_n]$, $\xi_1, \ldots, \xi_n$ are algebraically independent over R, and D is a matrix over R. Then G is of properties 1 and 2.*

Proof. Clearly, for any complex constant λ_0, $\det(\lambda_0 I - G)$ is an n-degree polynomial in $\xi_1, \ldots, \xi_n$. It is impossible that the polynomial is identically zero, that is, any constant cannot be an eigenvalue of G over F_ξ. G is of property 2.

G is of property 1 by Theorem 26.1 of [18]. □

It is well known that the characteristic polynomial of an $n \times n$ matrix A can be written as

$$\det(\lambda I - A) = \lambda^n + a_1\lambda^{n-1} + \cdots + a_{n-1}\lambda + a_n, \tag{3.4}$$

where $a_k = (-1)^k D_k$; D_k is a sum of all the principal minors of order k in the matrix A, $1 \leqq k \leqq n$. If $a_r \neq 0$ $(1 \leqq r \leqq n)$, but $a_{r+1} = \cdots = a_n = 0$, r is a generic order of A. If the generic order of A is r, then

$$\det(\lambda I - A) = \lambda^{n-r}\phi(\lambda), \tag{3.5}$$

where $\phi(\lambda) = \lambda^r + a_1\lambda^{r-1} + \cdots + a_{r-1}\lambda + a_r$, $a_r \neq 0$. $\phi(\lambda)$ is called a nonzero part of $\det(\lambda I - A)$.

THEOREM 3. *If $A = (C + V)^{-1}U$, where $C = \operatorname{diag}[\xi_1, \xi_2, \ldots, \xi_n]$, V and U are two $n \times n$ constant matrices, $U \neq 0$, then the following propositions are equivalent.*

1. The nonzero part $\phi(\lambda)$ of $\det(\lambda I - A)$ is a reducible polynomial in $F_\xi[\lambda]$.

2(i). There exists some permutation matrix P such that

$$P(C + V - Ut)P' = \begin{pmatrix} G_1 & & & 0 \\ & G_2 & & \\ & & \ddots & \\ * & & & G_k \end{pmatrix}, \tag{3.6}$$

where G_i is an $n_i \times n_i$ irreducible matrix, $G_i = C_i + V_i - U_i t$, $C_i = \operatorname{diag}[\xi_{i_1}, \xi_{i_2}, \ldots, \xi_{in_i}]$, $i = 1, 2, \ldots, k$ $(k > 1)$, $n_1 + n_2 + \cdots + n_k = n$; and $\xi_{11}, \ldots, \xi_{1n_1}; \xi_{21}, \ldots, \xi_{2n_2}; \ldots; \xi_{k1}, \ldots, \xi_{kn_k}$ are a permutation of $\xi_1, \xi_2, \ldots, \xi_n$.

2(ii). In (3.6) there exist at least two submatrices $U_i \neq 0$ and $U_j \neq 0$, $i \neq j$, $1 \leqq i, j \leqq k$ such that $\det [\lambda I - (C_i + V_i)^{-1}U_i]$ and $\det [\lambda I - (C_j + V_j)^{-1}U_j]$ have, respectively, nonzero parts that are irreducible polynomials in $F_\xi[\lambda]$.

3. There exists some permutation matrix P such that

$$PAP' = \begin{pmatrix} A_{11} & 0 \\ * & A_{22} \end{pmatrix}, \tag{3.7}$$

where A_{11} is a $\mu \times \mu$ matrix, $1 \leqq \mu < n$, and A_{11} and A_{22} have, respectively, nonzero eigenvalues.

Proof. We use the cyclic method.

$1 \Rightarrow 2(i)$. Assume that the nonzero part of $\det (\lambda I - A)$ is reducible. Then

$$\begin{aligned} \det (\lambda I - A) &= \det [\lambda I - (C + V)^{-1}U] = \det (C + V)^{-1} \det [\lambda(C + V) - U] \\ &= \lambda^n \det (C + V)^{-1} \det [(C + V) - Ut], \end{aligned} \tag{3.8}$$

where $t = 1/\lambda$. Suppose that the generic order of A is r, $r \leqq n$. Then

$$\phi(\lambda) = \lambda^r \det (C + V)^{-1} \det [(C + V) - Ut]. \tag{3.9}$$

By Lemma 2, $\phi(\lambda)$ is reducible iff $\det (C + V)^{-1} \det [(C + V) - Ut]$ is a reducible polynomial in $F_\xi[t]$. Thus, $\varphi(t) \triangleq \det [(C + V) - Ut]$ is a reducible polynomial in $F_\xi[t] \cdot \varphi(t) = \varphi_1(t)\varphi_2(t) \ldots \varphi_s(t)$, where $s \geqq 2$, $\varphi_i(t)$ $(1 \leqq i \leqq s)$ is prime polynomial. Certainly, every $\varphi_i(t)$ is a nonzero-degree polynomial of $\xi_1, \ldots, \xi_n$; otherwise contradicting the fact that the coefficient of the highest degree power $\xi_1\xi_2 \cdots \xi_n$ in $\varphi(t)$ is one. Thus, $\det [C + V - Ut]$ is a reducible polynomial in $F(t)[\xi]$, where $F(t)$ denotes the field of all rational functions in t. Obviously, $C + (V - Ut)$ is a mixed matrix with respect to $F(t)$. Thus, (3.6) holds by Corollary 1.

$1 \Rightarrow 2(ii)$. By the relation

$$PAP' = \begin{pmatrix} (C_1 + V_1)^{-1}U_1 & & & 0 \\ & (C_2 + V_2)^{-1}U_2 & & \\ & & \ddots & \\ * & & & (C_k + V_k)^{-1}U_k \end{pmatrix}, \tag{3.10}$$

one can know that there exists i_0 $(1 \leqq i_0 \leqq k)$ such that $U_{i_0} \neq 0$. Suppose that the nonzero part of $\det [\lambda I - (C_{i_0} + V_{i_0})^{-1}U_{i_0}]$ is equal to $\phi(\lambda)$. Then $C_{i_0} + V_{i_0} - U_{i_0}t$ is reducible by 2(i), which is a contradiction.

$2 \Rightarrow 3$. This proposition is obvious from (3.10).

$3 \Rightarrow 1$. This proposition is obvious. □

COROLLARY 3. *If* $A = (C + V)^{-1}U$, *where* $C = \operatorname{diag} [\xi_1, \xi_2, \ldots, \xi_n]$, *$V$ and U are two $n \times n$ matrices over R, and U is invertible, then* $\det (\lambda I - A)$ *is reducible if and only if there exists some permutation matrix P such that* (3.7) *holds.*

Proof. It is only necessary to note that if U is invertible, the nonzero part of $\det (\lambda I - A)$ is, itself, right, and so $\det (\lambda I - A)$ has no zero eigenvalues. Thus, (3.7) holds by Theorem 3, and A_{11} and A_{22} have nonzero eigenvalues, respectively. □

PROPOSITION 1. *Let* $A = (C + V)^{-1}U$, *where* $C = \operatorname{diag} [\xi_1, \xi_2, \ldots, \xi_n]$; *$V$ and U are two $n \times n$ matrices over R. Then A is of property* 2.

Proof. Conversely, suppose that there exists a complex constant $\lambda_0 \neq 0$ such that $\det (\lambda_0 I - A) = 0$ for all the parameters $\xi_1, \xi_2, \ldots, \xi_n$. By (3.8) there exists $t_0 = 1/\lambda_0 \neq 0$ such that for all the parameters $\xi_1, \xi_2, \ldots, \xi_n$,

$$\det [(C + V) - Ut_0] = \frac{\perp}{\lambda_0^n} \det (C + V) \det (\lambda_0 I - A) = 0. \tag{3.11}$$

However, the left-hand side of (3.11) is an n-degree polynomial of $\xi_1, \xi_2, \ldots, \xi_n$, which contradicts the fact that (3.11) is an identity. □

COROLLARY 4. *If* $A = (C + V)^{-1}U$, *where* $C = \operatorname{diag}[\xi_1, \xi_2, \ldots, \xi_n]$, *$V$ and U are two $n \times n$ matrices over R, then A is of property* 1.

Proof. If, on the contrary, A has nonzero multiple eigenvalues, then the nonzero part of $\det(\lambda I - A)$ is reducible. Then (3.10) holds and $\det(\lambda I - A) = \prod_{i=1}^{k} \det[\lambda I - (C_i + V_i)^{-1}U_i]$. But $\det[\lambda I - (C_i + V_i)^{-1}U_i]$ has no nonzero multiple roots for every $i = 1, 2, \ldots, k$; otherwise, $C_i + V_i - U_i t$ is reducible still. On the other hand, each $\det[\lambda I - (C_i + V_i)^{-1}U_i]$ has no nonzero constant roots by Proposition 1. Moreover, since $(C_i + V_i)^{-1}U_i$ and $(C_j + V_j)^{-1}U_j$ have no parameters in common, where $i, j = 1, 2, \ldots, k$, $i \neq j$, they have no nonzero roots in common, which contradicts the assumption. Hence, this corollary is true. □

The assumption of Theorem 3 is that $\det[\lambda I - A]$ has the nonzero part $\phi(\lambda)$. Theorem 4 and Corollary 5 indicate the condition under which $\deg[\phi(\lambda)] \geqq 1$.

Let $M = (m_{ij})_{n\times n}$, $i, j = 1, \ldots, n$. By the definition of determinant,

$$\det M = \sum (-1)^{\tau(j_1 j_2 \cdots j_n)} m_{1 j_1} m_{2 j_2} \cdots m_{n j_n}, \tag{3.12}$$

where $\sum$ denotes the sum of all permutations. Equation (3.12) is called the expansion of the determinant $\det M$ and $m_{1 j_1} \cdots m_{n j_n}$ is called a term in the expansion.

LEMMA 4. *Let* $M = (m_{ij}) = C + D$, *where* $C = \operatorname{diag}[\xi_1, \ldots, \xi_n]$, *and D is an $n \times n$ matrix over R. If the expansion of* $\det M$ *has two similar terms* $m_{i_1 i_2} \cdots m_{i_{r-1} i_r} m_{i_r i_1} \psi$ *and* $m_{j_1 j_2} \cdots m_{j_{r-1} j_r} m_{j_r j_1} \psi$, *where* $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, n\}$, $n \geqq r \geqq 3$; $j_1, \ldots, j_r$ *is a permutation of* $i_1, \ldots, i_r$ *and* $(i_1, \ldots, i_r) \neq (j_1, \ldots, j_r)$; $m_{i_b i_{b+1}} \neq 0$, $m_{i_r i_1} \neq 0$, $m_{j_b j_{b+1}} \neq 0$, $m_{j_r j_1} \neq 0$, $b = 1, \ldots, r - 1$; $\psi = \xi_1 \cdots \xi_n / \xi_{i_1} \xi_{i_2} \cdots \xi_{i_r}$, *then there is some nonzero term that contains nondiagonal entries and whose degree is greater than* $\deg \psi$.

Proof. Without loss generality, let $i_1 = j_1, \ldots, i_s = j_s$, $1 \leqq s < r - 1$; $(i_{s+1}, \ldots, i_r) \neq (j_{s+1}, \ldots, j_r)$, where $j_{s+1}, \ldots, j_r$ is a permutation of $i_{s+1}, \ldots, i_r$. If $j_{s+1} \neq i_{s+1}$, then there must exist $j_x \in \{j_{s+2}, \ldots, j_r\}$ such that $j_x = i_{s+1}$, $s + 2 \leqq x \leqq r$. This implies that the expansion of $\det M$ has a nonzero term $m_{i_1 i_2} \cdots m_{i_s i_{s+1}} m_{j_x j_{x+1}} \cdots m_{j_{r-1} j_r} m_{j_r j_1} \xi_{j_{s+1}} \xi_{j_{s+2}} \cdots \xi_{j_{x-1}} \psi$, whose degree is greater than $\deg \psi$. □

THEOREM 4. *Let* $\phi(\lambda)$ *be the nonzero part of* $\det[\lambda I - (C + V)^{-1}U]$ *and* $G = C + V - Ut$ *be irreducible, where* $C = \operatorname{diag}[\xi_1, \ldots, \xi_n]$, *$V$ and U are two $n \times n$ matrices over R, and t is a parameter independently of* $\xi_1, \ldots, \xi_n$. *Then* $\deg[\phi(\lambda)] \geqq 1$ *iff* $U \neq 0$.

Proof. Necessity is obvious. Sufficiency will be proven. Since the coefficient of the term $\xi_1, \xi_2 \cdots \xi_n$ in $\varphi(t) = \det G$ is equal to one, $\varphi(t) = 0$ has no zero roots. It is thus known by (3.9) that $\deg[\varphi(t)] = \deg[\phi(\lambda)]$. It is only necessary to prove $\deg[\varphi(t)] \geqq 1$.

Let $G = (g_{ij})_{n\times n}$, $V = (v_{ij})_{n\times n}$, and $U = (u_{ij})_{n\times n}$. Then $g_{ii} = \xi_i + v_{ii} - u_{ii}t$, $i = 1, 2, \ldots, n$; $g_{ij} = v_{ij} - u_{ij}t$, $i \neq j$, $i, j = 1, 2, \ldots, n$.

Since $U \neq 0$, assume the entry $u_{ij} \neq 0$. If there exists some i, $1 \leqq i \leqq n$, such that $u_{ii} \neq 0$, then in the expansion of $\det G$ there is a term $u_{ii} t \xi_1 \cdots \xi_{i-1} \xi_{i+1} \cdots \xi_n \neq 0$ that is unique and may not vanish when all the nonzero terms are added. Then, assume $u_{ii} = 0$, $i = 1, 2, \ldots, n$. If the symmetric entry g_{ji} of g_{ij} is not zero, $\det G$ has a nonzero term $u_{ij} g_{ji} t \xi_1 \cdots \xi_n / \xi_i \xi_j$, which is unique, and so this lemma holds also for $u_{ii} = 0$, but $g_{ji} \neq 0$. Assume further that $g_{ji} = 0$. Moreover, since any nondiagonal entry can be placed in the position with the nth row and $(n - 1)$th column by performing a permutation transformation, without loss of generality one can assume that $u_{n-1,n} \neq 0$, but $g_{n,n-1} = 0$.

Now it will be proven that the cofactor of $g_{n-1,n}$, denoted by $\det G_{n-1,n}$, is not zero. We have

$$\det G_{n-1,n} = \det \begin{pmatrix} G_{n-2} & \zeta \\ \eta' & 0 \end{pmatrix}, \tag{3.13}$$

where G_{n-2} is an $(n-2) \times (n-2)$ principal submatrix with the first $n-2$ rows and the first $n-2$ columns of G, $\zeta = (g_{1,n-1}, g_{2,n-1}, \ldots, g_{n-2,n-1})'$, $\eta = (g_{n1}, g_{n2}, \ldots, g_{n,n-2})'$. Since $g_{n,n-1} = 0$, G is reducible under PGP' when $\zeta = 0$ or $\eta = 0$. By Theorem 2, G is reducible, which is a contradiction. Hence, $\zeta \neq 0$ and $\eta \neq 0$. In the case of $\zeta \neq 0$ and $\eta \neq 0$, if $\det G_{n-1,n} = 0$, which is an identity for all the parameters ξ, then for any parameter λ, we have

$$\det \begin{pmatrix} \lambda I - G_{n-2} & \zeta \\ \eta' & 0 \end{pmatrix} = 0,$$

which means that (G_{n-2}, ζ) is uncontrollable and/or (G'_{n-2}, η) is unobservable. By duality, it is only necessary to discuss that (G_{n-2}, ζ) is uncontrollable. Since $\zeta \neq 0$, it is impossible that all the modes of G_{n-2} are uncontrollable. Then there must exist some invertible matrix $\tilde{P}_1$ over F_ξ (see § 2.4.2, [24]) such that

$$\tilde{G}_{n-2} = \tilde{P}_1 G_{n-2} \tilde{P}_1^{-1} = \begin{pmatrix} \tilde{G}_1 & 0 \\ \tilde{G}_{21} & \tilde{G}_2 \end{pmatrix}, \qquad \tilde{P}_1 \zeta = \begin{pmatrix} 0 \\ \tilde{\zeta}_2 \end{pmatrix},$$

where $\tilde{G}_1$ is an $r \times r$ matrix, $\tilde{\zeta}_2$ is an $(n-2-r)$-dimensional nonzero vector, and $(\tilde{G}_2, \tilde{\zeta}_2)$ is controllable, $r < n-2$. There exists some permutation matrix P_1 by Theorem 2 such that

$$\bar{G}_{n-2} = P_1 G_{n-2} P_1' = \begin{pmatrix} \bar{G}_1 & 0 \\ \bar{G}_{21} & \bar{G}_2 \end{pmatrix}, \qquad P_1 \zeta = \begin{pmatrix} 0 \\ \zeta_2 \end{pmatrix}, \tag{3.14}$$

where $\bar{G}_1$ is an $r \times r$ matrix whose eigenvalues are the same with the ones of $\tilde{G}_1$, ζ_2 is an $(n-2-r)$-dimensional nonzero vector, and $(\bar{G}_2, \zeta_2)$ is controllable, $r < n-2$.

Since G_{n-2} is invertible, $\bar{G}_1$ and $\bar{G}_2$ are also invertible. We take

$$P = \begin{pmatrix} P_1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{3.15}$$

reduce (3.13), and obtain

$$\det \begin{pmatrix} \bar{G}_1 & 0 & 0 \\ \bar{G}_{21} & \bar{G}_2 & \zeta_2 \\ \eta_1' & \eta_2' & 0 \end{pmatrix} = 0. \tag{3.16}$$

Since $\det \bar{G}_1 \neq 0$,

$$\det \begin{pmatrix} \bar{G}_2 & \zeta_2 \\ \eta_2' & 0 \end{pmatrix} = 0. \tag{3.17}$$

Since $(\bar{G}_2, \zeta_2)$ is controllable, $\eta_2' = 0$. Then, let

$$\bar{P} = \begin{pmatrix} I_r & 0 & 0 \\ 0 & 0 & I_{n-1-r} \\ 0 & I_1 & 0 \end{pmatrix} \begin{pmatrix} P_1 & 0 \\ 0 & I_2 \end{pmatrix}, \tag{3.18}$$

where I_μ is a $\mu \times \mu$ unit matrix for each $\mu = 1, 2, r, n-1-r$. Thus,

$$\bar{P}G\bar{P}' = \begin{pmatrix} \bar{G}_1 & \text{x} & 0 & 0 \\ \eta_1' & g_{nn} & 0 & 0 \\ \bar{G}_{21} & \text{x} & \bar{G}_2 & \zeta_2 \\ \text{x} & g_{n-1,n} & \text{x} & g_{n-1,n-1} \end{pmatrix}, \tag{3.19}$$

which contradicts the fact that G is irreducible. Thus, $\det G_{n-1,n} \neq 0$. Thus, in the expansion of $\det G$ there exists at least one nonzero term $g_{1j_1}g_{2j_2}\cdots g_{nj_n}$ that is a nonzero degree term of t.

Now we shall prove that the sum of all nonzero terms containing t is not zero.

Let the nonzero term containing t with the highest degree of $\xi_1, \ldots, \xi_n$ be

$$g_{i_1i_2}g_{i_2i_3}\cdots g_{i_{r-1}i_r}g_{i_ri_1}\psi, \tag{3.20}$$

where $g_{i_1i_2} = v_{i_1i_2} - u_{i_1i_2}t$, $i_1 \neq i_2$, $u_{i_1i_2} \neq 0$, $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, n\}$, $3 \leqq r \leqq n$, $\psi = \xi_1\xi_2\cdots\xi_n/\xi_{i_1}\xi_{i_2}\cdots\xi_{i_r}$.

The term (3.20) is unique by Lemma 4, otherwise it is not the term containing t with the highest degree of $\xi_1, \ldots, \xi_n$. □

COROLLARY 5. *Let* $\phi(\lambda)$ *be the nonzero part of* $\det[\lambda I - (C+V)^{-1}U]$, $G = C + V - Ut$ *be reducible, and so*

$$PGP' = \begin{pmatrix} G_1 & & & 0 \\ & G_2 & & \\ & & \ddots & \\ * & & & G_k \end{pmatrix},$$

where $C = \operatorname{diag}[\xi_1, \ldots, \xi_n]$, *$V$ and U are two $n \times n$ matrices over R, t is a parameter independently of $\xi_1, \ldots, \xi_n$, P is a permutation matrix, $G_i = C_i + V_i - U_it$ is an $n_i \times n_i$ irreducible submatrix, $i = 1, \ldots, k$, $k \geqq 2$, $n_i \geqq 1$, $\sum_{i=1}^{k} n_i = n$.* $\operatorname{Deg}[\phi(\lambda)] \geqq 1$ *iff there is at least one submatrix $U_{i_0} \neq 0$, $i_0 \in \{1, 2, \ldots, k\}$.*

Remark 1. For Theorem 3, in (3.6), G_i is irreducible, $i = 1, \ldots, k$. Let the nonzero part $\phi(\lambda) = \phi_1(\lambda)\phi_2(\lambda)\cdots\phi_s(\lambda)$, where $2 \leqq s \leqq k$, $\deg[\phi_j(\lambda)] \geqq 1$, $\phi_j(\lambda)$ is a prime polynomial in $F_\xi[\lambda]$, $j = 1, \ldots, s$. Then there are only s submatrices $U_{i_1}, U_{i_2}, \ldots, U_{i_s}$ are not zero matrices, $\{i_1, i_2, \ldots, i_s\} \subseteq \{1, 2, \ldots, k\}$.

4. Applications to RLC networks. According to [25], the state equation of any RLC network can be written in the form (the notations in Table 2 [26] will be adopted here)

$$\dot{X} = AX + B_1e, \qquad Y = CX + D_1e + D_2\dot{e}, \tag{4.1}$$

where

$$A = A_1^{-1}A_2 \tag{4.2}$$

is an $n \times n$ matrix,

$$A_1 = \begin{pmatrix} C_t & 0 \\ 0 & L_l \end{pmatrix} + \begin{pmatrix} Q_{cc}C_lQ_{cc}' & 0 \\ 0 & Q_{LL}'L_tQ_{LL} \end{pmatrix},$$

$C_t = \operatorname{diag}[C_t(1), \ldots, C_t(n_c)]$, $C_t(1), \ldots, C_t(n_c)$ are n_c capacitor twigs; $L_l = \operatorname{diag}[L_l(1), \ldots, L_l(n_l)]$, $L_l(1), \ldots, L_l(n_l)$ are n_l inductor links; $n_c + n_l = n$; $C_l = \operatorname{diag}[C_l(1), \ldots, C_l(n_{cl})]$, $C_l(1), \ldots, C_l(n_{cl})$ are n_{cl} capacitor links; $L_t = \operatorname{diag}[L_t(1), \ldots, L_t(n_{Lt})]$, $L_t(1), \ldots, L_t(n_{Lt})$ are n_{Lt} inductor twigs; and Q_{cc} and Q_{LL} are both constant matrices. Assume that the network (4.1) has n_R resistors. By circuit theory, each entry of the matrix A_2 in (4.2) is a rational function of only n_R resistors. Thus, the network has, in number, $n + n_{cl} + n_{Lt} + n_R$ independently variable parameters. Obviously, the coefficient matrix

A of (4.1) is one of the matrices defined by (2.1). So the reducibility criteria in § 3 can be applied to RLC networks, and any RLC network expressed by eqn. (4.1) is of property 1 and property 2.

As an illustrative example consider an RLC network as shown in Fig. 1. The network has seven free parameters $\xi = (R_1, R_2, C_1, C_2, C_3, L, L_1) \in R^7$. The state equation is $\dot{X} = A_1^{-1}A_2X + \text{bu}$, where

$$X = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ i \\ i_1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} C_1 & & & & 0 \\ & C_2 & & & \\ & & C_3 & & \\ & & & L & \\ 0 & & & & L_1 \end{pmatrix} + [0]_{5\times 5},$$

$$A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & -R_2^{-1} & -R_2^{-1} & 0 & 0 \\ 0 & -R_2^{-1} & -R_2^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -R_1 \end{pmatrix}.$$

By direct calculation we see

$$\det(\lambda I - A_1^{-1}A_2) = \lambda^2\left(\lambda^2 + \frac{R_1}{L_1}\lambda + \frac{1}{L_1C_1}\right)\left(\lambda + \frac{C_2 + C_3}{R_2C_2C_3}\right),$$

whose nonzero part is a reducible polynomial in $F_\xi[\lambda]$. $A_1 - A_2t$ is reducible by Theorem 3. Indeed, when

$$\bar{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad \bar{P}(A_1 - A_2t)\bar{P}' = \begin{pmatrix} G_1 & & 0 \\ & G_2 & \\ 0 & & G_3 \end{pmatrix},$$

where

$$G_1 = \begin{pmatrix} C_1 & 0 \\ 0 & L_1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -1 & -R_1 \end{pmatrix}t \triangleq H_1 - U_1t,$$

$$G_2 = \begin{pmatrix} C_2 & 0 \\ 0 & C_3 \end{pmatrix} - \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix}\frac{t}{R_2} \triangleq H_2 - U_2t, \quad G_3 = L - 0t \triangleq H_3 - U_3t.$$

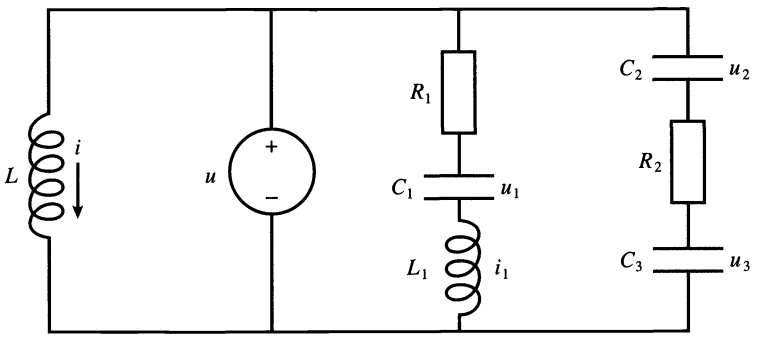


FIG. 1. *An* RLC *network.*

Clearly, G_i is irreducible under PG_iP', $i = 1, 2, 3$. Thus, G_i is irreducible under TG_iT^{-1} by Corollary 1, $i = 1, 2, 3$. Since $U_1 \neq 0$, $U_2 \neq 0$, but $U_3 = 0$, we know by Theorem 4 that

$$\det(\lambda I - H_1^{-1}U_1) = \lambda^2 + \frac{R_1}{L_1}\lambda + \frac{1}{L_1C_1} \quad \text{and} \quad \det(\lambda I - H_2^{-1}U_2) = \lambda\left(\lambda + \frac{C_2 + C_3}{R_2C_2C_3}\right)$$

have the irreducible nonzero parts, respectively, but $\det(\lambda I - H_3^{-1}U_3) = \lambda$ has no nonzero part.

5. Summary. As far as we know, the coefficient matrices of almost all linear physical systems can be considered to be RFMs. However, little is known about the properties of RFMs. To explore the properties of physical systems, particularly the effects of physical structures, it becomes necessary to investigate the properties of RFMs mathematically.

The reducibility criteria for a class of RFMs have been obtained here. What are the reducibility conditions of the other RFMs? This question is complex and interesting. The reducibility criteria will have an application to the problem of controllability and observability of RLC networks over the field F_ξ. These problems are left for further research.

Acknowledgments. We are grateful to the reviewers for their thoughtful comments that substantially improved this paper and to Prof. K. Murota of the University of Tokyo for supplying relevant literature.

REFERENCES

[1] R. E. KALMAN, *On the general theory of control systems*, in Proc. 1st Internat. Congress on Automatic Control, Butterworth's, London, 1960, pp. 481–493.
[2] ———, *Mathematical description of linear dynamical systems*, SIAM J. Control, 1 (1963), pp. 152–192.
[3] E. GILBERT, *Controllability and observability in multivariable control systems*, SIAM J. Control, 1 (1963), pp. 128–151.
[4] V. M. POPOV, *On a new problem of stability for control systems*, Automat. Remote Control, 24 (1963), pp. 1–23.
[5] C. T. CHEN, *Representations of linear time-invariant composite systems*, IEEE Trans., AC-13 (1966), pp. 277–283.
[6] V. BELEVITCH, *Classical Network Theory*, Holden-Day, San Francisco, 1968.
[7] M. L. J. HAUTUS, *Controllability and observability conditions of linear autonomous systems*, Nederl. Akad. Wetensch. Proc. Ser. A, 72 (1969), pp. 443–448.
[8] E. B. LEE AND L. MARKUS, *Foundations of Optimal Control Theory*, John Wiley & Sons, New York, 1967.
[9] C. T. LIN, *Structural controllability*, IEEE Trans., AC-19 (1974), pp. 201–208.
[10] R. W. SHIELDS AND J. B. PEARSON, *Structural controllability of multi-input linear systems*, IEEE Trans., AC-21 (1976), pp. 203–212.
[11] K. GLOVER AND L. M. SILVERMAN, *Characterization of structural controllability*, IEEE Trans., AC-21 (1976), pp. 534–537.
[12] E. J. DAVISON, *Connectability and structural controllability of composite systems*, Automatica, 13 (1977), pp. 109–123.
[13] S. HOSOE AND K. MATSUMOTO, *On the irreducibility condition in the structural controllability theorem*, IEEE Trans., AC-24 (1979), pp. 963–966.
[14] H. MAYEDA, *On structural controllability theorem*, IEEE Trans., AC-26 (1981), pp. 795–798.
[15] J. P. CORFMAT AND A. S. MORSE, *Structurally controllable and structurally canonical systems*, IEEE Trans., AC-21 (1976), pp. 129–131.
[16] B. D. O. ANDERSON AND H. M. HONG, *Structural controllability and matrix nets*, Internat. J. Control, 35 (1982), pp. 397–416.
[17] J. L. WILLEMS, *Structural controllability and observability*, Systems Control Lett., 8 (1986), pp. 5–12.

[18] K. MUROTA, *Systems Analysis by Graphs and Matroids—Structural Solvability and Controllability*, Springer-Verlag, New York, 1987.
[19] ———, *On the irreducibility of layered mixed matrices*, Linear Multi-linear Algebra, 24 (1989), pp. 273–288.
[20] ———, *On the Smith normal form of structured polynomial matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 747–765.
[21] T. YAMADA AND D. G. LUENBERGER, *Generic properties of column-structured matrices*, Linear Algebra Appl., 65 (1985), pp. 186–206.
[22] K. S. LU AND J. N. WEI, *Rational function matrices and structural controllability and observability*, IEE Proc.-D, 138 (1991), pp. 388–394.
[23] N. JACOBSON, *Lectures in Abstract Algebra. Vol.* II. *Linear algebra*, D. Van Nostrand, New York, 1953.
[24] T. KAILATH, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[25] E. S. KUH AND R. A. ROHRER, *State variable approach to network analysis*, Proc. IEEE, 53 (1965), pp. 672–686.
[26] N. BALABANIAN AND T. A. BICKART, *Electrical Network Theory*, John Wiley & Sons, New York, 1969.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 162–174, January 1994

013

FAST PLANE ROTATIONS WITH DYNAMIC SCALING*

ANDREW A. ANDA†‡ AND HAESUN PARK†§

Abstract. This paper presents fast plane rotations for orthogonal similarity and orthogonal one-sided transformations. Fast rotations have the advantage that they reduce the number of square roots and multiplications. The authors' new rotations have further advantages over the existing fast rotations: they obviate the rescaling that has been necessary to guard against underflow or overflow and they give higher efficiency, especially on vector processors. An error analysis, in the case of the *QR* decomposition, and computational results that illustrate the effects of the dynamic scaling are presented.

Key words. fast plane rotations, orthogonal transformations, scaling

AMS subject classifications. 65F25, 15A23, 15A18

1. Introduction.

A plane rotation $J(\theta, p, q)$ of order n through an angle θ in a (p, q) plane is the same as the identity matrix I_n, except for the four elements at the intersections of the pth and qth rows and columns. The general form of an $n \times n$ rotation matrix $J(\theta, p, q)$ in the (p, q) plane, which we will denote as J, is

$$J = J(\theta, p, q) = \begin{bmatrix} I_{p-1} & 0 & 0 & 0 & 0 \\ 0 & c & 0 & -s & 0 \\ 0 & 0 & I_{q-p-1} & 0 & 0 \\ 0 & s & 0 & c & 0 \\ 0 & 0 & 0 & 0 & I_{n-q} \end{bmatrix}, \tag{1}$$

where $c \equiv \cos\theta$, $s \equiv \sin\theta$. Plane rotations are used in various algorithmic contexts, e.g., the *QR* decomposition, the Jacobi and *QR* algorithms for eigendecompositions, the Hestenes algorithm for the singular value decomposition, and reduction to Hessenberg form. Although these algorithms differ in the purpose and range of the angles, we can classify them largely into two cases: those that apply similarity transformations

$$\tilde{X} = J(\theta, p, q)^T X J(\theta, p, q) \tag{2}$$

and others that apply one-sided transformations

$$\tilde{X} = XJ(\theta, p, q) \quad \text{or} \quad \tilde{X} = J^T(\theta, p, q)X. \tag{3}$$

We use the notations X_{pq} and J_{pq} to denote the 2×2 submatrices of X and $J(\theta, p, q)$ in the (p, q) plane, respectively, i.e.,

$$X_{pq} = \begin{bmatrix} x_{pp} & x_{pq} \\ x_{qp} & x_{qq} \end{bmatrix} \quad \text{and} \quad J_{pq} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}.$$

Fast rotations were developed with the motivations to reduce the number of scalar-vector multiplications and eliminate square roots from the calculation of the plane rotations. Gentleman [5] first formulated a method for a fast Givens rotation. Hammarling [9] later modified that formulation into several computational schemes and pointed out

* Received by the editors December 10, 1990; accepted for publication (in revised form) May 12, 1992.

† Computer Science Department, University of Minnesota, Minneapolis, Minnesota 55455 (anda@cs.umn.edu, hpark@cs.umn.edu).

‡ The work of this author was supported in part by the University of Minnesota Army High Performance Computing Research Center grant DAAL03-89-C-0038.

§ The work of this author was supported in part by National Science Foundation grant CCR-8813493 and University of Minnesota Army High Performance Computing Research Center grant DAAL03-89-C-0038.

that the method can be applied to similarity transformations. There are other types of fast rotations that are developed for the purpose of reducing the number of multiplications in a product of plane rotations [1] or to reduce the device area on a systolic array [8]. However, a literature survey of research and production algorithms that utilize plane rotations has shown a pronounced avoidance of the fast plane rotation. This is partly a consequence of the fact that careful monitoring for the prevention of underflow and overflow is not conveniently achieved wherever the standard fast plane rotation algorithm is utilized [2], [7]. Hammarling [9] suggested that underflow can be avoided either by storing the exponent separately, by normalizing occasionally, or by performing row (or column) interchanges. Separate storage of the exponent is clearly not efficient for currently popular high level languages. Row (or column) interchanges introduce an overhead. Moreover, occasional normalization can be problematic to implement because a nontrivial amount of monitoring is necessary to successfully implement fast Givens transformations.

We present a set of new fast rotations that obviate the monitoring and periodic rescaling necessitated by the standard fast plane rotations. Our fast rotations dynamically scale the diagonal factor matrix to be close to an identity matrix. Variations on the standard fast rotation matrix are developed and algorithms that implement them are offered. Issues of accuracy and efficiency are also discussed. Computational results on the Cray-2 illustrating the effects of dynamic scaling are presented.

2. Fast rotations. Suppose one step of a one-sided transformation via a rotation $J(\theta, p, q)$ gives

$$\tilde{X} = XJ(\theta, p, q) \tag{4}$$

as in the one-sided Jacobi [4], [11] and the Hestenes [10] algorithms and in the QR decomposition (transposed). The essential idea of a fast rotation is that the number of multiplications is reduced by keeping the matrix X in the factored form YD, where D is a diagonal matrix and Y is accordingly scaled, and these two factors are updated separately. The calculation of the product of the two factors may be postponed until the explicit result is required. The diagonal matrix D is initialized as the identity matrix. The secondary advantage of the fast rotation is that the square roots for the computation of cosine and sine can be eliminated [13]. If $X = YD$, then the rotation from the right by J can be represented as

$$\tilde{X} = XJ = YDJ = YF(p, q)\tilde{D} = \tilde{Y}\tilde{D}, \tag{5}$$

where $\tilde{D}$ is a diagonal matrix and $F(p, q)$ is defined as

$$F(p, q) \equiv \begin{bmatrix} I_{p-1} & 0 & 0 & 0 & 0 \\ 0 & f_{pp} & 0 & f_{pq} & 0 \\ 0 & 0 & I_{q-p-1} & 0 & 0 \\ 0 & f_{qp} & 0 & f_{qq} & 0 \\ 0 & 0 & 0 & 0 & I_{n-q} \end{bmatrix}. \tag{6}$$

We will call $F(p, q)$ a *fast rotation* if the choices of f_{pp}, f_{pq}, f_{qp}, and f_{qq} result in halving the number of multiplications in applying the rotation J. If a typical step is represented as

$$X^{(k+1)} = X^{(k)}J^{(k)}, \tag{7}$$

then

$$X^{(k+1)} = Y^{(k+1)}D^{(k+1)} = Y^{(1)}F^{(1)}\cdots F^{(k)}D^{(k+1)} = X^{(1)}J^{(1)}\cdots J^{(k)}, \tag{8}$$

and we have

$$J^{(i)} = D^{(i)^{-1}} F^{(i)} D^{(i+1)}. \tag{9}$$

In the case of the two-sided transformations, if $X = DYD$, then

$$\tilde{X} = J^T X J = J^T DYDJ = \tilde{D} F^T Y F \tilde{D} = \tilde{D} \tilde{Y} \tilde{D}. \tag{10}$$

If a typical step is represented as $X^{(k+1)} = J^{(k)^T} X^{(k)} J^{(k)}$, then

$$\begin{aligned} X^{(k+1)} &= D^{(k+1)} Y^{(k+1)} D^{(k+1)} \\ &= D^{(k+1)} F^{(k)^T} \cdots F^{(1)^T} Y^{(1)} F^{(1)} \cdots F^{(k)} D^{(k+1)} \\ &= J^{(k)^T} \cdots J^{(1)^T} X^{(1)} J^{(1)} \cdots J^{(k)} \end{aligned}$$

and, again, we have

$$J^{(i)} = D^{(i)^{-1}} F^{(i)} D^{(i+1)}. \tag{11}$$

We discuss only column oriented one-sided transformations because we have implemented the new algorithms in Fortran in which the column oriented one-sided transformations give higher efficiency and our work has been motivated by one-sided Jacobi algorithms [11]. The results for row-oriented and the two-sided transformations follow easily.

2.1. Standard fast rotations. Suppose a fast rotation transforms $X = YD$ into $\tilde{X} = \tilde{Y}\tilde{D}$. The transformation can be shown as follows:

$$\begin{aligned} [\tilde{x}_p, \tilde{x}_q] &= [x_p, x_q] \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \\ &= [y_p, y_q] \begin{bmatrix} d_p & 0 \\ 0 & d_q \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \\ &= [y_p, y_q] \begin{bmatrix} f_{pp} & f_{pq} \\ f_{qp} & f_{qq} \end{bmatrix} \begin{bmatrix} \tilde{d}_p & 0 \\ 0 & \tilde{d}_q \end{bmatrix} \\ &= [\tilde{y}_p, \tilde{y}_q] \begin{bmatrix} \tilde{d}_p & 0 \\ 0 & \tilde{d}_q \end{bmatrix}, \end{aligned}$$

where x_i denotes the ith column of X.

There are several ways to choose

$$F_{pq} = \begin{bmatrix} f_{pp} & f_{pq} \\ f_{qp} & f_{qq} \end{bmatrix}$$

and $\tilde{D}$ so that the number of multiplications is reduced by half compared to the standard rotation. The two most commonly used F_{pq} are either of the type $\left[\begin{smallmatrix} 1 & \alpha \\ \beta & 1 \end{smallmatrix}\right]$ or $\left[\begin{smallmatrix} \beta & 1 \\ 1 & \alpha \end{smallmatrix}\right]$. With the first type, we have

$$[\tilde{x}_p, \tilde{x}_q] = [y_p, y_q] \begin{bmatrix} 1 & -\alpha \\ \beta & 1 \end{bmatrix} \begin{bmatrix} cd_p & 0 \\ 0 & cd_q \end{bmatrix};$$

thus,

$$\left\{\begin{array}{c} \tilde{d}_p \Leftarrow cd_p \\ \tilde{d}_q \Leftarrow cd_q \\ \tilde{y}_p \Leftarrow y_p + \beta y_q \\ \tilde{y}_q \Leftarrow y_q - \alpha y_p \end{array}\right\}, \tag{12}$$

where $\alpha = t(d_p/d_q)$, $\beta = t(d_q/d_p)$, and $t = \tan\theta$.

When $|\theta| > (\frac{\pi}{4})$, especially when $|c| \ll 1$, successive multiplications by small factors in D can quickly lead to underflow. To bound the maximum decrease in the diagonal factor matrix D, one must use an alternative formulation of the fast rotation that updates the diagonal elements of D with sines rather than cosines:

$$[\tilde{x}_p, \tilde{x}_q] = [y_p, y_q]\begin{bmatrix} \beta & -1 \\ 1 & \alpha \end{bmatrix}\begin{bmatrix} sd_q & 0 \\ 0 & sd_p \end{bmatrix};$$

thus,

$$\left\{\begin{array}{c} \tilde{d}_p \Leftarrow sd_q \\ \tilde{d}_q \Leftarrow sd_p \\ \tilde{y}_p \Leftarrow y_q + \beta y_p \\ \tilde{y}_q \Leftarrow -y_p + \alpha y_q \end{array}\right\}, \tag{13}$$

where $\alpha = (1/t)(d_q/d_p)$ and $\beta = (1/t)(d_p/d_q)$.

Although, for any rotation, the decrease in magnitude of each element of the diagonal factor D can be bounded by $1/\sqrt{2}$, with the use of the above two formulations, the diagonal elements of D are reduced each time and may eventually cause underflow.

2.2. Modified fast rotations. In [3], de Rijk further developed the fast rotation for the Hestenes algorithm for computing the singular value decomposition on vector processors to eliminate a temporary copy of one of the columns. The idea is that the expression $\tilde{y}_p \Leftarrow y_p + \beta y_q$ in (12) can be rearranged as $y_p = \tilde{y}_p - \beta y_q$; thus, substituting it into the other triad, we get

$$\begin{aligned} \tilde{y}_q &= y_q - \alpha y_p \\ &= y_q - \alpha(\tilde{y}_p - \beta y_q) \\ &= \left(y_q - \left(\frac{\alpha}{1+t^2}\right)\tilde{y}_p\right)(1+t^2) \\ &= (y_q - (\alpha c^2)\tilde{y}_p)(c^{-2}). \end{aligned}$$

Letting the diagonal element d_q of D take care of the value c^{-2}, we get $\tilde{d}_q \Leftarrow \tilde{d}_q/c^2 = d_q/c$ and

$$[\tilde{x}_p, \tilde{x}_q] = [y_p, y_q]\begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix}\begin{bmatrix} 1 & -\alpha \\ 0 & 1 \end{bmatrix}\begin{bmatrix} cd_p & 0 \\ 0 & c^{-1}d_q \end{bmatrix}$$

$$\left\{\begin{array}{c} \tilde{d}_p \Leftarrow cd_p \\ \tilde{d}_q \Leftarrow c^{-1}d_q \\ \tilde{y}_p \Leftarrow y_p + \beta y_q \\ \tilde{y}_q \Leftarrow y_q - \alpha \tilde{y}_p \end{array}\right\}, \tag{14}$$

where $\alpha \Leftarrow \alpha c^2 = cs(d_p/d_q)$ and $\beta = t(d_q/d_p)$. In the Hestenes algorithm for computing the singular value decomposition, the angle can be always chosen in $[-\frac{\pi}{4}, \frac{\pi}{4}]$, thus $1/\sqrt{2} \leqq |c|$.

The same procedures that yield expression (14) also yield a modification of expression (13) for angles $\theta \in [\frac{\pi}{4}, \frac{\pi}{2}]$ in magnitude to scale the diagonal factor:

$$[\tilde{x}_p, \tilde{x}_q] = [y_p, y_q]\begin{bmatrix}\beta & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & \alpha\\ 0 & -1\end{bmatrix}\begin{bmatrix}sd_q & 0\\ 0 & s^{-1}d_p\end{bmatrix}$$

$$\left\{\begin{array}{c}\tilde{d}_p \Leftarrow sd_q\\ \tilde{d}_q \Leftarrow s^{-1}d_p\\ \tilde{y}_p \Leftarrow y_q + \beta y_p\\ \tilde{y}_q \Leftarrow -y_p + \alpha\tilde{y}_p\end{array}\right\}, \tag{15}$$

where $\alpha = cs(d_q/d_p)$ and $\beta = (1/t)(d_p/d_q)$.

For the fast rotations, presented in (12) and (13), the order in which the columns are updated is inconsequential providing that the correct vector has been copied for reuse. However, the de Rijk modified fast rotation presented in (14) requires that the pth column be updated before the updating of the qth column. A variant of the de Rijk rotation follows naturally from interchanging the order of the column updates, for angles $|\theta| \leqq \frac{\pi}{4}$:

$$[\tilde{x}_p, \tilde{x}_q] = [y_p, y_q]\begin{bmatrix}1 & -\alpha\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \beta & 1\end{bmatrix}\begin{bmatrix}c^{-1}d_p & 0\\ 0 & cd_q\end{bmatrix}$$

$$\left\{\begin{array}{c}\tilde{d}_q \Leftarrow cd_q\\ \tilde{d}_p \Leftarrow c^{-1}d_p\\ \tilde{y}_q \Leftarrow y_q - \alpha y_p\\ \tilde{y}_p \Leftarrow y_p + \beta\tilde{y}_q\end{array}\right\}, \tag{16}$$

where $\alpha = t(d_p/d_q)$ and $\beta = cs(d_q/d_p)$, and, for angles $\theta \in [\frac{\pi}{4}, \frac{\pi}{2}]$ in magnitude,

$$[\tilde{x}_p, \tilde{x}_q] = [y_p, y_q]\begin{bmatrix}0 & -1\\ 1 & \alpha\end{bmatrix}\begin{bmatrix}1 & 0\\ -\beta & 1\end{bmatrix}\begin{bmatrix}s^{-1}d_q & 0\\ 0 & sd_p\end{bmatrix}$$

$$\left\{\begin{array}{c}\tilde{d}_q \Leftarrow sd_p\\ \tilde{d}_p \Leftarrow s^{-1}d_q\\ \tilde{y}_q \Leftarrow -y_p + \alpha y_q\\ \tilde{y}_p \Leftarrow y_q - \beta\tilde{y}_q\end{array}\right\}, \tag{17}$$

where $\alpha = (1/t)(d_q/d_p)$ and $\beta = cs(d_p/d_q)$.

We call the new fast rotations presented in (14)–(17) *chained* fast rotations. The following simplified representations of three different methods for applying a plane rotation, $\left[\begin{smallmatrix}c & -s\\ s & c\end{smallmatrix}\right]$, to update the two vectors, w and $v \in \mathscr{R}^n$ illustrate that the new fast rotations can be chained.

Standard rotation:

$$[\tilde{v}, \tilde{w}] = [v, w]\begin{bmatrix}c & -s\\ s & c\end{bmatrix} = [cv + sw, -sv + cw]. \tag{18}$$

Standard fast rotation:

$$[\tilde{v}, \tilde{w}] = [v, w]\begin{bmatrix}1 & \alpha\\ \beta & 1\end{bmatrix} = [v + \beta w, w + \alpha v]. \tag{19}$$

Chained fast rotation:

$$[\tilde{v}, \tilde{w}] = [v, w]\begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix}\begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} = [v + \beta w,\ w + \alpha(v + \beta w)]. \tag{20}$$

The computational efficiency of the chained fast plane rotation over the standard fast plane rotation results from the elimination of the temporary vector store and read that are necessary for the standard fast plane rotation.

3. Self-scaling algorithms. Using the preceding expressions for fast plane rotations, we now construct algorithms that dynamically scale the elements of the diagonal matrix to be close to one. In the standard fast plane rotation, presented in (12) and (13), both diagonal elements are diminished at each rotation. In the chained fast rotations, presented in (14)–(17), one diagonal element is augmented by the same factor as the other diagonal element is diminished. Expressions (14) and (15) diminish the diagonal element having the smaller index and augment the element having the larger index. This way, the smaller the index is, the more often the corresponding diagonal element will be diminished. Likewise, the larger the index is, the more often the corresponding diagonal element will be augmented. The resulting distribution, after one or more cycles, tends to have the largest diagonal entries in the highest indexed locations and the smallest diagonal entries in the lowest indexed locations. We develop two fast plane rotation algorithms that avoid this unbalanced diagonal element distribution by incorporating dynamic scaling: a four-way branching algorithm and a two-way branching algorithm.

The motivation for the four-way branch algorithm is to force each diagonal element to be close to unity. Thus, the choice of the fast rotation expression will be based on whether each of the two diagonal elements is either greater than or less than unity in magnitude. This allows four possibilities that yield an absolute bound $1/\sqrt{2} \leqq |d_i| \leqq \sqrt{2}$ on the magnitude of the diagonal elements d_i's after any rotation. Table 1 presents the essential idea in the four-way branch algorithms.

The motivation for the two-way branch algorithm is to simplify the four-way branch algorithm and avoid the slow rotation, while constraining the diagonal elements from deviating far from unity. The relative sizes of the two diagonal elements, d_p and d_q, are compared before each rotation in a (p, q) plane is applied, and an appropriate choice from equations (14)–(17) is applied to achieve tighter clustering of the diagonal elements about unity. This idea is summarized in Table 2, where the typical fast rotation F_{pq} for

TABLE 1
Four-way branch.

Expression		$\lvert d_p\rvert \geqq 1$	$\lvert d_p\rvert < 1$
$\lvert d_q\rvert \geqq 1$	$\lvert\theta\rvert \leqq \frac{\pi}{4}$	$\tilde{d}_p \leftarrow cd_p$ $\tilde{d}_q \leftarrow cd_q$	$\tilde{d}_p \leftarrow d_p/c$ $\tilde{d}_q \leftarrow cd_q$
	$\lvert\theta\rvert > \frac{\pi}{4}$	$\tilde{d}_p \leftarrow sd_q$ $\tilde{d}_q \leftarrow sd_p$	$\tilde{d}_p \leftarrow sd_q$ $\tilde{d}_q \leftarrow d_p/s$
$\lvert d_q\rvert < 1$	$\lvert\theta\rvert \leqq \frac{\pi}{4}$	$\tilde{d}_p \leftarrow cd_p$ $\tilde{d}_q \leftarrow d_q/c$	$\tilde{d}_p \leftarrow 1$ $\tilde{d}_q \leftarrow 1$
	$\lvert\theta\rvert > \frac{\pi}{4}$	$\tilde{d}_p \leftarrow d_q/s$ $\tilde{d}_q \leftarrow sd_p$	$\tilde{d}_p \leftarrow 1$ $\tilde{d}_q \leftarrow 1$

TABLE 2
Two-way branch.

Expression	$\lvert\theta\rvert \leqq \frac{\pi}{4}$	$\lvert\theta\rvert > \frac{\pi}{4}$
$\lvert d_p\rvert \geqq \lvert d_q\rvert$	$\tilde{d}_p \Leftarrow cd_p$ $\tilde{d}_q \Leftarrow d_q/c$ $\alpha = csd_p/d_q$ $\beta = td_q/d_p$ $F_{pq} = \begin{pmatrix} 1 & 0 \\ \beta & 1 \end{pmatrix}\begin{pmatrix} 1 & -\alpha \\ 0 & 1 \end{pmatrix}$	$\tilde{d}_p \Leftarrow d_q/s$ $\tilde{d}_q \Leftarrow sd_p$ $\alpha = d_q/(td_p)$ $\beta = csd_p/d_q$ $F_{pq} = \begin{pmatrix} 0 & -1 \\ 1 & \alpha \end{pmatrix}\begin{pmatrix} 1 & 0 \\ -\beta & 1 \end{pmatrix}$
$\lvert d_p\rvert < \lvert d_q\rvert$	$\tilde{d}_p \Leftarrow d_p/c$ $\tilde{d}_q \Leftarrow cd_q$ $\alpha = td_p/d_q$ $\beta = csd_q/d_p$ $F_{pq} = \begin{pmatrix} 1 & -\alpha \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \beta & 1 \end{pmatrix}$	$\tilde{d}_p \Leftarrow sd_q$ $\tilde{d}_q \Leftarrow d_p/s$ $\alpha = csd_q/d_p$ $\beta = d_p/(td_q)$ $F_{pq} = \begin{pmatrix} \beta & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & \alpha \\ 0 & -1 \end{pmatrix}$

each case is also presented. In the two-way branch algorithm, if two diagonal elements are both greater than unity, the smaller will be augmented. Likewise, the larger of two subunity elements will be diminished. Thus, it allows larger deviations in the diagonal elements compared with the four-way branch algorithms. However, the results of numerical tests show that the bound in the two-way branch algorithms is kept within a small range around one.

The new fast rotations presented in relations (14)–(17) and employed in the two-way branch algorithms have the typical form

$$F_{pq} = \begin{bmatrix} 1 & \alpha \\ \beta & 1 + \alpha\beta \end{bmatrix}$$

or F_{pq} with its rows and/or columns permuted. Although only one element of F_{pq} is unity, the total multiplications required for the transformation by F_{pq} is the same as those for the standard fast rotations. This is because F_{pq} can be applied in two steps as shown in (14)–(17), e.g.,

$$F_{pq} = \begin{bmatrix} 1 & \alpha \\ \beta & 1 + \alpha\beta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix}\begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}.$$

The implementation results are obtained for the Jacobi algorithm for symmetric eigenvalue decomposition, the Hestenes algorithm for the singular value decomposition, and the QR decomposition. The purpose of the tests is to compare the deviations from unity of the elements in the diagonal factor matrix in the standard fast rotation, the de Rijk fast rotation, and the new fast rotation presented in this paper. For the eigenvalue and singular value decompositions, the total number of sweeps was assigned to be $\log n$, where n is the matrix order. The test matrices were generated with random numbers in the interval $[-1, 1]$ having a uniform distribution. In Tables 3–5, the entries are the average, for 32 tests, of the minimum and maximum of the diagonal elements throughout each test with $\log_{10}$ scaling. The standard deviations of the $\log_{10}$ scaled data are within the parentheses. For the standard fast rotation, only the minimum is shown because the diagonal elements, which are initially 1, are only decreasing. Note that for all the tests

TABLE 3[1]
Symmetric eigenvalue decomposition (32 *tests*).

$\log_{10}$ (scaling)	$N = 32$	$N = 64$	$N = 128$
min (standard)	−1.768 (0.2026)	−2.832 (0.2487)	−4.282 (0.3253)
min (de Rijk)	−1.396 (0.1404)	−2.237 (0.2538)	−3.539 (0.4733)
max (de Rijk)	1.344 (0.2209)	2.157 (0.2504)	3.369 (0.3795)
min (new)	−0.2024 (2.271E−2)	−0.2175 (2.490E−2)	−0.2330 (1.796E−2)
max (new)	0.2014 (1.135E−2)	0.2183 (2.088E−2)	0.2332 (2.189E−2)

TABLE 4[1]
Singular value decomposition (32 *tests*).

$\log_{10}$ (scaling)	$N = 32$	$N = 64$	$N = 128$
min (standard)	−1.884 (0.1984)	−3.052 (0.2149)	−4.620 (0.3295)
min (de Rijk)	−1.567 (0.2098)	−2.515 (0.2828)	−3.703 (0.3496)
max (de Rijk)	1.410 (0.2136)	2.455 (0.3099)	3.733 (0.3759)
min (new)	−0.2133 (2.259E−2)	−0.2260 (1.904E−2)	−0.2384 (1.497E−2)
max (new)	0.2066 (2.739E−2)	0.2198 (2.272E−2)	0.2308 (1.585E−2)

TABLE 5[1]
QR decomposition (32 *tests*).

$\log_{10}$ (scaling)	$N = 64$	$N = 128$	$N = 256$
min (standard)	−2.692 (0.4532)	−3.152 (0.3360)	−4.258 (0.5092)
min (de Rijk)	−0.8387 (8.995E−2)	−0.9976 (8.916E−2)	−1.141 (7.915E−2)
max (de Rijk)	1.241 (0.1949)	1.511 (0.1736)	1.858 (0.2492)
min (new)	−0.3727 (7.762E−2)	−0.3943 (7.234E−2)	−0.4491 (7.085E−2)
max (new)	0.3614 (7.7540E−2)	0.3887 (8.318E−2)	0.4404 (6.147E−2)

[1] Each entry denotes $\log_{10}$ scaling of the minimum or maximum diagonal elements in the diagonal factor matrix throughout each test. The standard deviations of the $\log_{10}$ scaled data are within the parentheses. Standard, the standard fast rotation; de Rijk, the de Rijk fast rotation; new, the new chained fast rotation with dynamic scaling.

(for the matrices of order up to 128 for the symmetric eigenvalue decomposition and the singular value decomposition and for the matrices of order up to 256 for the *QR* decomposition), the averages for the extremes of the elements in the diagonal factor of the new algorithms stay in $[\frac{1}{3}, 3]$. Also, the standard deviations for the new algorithms are extremely small. The average of the minimum diagonal elements in the standard fast algorithms diminished to as small as $10^{-4.62}$, and the range of the average extremes of the diagonal elements in the diagonal factor matrix in the de Rijk method was as large as $[10^{-3.70}, 10^{3.73}]$. We expect that as the matrix order becomes larger, the advantage of the new fast rotations will become more pronounced. The computational experiments indicate that only the new chained fast rotation algorithm can control the sizes of elements in the diagonal factor for larger matrices.

4. Error analysis.

4.1. Standard fast rotations. We first review the error analysis of the standard fast rotation for the *QR* decomposition, which is based on Parlett [12]. A rotation of two rows, p and q, may be represented as

$$\begin{bmatrix} \tilde{d}_p & 0 \\ 0 & \tilde{d}_q \end{bmatrix}\begin{bmatrix} \tilde{y}_{p1} & \tilde{y}_{p2} & \cdots & \tilde{y}_{pn} \\ 0 & \tilde{y}_{q2} & \cdots & \tilde{y}_{qn} \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}\begin{bmatrix} d_p & 0 \\ 0 & d_q \end{bmatrix}\begin{bmatrix} y_{p1} & y_{p2} & \cdots & y_{pn} \\ y_{q1} & y_{q2} & \cdots & y_{qn} \end{bmatrix},$$

where

$$\begin{bmatrix} \tilde{y}_{p1} & \tilde{y}_{p2} & \cdots & \tilde{y}_{pn} \\ 0 & \tilde{y}_{q2} & \cdots & \tilde{y}_{qn} \end{bmatrix} = \begin{bmatrix} 1 & -\alpha \\ \beta & 1 \end{bmatrix}\begin{bmatrix} y_{p1} & y_{p2} & \cdots & y_{pn} \\ y_{q1} & y_{q2} & \cdots & y_{qn} \end{bmatrix}.$$

Also, let $\tau \equiv \tan\theta$ and $\omega \equiv \cos^2\theta$.

Let ε, $|\varepsilon| \ll 1$, which may be different at every instance, be a tiny number close to *machine-ε*. The inclusion of floating point roundoff error in a computed quantity will be represented by the product of the exact theoretical value with $(1+\varepsilon)$, with the notational shorthand of representing $(1+\varepsilon_1)(1+\varepsilon_2)$ as $(1+\varepsilon)^2$. We will assume that the initial variables are exact, and computed quantities will be denoted by primes.

The computed scalars are

$$\begin{aligned} \alpha' &= \alpha(1+\varepsilon) && \{= y_{q1}/y_{p1}\}, \\ \beta' &= \beta(1+\varepsilon)^3 && \{= \alpha' d_q^2/d_p^2\}, \\ \omega' &= \omega(1+\varepsilon)^6 && \left\{= 1\Big/\left(1+\left(\frac{d_q y_q}{d_p y_p}\right)^2\right)\right\} \\ &\approx (c(1+\varepsilon)^3)^2. \end{aligned} \tag{21}$$

Although we have only the squared values of d_p and d_q in actual computation, we will use d_p and d_q in this analysis.

Then, after the rotation,

$$\begin{Bmatrix} \tilde{d}'_p = d_p c(1+\varepsilon)^4 \\ \tilde{d}'_q = d_q c(1+\varepsilon)^4 \end{Bmatrix}, \tag{22}$$

and the vector equations are

$$\begin{Bmatrix} \tilde{y}'_{pi} = [y_{pi} + \beta' y_{qi}(1+\varepsilon)](1+\varepsilon) = y_{pi}(1+\varepsilon) + \beta y_{qi}(1+\varepsilon)^5 \\ \tilde{y}'_{qi} = [y_{qi} - \alpha' y_{pi}(1+\varepsilon)](1+\varepsilon) = y_{qi}(1+\varepsilon) - \alpha y_{pi}(1+\varepsilon)^3 \end{Bmatrix}. \tag{23}$$

From (22) and (23),

$$\left.\begin{aligned} \tilde{d}'_p\tilde{y}'_{pi} &= d_p c y_{pi}(1+\varepsilon)^5 + d_p c\beta y_{qi}(1+\varepsilon)^9 \\ \tilde{d}'_q\tilde{y}'_{qi} &= d_q c y_{qi}(1+\varepsilon)^5 - d_q c\alpha y_{pi}(1+\varepsilon)^7 \end{aligned}\right\}, \tag{24}$$

and using the first order approximation, $1 + k\varepsilon \approx (1+\varepsilon)^k$, for any integer k,

$$\left.\begin{aligned} \tilde{d}'_p\tilde{y}'_{pi} &= d_p c y_{pi}(1+5\varepsilon) + d_q s y_{qi}(1+9\varepsilon) \\ \tilde{d}'_q\tilde{y}'_{qi} &= d_q c y_{qi}(1+5\varepsilon) - d_p s y_{pi}(1+7\varepsilon) \end{aligned}\right\}. \tag{25}$$

Extracting the error terms, we have the equation

$$\begin{bmatrix} \tilde{d}'_p\tilde{y}'_{pi} \\ \tilde{d}'_q\tilde{y}'_{qi} \end{bmatrix} = \begin{bmatrix} \tilde{d}_p\tilde{y}_{pi} \\ \tilde{d}_q\tilde{y}_{qi} \end{bmatrix} + \begin{bmatrix} 5c\varepsilon & 9s\varepsilon \\ -7s\varepsilon & 5c\varepsilon \end{bmatrix} \begin{bmatrix} d_p y_{pi} \\ d_q y_{qi} \end{bmatrix}, \tag{26}$$

which is of exactly the same form as the equation for the errors incurred in a standard Givens rotation [6]. The same analysis and conclusion holds for the large angle formulas. Equation (26) may be reformulated as the inequality

$$\left\| \begin{bmatrix} \tilde{d}'_p\tilde{y}'_{pi} \\ \tilde{d}'_q\tilde{y}'_{qi} \end{bmatrix} - \begin{bmatrix} \tilde{d}_p\tilde{y}_{pi} \\ \tilde{d}_q\tilde{y}_{qi} \end{bmatrix} \right\| \leqq \left\| \begin{bmatrix} 5c\varepsilon & 9s\varepsilon \\ 7s\varepsilon & 5c\varepsilon \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} d_p y_{pi} \\ d_q y_{qi} \end{bmatrix} \right\|, \tag{27}$$

where $|A| = (|a_{ij}|)$.

4.2. New fast rotations. To differentiate the computed scalars in the new fast rotations from those of the standard fast rotations, we will use the *hat* notation ($\hat{}$) whenever it is necessary. The computed scalars in the new fast rotations are the same as those in (21) except that α is redefined as $\hat{\alpha}$:

$$\hat{\alpha}' = \hat{\alpha}(1+\varepsilon)^8 \quad \{= \alpha'\omega' = \alpha\omega(1+\varepsilon)^8\}. \tag{28}$$

As before, the unsquared values of d_p and d_q will be used. However, the change from d_q to $\hat{d}_q$ should be noted:

$$\left.\begin{aligned} \tilde{d}'_p &= d_p c(1+\varepsilon)^4 \\ \hat{d}'_q &= (d_q/c)(1+\varepsilon)^4 \end{aligned}\right\}. \tag{29}$$

The chained vector equations are:

$$\left.\begin{aligned} \tilde{y}'_{pi} &= [y_{pi} + \beta' y_{qi}(1+\varepsilon)](1+\varepsilon) \\ \hat{y}'_{qi} &= [y_{qi} - \hat{\alpha}'\tilde{y}'_{pi}(1+\varepsilon)](1+\varepsilon) \end{aligned}\right\}. \tag{30}$$

Expanding (30), we get

$$\left.\begin{aligned} \tilde{y}'_{pi} &= y_{pi}(1+\varepsilon) + \beta y_{qi}(1+\varepsilon)^5 \\ \hat{y}'_{qi} &= y_{qi}(1+\varepsilon) - \alpha\omega[y_{pi}(1+\varepsilon) + \beta y_{qi}(1+\varepsilon)^5](1+\varepsilon)^{10} \\ &= [1 - \alpha\omega\beta(1+\varepsilon)^{14}]y_{qi}(1+\varepsilon) - \alpha\omega y_{pi}(1+\varepsilon)^{11} \end{aligned}\right\}. \tag{31}$$

Factoring in the diagonal elements by merging (29) and (31) with approximation of ε terms up to the first order, we have (noting that $(1 - \alpha\omega\beta(1+\varepsilon)^{14})(1+\varepsilon) \approx (c^2(1+\varepsilon) - 14s^2\varepsilon)$),

$$\left.\begin{aligned} \tilde{d}'_p\tilde{y}'_{pi} &= d_p c y_{pi}(1+5\varepsilon) + d_q s y_{qi}(1+9\varepsilon) \\ \hat{d}'_q\hat{y}'_{qi} &= d_q y_{qi}[c(1+5\varepsilon) - 14s\tau\varepsilon] - d_p s y_{pi}(1+15\varepsilon) \end{aligned}\right\}. \tag{32}$$

Separating out the error terms,

$$\begin{bmatrix} \tilde{d}'_p \tilde{y}'_{pi} \\ \hat{d}'_q \hat{y}'_{qi} \end{bmatrix} = \begin{bmatrix} \tilde{d}_p \tilde{y}_{pi} \\ \hat{d}_q \hat{y}_{qi} \end{bmatrix} + \begin{bmatrix} 5c\varepsilon & 9s\varepsilon \\ -15s\varepsilon & (5c - 15s\tau)\varepsilon \end{bmatrix} \begin{bmatrix} d_p y_{pi} \\ d_q y_{qi} \end{bmatrix}. \tag{33}$$

Because $|\theta| \leqq \frac{\pi}{4}$, $|5c - 14s\tau| \leqq |9c|$, and the following inequality may be formulated

$$\left\| \begin{bmatrix} \tilde{d}'_p \tilde{y}'_{pi} \\ \hat{d}'_q \hat{y}'_{qi} \end{bmatrix} - \begin{bmatrix} \tilde{d}_p \tilde{y}_{pi} \\ \hat{d}_q \hat{y}_{qi} \end{bmatrix} \right\| \leqq \left\| \begin{bmatrix} 5c\varepsilon & 9s\varepsilon \\ 15s\varepsilon & 9c\varepsilon \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} d_p y_{pi} \\ d_q y_{qi} \end{bmatrix} \right\|. \tag{34}$$

The error analysis for the fast rotation when the qth row is updated first is procedurally identical and yields proportional results:

$$\hat{\beta}' = \hat{\beta}(1 + \varepsilon)^{10} \quad \{ = \beta' \omega' = \beta\omega(1 + \varepsilon)^{10} \}, \tag{35}$$

$$\left\{ \begin{aligned} \tilde{d}'_q &= d_q c(1 + \varepsilon)^4 \\ \hat{d}'_p &= (d_p / c)(1 + \varepsilon)^4 \end{aligned} \right\}, \tag{36}$$

$$\left\{ \begin{aligned} \tilde{y}'_{qi} &= [y_{qi} - \alpha' y_{pi}(1 + \varepsilon)](1 + \varepsilon) \\ \hat{y}'_{pi} &= [y_{pi} - \hat{\alpha}' \tilde{y}'_{qi}(1 + \varepsilon)](1 + \varepsilon) \end{aligned} \right\}, \tag{37}$$

$$\left\{ \begin{aligned} \tilde{y}'_{qi} &= y_{qi}(1 + \varepsilon) - \alpha y_{pi}(1 + \varepsilon)^3 \\ \hat{y}'_{pi} &= y_{pi}(1 + \varepsilon) + \beta\omega[y_{qi}(1 + \varepsilon) - \alpha y_{pi}(1 + \varepsilon)^3](1 + \varepsilon)^{12} \\ &= [1 - \alpha\omega\beta(1 + \varepsilon)^{14}] y_{pi}(1 + \varepsilon) + \beta\omega y_{qi}(1 + \varepsilon)^{13} \end{aligned} \right\}, \tag{38}$$

$$\left\{ \begin{aligned} \tilde{d}'_q \tilde{y}'_{qi} &= d_q c y_{qi}(1 + 5\varepsilon) - d_p s y_{pi}(1 + 7\varepsilon) \\ \hat{\tilde{d}}'_p \hat{y}'_{pi} &= d_p y_{pi}[c(1 + 5\varepsilon) - 14s\tau\varepsilon] + d_q s y_{pi}(1 + 17\varepsilon) \end{aligned} \right\}, \tag{39}$$

$$\begin{bmatrix} \tilde{d}'_q \tilde{y}'_{qi} \\ \hat{d}'_p \hat{y}'_{pi} \end{bmatrix} = \begin{bmatrix} \tilde{d}_q \tilde{y}_{qi} \\ \hat{d}_p \hat{y}_{pi} \end{bmatrix} + \begin{bmatrix} 5c\varepsilon & -7s\varepsilon \\ 17s\varepsilon & (5c - 14s\tau)\varepsilon \end{bmatrix} \begin{bmatrix} d_q y_{qi} \\ d_p y_{pi} \end{bmatrix}. \tag{40}$$

Finally, (40) yields the inequality,

$$\left\| \begin{bmatrix} \tilde{d}'_q \tilde{y}'_{qi} \\ \hat{d}'_p \hat{y}'_{pi} \end{bmatrix} - \begin{bmatrix} \tilde{d}_q \tilde{y}_{qi} \\ \hat{d}_p \hat{y}_{pi} \end{bmatrix} \right\| \leqq \left\| \begin{bmatrix} 5c\varepsilon & 7s\varepsilon \\ 17s\varepsilon & 9c\varepsilon \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} d_q y_{qi} \\ d_p y_{pi} \end{bmatrix} \right\|. \tag{41}$$

As the derivations (for the error analyses of the large angle formulas) are similar to those of the small angle formulas, we state only the results. For $|\theta| > \frac{\pi}{4}$ and when the qth row is updated based on the updated pth row ($p < q$),

$$\begin{bmatrix} \tilde{d}'_p \tilde{y}'_{pi} \\ \hat{d}'_q \hat{y}'_{qi} \end{bmatrix} = \begin{bmatrix} \tilde{d}_p \tilde{y}_{pi} \\ \hat{d}_q \hat{y}_{qi} \end{bmatrix} + \begin{bmatrix} 9c\varepsilon & 5s\varepsilon \\ -(5s - 14c\tau^{-1})\varepsilon & 15c\varepsilon \end{bmatrix} \begin{bmatrix} d_p y_{pi} \\ d_q y_{qi} \end{bmatrix}, \tag{42}$$

$$\left\| \begin{bmatrix} \tilde{d}'_p \tilde{y}'_{pi} \\ \hat{d}'_q \hat{y}'_{qi} \end{bmatrix} - \begin{bmatrix} \tilde{d}_p \tilde{y}_{pi} \\ \hat{d}_q \hat{y}_{qi} \end{bmatrix} \right\| \leqq \left\| \begin{bmatrix} 9c\varepsilon & 5s\varepsilon \\ 9s\varepsilon & 15c\varepsilon \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} d_p y_{pi} \\ d_q y_{qi} \end{bmatrix} \right\|, \tag{43}$$

and, for the counter-chained rotation (the pth row is updated based on the updated qth row, $p < q$),

$$\begin{bmatrix} \tilde{d}'_q \tilde{y}'_{qi} \\ \hat{d}'_p \hat{y}'_{pi} \end{bmatrix} = \begin{bmatrix} \tilde{d}_q \tilde{y}_{qi} \\ \hat{d}_p \hat{y}_{pi} \end{bmatrix} + \begin{bmatrix} 7c\varepsilon & -5s\varepsilon \\ (5s - 14c\tau^{-1})\varepsilon & 17c\varepsilon \end{bmatrix} \begin{bmatrix} d_q y_{qi} \\ d_p y_{pi} \end{bmatrix}, \tag{44}$$

$$\left\| \begin{bmatrix} \tilde{d}'_q \tilde{y}'_{qi} \\ \hat{d}'_p \hat{y}'_{pi} \end{bmatrix} - \begin{bmatrix} \tilde{d}_q \tilde{y}_{qi} \\ \hat{d}_p \hat{y}_{pi} \end{bmatrix} \right\| \leqq \left\| \begin{bmatrix} 7c\varepsilon & 5s\varepsilon \\ 9s\varepsilon & 17c\varepsilon \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} d_q y_{qi} \\ d_p y_{pi} \end{bmatrix} \right\|. \tag{45}$$

As we can see from (34), (41), (43), and (45), because of the chaining between the two linked triads, there is the potential for roughly doubled error in the vector that is updated second, compared with the standard fast rotations. However, the rotation is still stable, as the magnitudes of the elements of the error matrix are bounded by a small constant. For stability, it is important that the small angle formulas must be used when $|\tau| \leqq 1$ and the large angle formulas are required otherwise. Error analyses of other algorithms that employ fast rotations, e.g., the Jacobi and Hestenes algorithms, would proceed similarly and would yield similar stability results.

5. Remarks. We have presented new fast plane rotation algorithms that solve the long-standing overflow and underflow problem that is inherent in the standard fast rotations. The new fast rotation algorithms dynamically scale the elements in the diagonal factor matrix by using a simple comparison for each rotation. An additional advantage of the new fast rotations comes from the fact that the temporary vector copy that is necessary in the standard fast rotations is also eliminated. Accordingly, vector updates can be chained and higher efficiency can be achieved especially on vector processors.

It is essential, for scaling and the numerical stability of the new fast rotations, to choose the appropriate rotation based on the relative size of the angle with $\pm\frac{\pi}{4}$. Certain algorithms, such as the Jacobi algorithm for the symmetric eigenvalue decomposition, do not require rotation angles to exceed $[-\frac{\pi}{4}, \frac{\pi}{4}]$, simplifying the choices in fast rotations. We have shown that the scaling in the chained fast rotation is highly dependent on the direction of the chaining, i.e., which of the two vectors uses the updated value of the other vector for its own update. Although the four-way branch algorithm (Table 1) can guarantee that the diagonal elements in the diagonal factor matrix are in the range $[1/\sqrt{2}, \sqrt{2}]$ at any stage, it can be much slower than the two-way branch algorithm because the standard slow rotations are occasionally required. The two-way branch algorithm (Table 2) is simpler and more efficient. Although we do not have any rigorous proof that the diagonal elements stay in a constant range in the two-way branch algorithm, it provides excellent control of scaling according to our substantial numerical tests. We have found no instance for which the two-way branch algorithm performs poorly in scaling of the diagonal factor. For the case of the QR decomposition, we have shown that the chained fast rotation algorithms have stability that is essentially the same as that of the standard fast rotations.

REFERENCES

[1] J. L. BARLOW, *Stability analysis of the G-algorithm and a note on its application to sparse least squares problems*, BIT, 25 (1985), pp. 507–520.

[2] J. L. BARLOW AND I. C. F. IPSEN, *Scaled Givens rotations for the solution of linear least squares problems on systolic arrays*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 716–733.

[3] P. P. M. DE RIJK, *A one-sided Jacobi algorithm for computing the singular value decomposition on a vector computer*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 359–371.

[4] P. J. EBERLEIN AND H. PARK, *Efficient implementation of Jacobi algorithms and Jacobi sets on distributed memory architectures*, J. Par. Dist. Comput., special issue on Algorithms for Hypercube Computers, 8 (1990), pp. 358–366.

[5] W. M. GENTLEMAN, *Least squares computations by Givens transformations without square roots*, J. Inst. Math. Appl., 12 (1973), pp. 329–336.

[6] ———, *Error analysis of QR decomposition by Givens rotations*, Linear Algebra Appl., 10 (1975), pp. 189–197.

[7] G. H. GOLUB AND C. F. VAN LOAN, Matrix Computations, Second ed., Johns Hopkins Series in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, 1989.

[8] J. GÖTZE AND U. SCHWIEGELSHOHN, *A square root and division free Givens rotation for solving least squares problems on systolic arrays*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 800–807.

[9] S. HAMMARLING, *A note on modifications to the Givens plane rotation*, J. Inst. Math. Appl., 13 (1974), pp. 215–218.

[10] M. R. HESTENES, *Inversion of matrices by biorthogonalization and related results*, J. Soc. Indust. Appl. Math., 6 (1958), pp. 51–90.

[11] H. PARK AND P. J. EBERLEIN, *Factored Jacobi-like algorithms for eigensystem computations on vector processors*, Technical report TR90-11, Computer Science Dept., Univ. of Minnesota, Minneapolis, 1990.

[12] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, First ed., Prentice Hall Series in Computational Mathematics, Prentice Hall, Englewood Cliffs, NJ, 1980.

[13] W. RATH, *Fast Givens rotations for orthogonal similarity transformations*, Numer. Math., 40 (1982), pp. 47–56.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 175–184, January 1994

014

POSITIVE DEFINITENESS AND STABILITY OF INTERVAL MATRICES*

JIRI ROHN†

Abstract. Characterizations of positive definiteness, positive semidefiniteness, and Hurwitz and Schur stability of interval matrices are given. First it is shown that an interval matrix has some of the four properties if and only if this is true for a finite subset of explicitly described matrices, and some previous results of this type are improved. Second it is proved that a symmetric interval matrix is positive definite (Hurwitz stable, Schur stable) if and only if it contains at least one symmetric matrix with the respective property and is nonsingular (for Schur stability, two interval matrices are to be nonsingular). As a consequence, verifiable sufficient conditions are obtained for positive definiteness and Hurwitz and Schur stability of symmetric interval matrices.

Key words. interval matrix, positive definiteness, positive semidefiniteness, Hurwitz stability, Schur stability, nonsingularity

AMS subject classifications. 15A18, 15A48, 65G10, 93D09

Introduction. In this paper we study positive definiteness, positive semidefiniteness, and Hurwitz and Schur stability of square interval matrices defined in the following way: an interval matrix A^I is said to be positive definite (positive semidefinite, Hurwitz stable) if each matrix $A \in A^I$ is positive definite (positive semidefinite, Hurwitz stable); a slight deviation from this definition is made for Schur stability where a symmetric interval matrix A^I is said to be Schur stable if each *symmetric* $A \in A^I$ is Schur stable. Positive (semi)definiteness of interval matrices is studied in § 2, Hurwitz stability in § 3, and Schur stability in § 4. There are two main streams of results that run across these sections.

First, we show that for each of the four properties listed it holds that A^I (assumed to be symmetric in stability cases) has the property if and only if this is true for a finite subset of explicitly described matrices in A^I. The result for positive (semi)definiteness is given in Theorem 2, where the respective subset is shown to be of cardinality 2^{n-1} (in the worst case) for an $n \times n$ interval matrix A^I; this theorem improves considerably the earlier result by Shi and Gao [13], which used $2^{n(n-1)/2}$ test matrices. A similar result is given in Theorem 6 for Hurwitz stability of symmetric interval matrices, which is again characterized by a subset of matrices of cardinality 2^{n-1}. Hertz [6] has recently proved that stability of this subset implies stability of each symmetric matrix in A^I; our result shows that stability of this subset already implies stability of the whole of A^I.

Second, we show that a symmetric interval matrix A^I is positive definite (Hurwitz stable, Schur stable) if and only if it contains at least one symmetric matrix with the respective property and is regular (for Schur stability, two associated interval matrices are to be regular; A^I is called regular [9] if each $A \in A^I$ is nonsingular). These results, proved in Theorems 3, 8, and 11, reduce the number of test matrices to one but do not remove exponentiality from the verification process because all the necessary and sufficient regularity conditions known ([9], [12]) employ some subset of test matrices whose cardinality is exponential in the matrix size. Nevertheless, because there exists a sufficient regularity condition due to Beeck [2], which is known to cover most practical examples, employing it in the above characterizations leads to sufficient conditions for positive definiteness and Hurwitz and Schur stability of symmetric interval matrices (Theorems 4, 9, and 12), which can be expected to work well in practical cases. In the final remark in § 5, we give a modification of the Beeck's condition that enables us to use an approx-

* Received by the editors September 16, 1991; accepted for publication (in revised form) May 21, 1992.
† Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, 11800 Prague, Czech Republic (rohn@cspguk11.bitnet).

imation of the inverse of the center matrix of A^I instead of the exact inverse as required in the original formulation.

1. Notations and auxiliary results. We introduce some notations and prove a theorem that sums up the basic technical results to be used later in the proofs of the main theorems.

For a square real matrix $A = (a_{ij})$, we denote the transpose by A^T, the spectral radius by $\rho(A)$, and we introduce its absolute value as the matrix $|A| = (|a_{ij}|)$. A matrix A is called symmetric if $A = A^T$. Symmetric matrices are known to have all eigenvalues real; we shall denote by $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$, the minimum and maximum eigenvalue of A, respectively (obviously, $\lambda_{\min}(-A) = -\lambda_{\max}(A)$). Matrix inequalities, as $A \leqq B$ or $A < B$, are to be understood componentwise.

Let A_c and Δ be real $n \times n$ matrices, $\Delta \geqq 0$. The set of matrices

$$A^I = [A_c - \Delta, A_c + \Delta] = \{A; A_c - \Delta \leqq A \leqq A_c + \Delta\}$$

is called an interval matrix. A^I is said to be symmetric if both A_c and Δ are symmetric. With each interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ we shall associate the symmetric interval matrix

$$A_s^I = [A_c' - \Delta', A_c' + \Delta'],$$

where A_c' and Δ' are given by

$$A_c' = \tfrac{1}{2}(A_c + A_c^T)$$

and

$$\Delta' = \tfrac{1}{2}(\Delta + \Delta^T).$$

Obviously, if $A \in A^I$, then $\tfrac{1}{2}(A + A^T) \in A_s^I$ and A^I is symmetric if and only if $A^I = A_s^I$.

We introduce an auxiliary index set

$$Y = \{z \in R^n; |z_j| = 1 \quad \text{for } j = 1, \ldots, n\},$$

i.e., Y is the set of all ± 1-vectors; hence, its cardinality is 2^n. For each $z \in Y$ we shall denote by T_z the $n \times n$ diagonal matrix with diagonal vector z. Now for each $z \in Y$ let us define the matrix A_z by

$$A_z = A_c - T_z \Delta T_z.$$

Then for each i, j we have $(A_z)_{ij} = (A_c)_{ij} - z_i \Delta_{ij} z_j = (A_c - \Delta)_{ij}$ if $z_i z_j = 1$ and $(A_z)_{ij} = (A_c + \Delta)_{ij}$ if $z_i z_j = -1$; hence, $A_z \in A^I$ for each $z \in Y$ and because $A_{-z} = A_z$, the number of mutually different matrices A_z is at most 2^{n-1} (and equal to 2^{n-1} if $\Delta > 0$). If A^I is symmetric, then each A_z is symmetric. The matrices A_z, $z \in Y$, will be used in § 2 to characterize positive (semi)definiteness of an interval matrix by finite means.

Let us now introduce a function f: $R^{n \times n} \to R^1$ defined for a matrix $A \in R^{n \times n}$ by

$$f(A) = \min_{x \neq 0} \frac{x^T A x}{x^T x}. \tag{1}$$

Obviously, f is well defined. In the following theorem we sum up the basic properties of f that will be used in the proofs of the main theorems in the subsequent sections.

THEOREM 1. *The function f has the following properties*:

(i) $f(A) = f(\tfrac{1}{2}(A + A^T))$ *for each* $A \in R^{n \times n}$;

(ii) $f(A) = \lambda_{\min}(A)$ *for each symmetric* $A \in R^{n \times n}$;

(iii) $|f(A + D) - f(A)| \leqq \rho(\tfrac{1}{2}(D + D^T))$ *for each* $A, D \in R^{n \times n}$;

(iv) *f is continuous in $R^{n\times n}$*;

(v) *for each interval matrix A^I we have*

$$\min\{f(A); A\in A^I\} = \min\{f(A_z); z\in Y\};$$

(vi) *for each interval matrix A^I we have*

$$\min\{f(A); A\in A^I\} = \min\{f(A); A\in A_s^I\};$$

(vii) *each interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ satisfies*

$$\min\{f(A); A\in A^I\} \geqq f(A_c) - \rho(\Delta');$$

(viii) *if A is symmetric and $f(A) = 0$, then A is singular.*

Proof. (i) follows from the fact that $x^TAx = x^T(\frac{1}{2}(A + A^T))x$ for each $A\in R^{n\times n}$ and $x\in R^n$. (ii) is well known (cf., e.g., Parlett [10]). To prove (iii), first observe that from (1) it follows

$$f(A + D) \geqq f(A) + f(D)$$

for each A and D; this inequality implies

$$f(A) = f((A + D) + (-D)) \geqq f(A + D) + f(-D),$$

which together gives

$$\begin{aligned} |f(A + D) - f(A)| &\leqq \max\{|f(D)|, |f(-D)|\} \\ &= \max\{|f(\tfrac{1}{2}(D + D^T))|, |f(-\tfrac{1}{2}(D + D^T))|\} \\ &= \max\{|\lambda_{\min}(\tfrac{1}{2}(D + D^T))|, |\lambda_{\max}(\tfrac{1}{2}(D + D^T))|\} \\ &= \rho(\tfrac{1}{2}(D + D^T)). \end{aligned}$$

For (iv) take a matrix norm $\|\cdot\|$ such that $\|A^T\| = \|A\|$ for each A. Then from (iii) we obtain

$$|f(A + D) - f(A)| \leqq \|\tfrac{1}{2}(D + D^T)\| \leqq \|D\|$$

for each A and D, which proves that f is continuous in $R^{n\times n}$.

To prove (v), let $A\in A^I$ and $x\neq 0$. Because $|x^T(A - A_c)x| \leqq |x|^T\Delta|x|$, we obtain $x^TAx = x^TA_cx + x^T(A - A_c)x \geqq x^TA_cx - |x|^T\Delta|x|$. Define a $z\in Y$ as follows: $z_j = 1$ if $x_j \geqq 0$ and $z_j = -1$ otherwise ($j = 1, \ldots, n$), then $|x| = T_zx$ and we have

$$x^TAx \geqq x^TA_cx - x^TT_z\Delta T_zx = x^TA_zx;$$

hence,

$$\frac{x^TAx}{x^Tx} \geqq \frac{x^TA_zx}{x^Tx} \geqq f(A_z) \geqq \min\{f(A_z); z\in Y\},$$

which implies that

$$f(A) \geqq \min\{f(A_z); z\in Y\}$$

holds for each $A\in A^I$ and because $A_z\in A^I$ for each $z\in Y$, the assertion follows.

To prove (vi), for each $z\in Y$ denote by A'_z the matrix A_z for A_s^I, i.e.,

$$A'_z = A'_c - T_z\Delta'T_z = \tfrac{1}{2}(A_c + A_c^T) - T_z(\tfrac{1}{2}(\Delta + \Delta^T))T_z = \tfrac{1}{2}(A_z + A_z^T).$$

Then employing (i) we obtain

$$f(A_z) = f(\tfrac{1}{2}(A_z + A_z^T)) = f(A'_z);$$

hence, the assertion (v) implies that the minimum values of f over A^I and A_s^I are equal.

For (vii) let $A \in A^I$. Since $|A - A_c| \leqq \Delta$, using (iii) and Proposition 3.2.4 in [9] we obtain $|f(A) - f(A_c)| \leqq \rho(\frac{1}{2}(A - A_c) + \frac{1}{2}(A - A_c)^T) \leqq \rho(\frac{1}{2}(\Delta + \Delta^T)) = \rho(\Delta')$, which gives $f(A) \geqq f(A_c) - \rho(\Delta')$ and thus also

$$\min \{f(A); A \in A^I\} \geqq f(A_c) - \rho(\Delta').$$

For (viii) under the assumptions, zero is an eigenvalue of A due to (ii), hence A is singular. □

2. Positive (semi)definiteness of interval matrices. A square (not necessarily symmetric) matrix A is called positive semidefinite if $f(A) \geqq 0$, which, in view of (1), means that $x^T Ax \geqq 0$ for each x (hence our definition conforms to the usual one). Similarly, A is said to be positive definite if $f(A) > 0$ (i.e., $x^T Ax > 0$ for each $x \neq 0$). An *interval matrix* A^I is said to be positive (semi)definite if each $A \in A^I$ is positive (semi)definite. As a consequence of Theorem 1 we obtain this characterization.

THEOREM 2. *Let A^I be a square interval matrix. Then the following assertions are equivalent*:

(a) *A^I is positive (semi)definite*,

(b) *A_S^I is positive (semi)definite*,

(c) *A_z is positive (semi)definite for each $z \in Y$.*

Proof. We shall prove the theorem for the case of positive definiteness of A^I; the proof for positive semidefiniteness runs quite analogously. By definition, A^I is positive definite if and only if

$$\min \{f(A); A \in A^I\} > 0$$

holds. Then the equivalence of (a) and (b) follows from the assertion (vi) of Theorem 1 and that of (a) and (c) from the assertion (v) of the same theorem. □

The assertion (c) shows that positive (semi)definiteness of an interval matrix can be verified by testing 2^{n-1} matrices from A^I for positive (semi)definiteness. Hence, this theorem improves considerably the earlier result by Shi and Gao [13], which required testing $2^{n(n-1)/2}$ matrices from A^I (the so-called vertex matrices) for positive (semi)definiteness; moreover, their result was given for symmetric interval matrices only. We note that Białas and Garloff [4] proved a similar characterization of interval P-matrices (each $A \in A^I$ is a P-matrix if and only if each A_z, $z \in Y$ is a P-matrix), although they did not explicitly use the matrices A_z.

The equivalence "$(a) \Leftrightarrow (b)$" reveals another important property, namely that verification of positive (semi)definiteness of A^I always can be performed by inspecting the associated symmetric interval matrix A_S^I; hence, we can restrict our attention in the sequel to symmetric interval matrices only. First we have this corollary.

COROLLARY. *Let a symmetric interval matrix A^I be positive semidefinite. Then it is positive definite if and only if all the matrices A_z, $z \in Y$ are nonsingular.*

Proof. The "only if" part is obvious because each positive definite matrix is nonsingular. To prove the "if" part, assume to the contrary that A^I is positive semidefinite but not positive definite. Then from the assertion (c) of Theorem 2 it follows that there exists a matrix A_z that is positive semidefinite but not positive definite. Then $f(A_z) = 0$ and because A_z is symmetric, we have that A_z is singular (Theorem 1, (viii)), which is a contradiction. □

In the next theorem we prove that positive definiteness of symmetric interval matrices is closely related to regularity. Let us recall that a square interval matrix A^I is called regular [9] if each $A \in A^I$ is nonsingular.

THEOREM 3. *A symmetric interval matrix A^I is positive definite if and only if it is regular and contains at least one positive definite matrix.*

Proof. Again, the "only if" part is obvious. In the proof of the "if" part, assume to the contrary that A^I is regular and contains a positive definite matrix A_0 but is not positive definite, so that $x^T A_1 x \leqq 0$ for some $A_1 \in A^I$ and $x \neq 0$. Define $\tilde{A}_0 = \frac{1}{2}(A_0 + A_0^T)$ and $\tilde{A}_1 = \frac{1}{2}(A_1 + A_1^T)$, then both $\tilde{A}_0$ and $\tilde{A}_1$ are symmetric, belong to A^I, and satisfy

$$f(\tilde{A}_0) = f(A_0) > 0$$

and

$$f(\tilde{A}_1) = f(A_1) \leqq 0.$$

Now define a real function φ of one real variable by

$$\varphi(t) = f(t\tilde{A}_0 + (1 - t)\tilde{A}_1), \qquad t \in [0, 1].$$

Then φ is continuous by the assertion (iv) of Theorem 1 and because $\varphi(0)\varphi(1) = f(\tilde{A}_1)f(\tilde{A}_0) \leqq 0$, there exists a $t_0 \in [0, 1]$ with $\varphi(t_0) = 0$. Put

$$A = t_0\tilde{A}_0 + (1 - t_0)\tilde{A}_1,$$

then A is symmetric, $A \in A^I$ and $f(A) = 0$; hence, the assertion (viii) of Theorem 1 gives that A is singular, which is a contradiction. □

The necessary and sufficient condition of Theorem 3 requires only one matrix to be tested for positive definiteness. It bears a striking similarity with the characterization of nonnegative invertibility of interval matrices given in [11], Theorem 1 (each $A \in A^I$ is nonnegative invertible if and only if A^I is regular and $(A_c + \Delta)^{-1} \geqq 0$). However, the result is not as pleasant as it might seem because verifying regularity of an interval matrix is generally a difficult problem as it can be clearly seen from Theorem 5.1 in [12], where a number of necessary and sufficient regularity conditions are given, all of which require computation of at least 2^{n-1} quantities of some sort (as evaluating determinants, solving systems of linear equations, inverting matrices, and so on). Nevertheless, there exists an easily verifiable sufficient regularity condition that, in this author's experience, covers most practical examples. Employing it in Theorem 3 leads to this sufficient condition.

THEOREM 4. *Let $A^I = [A_c - \Delta, A_c + \Delta]$ be a symmetric interval matrix such that A_c is positive definite and*

$$\rho(|A_c^{-1}|\Delta) < 1 \tag{2}$$

holds. Then A^I is positive definite.

Proof. Because A_c is positive definite, it is invertible and the condition (2) guarantees regularity of A^I (see Beeck [2]). Hence, Theorem 3 gives that A^I is positive definite. □

We also note that if $(|A_c^{-1}|\Delta)_{jj} \geqq 1$ for some j, then A^I contains a singular matrix (assertion (iii) of Corollary 5.1 in [12]); hence, A^I is not positive definite.

Another sufficient condition can be derived from Theorem 1.

THEOREM 5. *Let a symmetric interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ satisfy*

$$\rho(\Delta) \leqq \lambda_{\min}(A_c). \tag{3}$$

Then A^I is positive semidefinite. Moreover, if the inequality (3) holds sharply, then A^I is positive definite.

Proof. According to the assertions (vii) and (ii) of Theorem 1, we have $\min\{f(A); A \in A^I\} \geqq \lambda_{\min}(A_c) - \rho(\Delta) \geqq 0$; hence, $f(A) \geqq 0$ for each $A \in A^I$, so that A^I is positive semidefinite. If (3) holds sharply, then $f(A) > 0$ for each $A \in A^I$; hence, A^I is positive definite. □

In the next section we shall apply the results obtained to characterize stability of symmetric interval matrices.

3. Hurwitz stability of interval matrices. A square matrix A is called Hurwitz stable (for the sake of brevity, we shall say only "stable") if $\operatorname{Re}\lambda < 0$ for each eigenvalue λ of A (in other words, if all its eigenvalues lie in the open left half of the complex plane). An interval matrix A^I is said to be stable if each $A \in A^I$ is stable. The problem of stability of interval matrices arises naturally in control theory in connection with the behavior of a linear time invariant system $\dot{x}(t) = Ax(t)$ under data perturbations and has been extensively studied recently; we refer the reader to the survey paper by Mansour [8] for a detailed list of references. We investigate here mainly stability of symmetric interval matrices, which turns out to be closely connected to the contents of the previous section due to the well-known result that states a symmetric matrix A is stable if and only if $-A$ is positive definite (see, e.g., [5]). However, some care must be taken because a symmetric interval matrix can contain nonsymmetric matrices whose eigenvalues are not real. As an example, consider the symmetric interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ with $A_c = 0$ and

$$\Delta = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

which contains the matrix

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

whose eigenvalues are $\pm i$.

In contrast to the previous section where we employed the matrices $A_z = A_c - T_z\Delta T_z$, here we shall characterize stability in terms of matrices

$$\bar{A}_z = A_c + T_z\Delta T_z, \qquad z \in Y.$$

Obviously, $\bar{A}_z \in A^I$ and all $\bar{A}_z$ are symmetric if A^I is symmetric.

THEOREM 6. *Let $A^I = [A_c - \Delta, A_c + \Delta]$ be a symmetric interval matrix. Then the following assertions are equivalent*:

(a) *A^I is stable*,

(b) *$[-A_c - \Delta, -A_c + \Delta]$ is positive definite*,

(c) *$\bar{A}_z$ is stable for each $z \in Y$.*

Proof. We shall prove that $(a) \Rightarrow (c) \Rightarrow (b) \Rightarrow (a)$. Let us denote $A_0^I = [-A_c - \Delta, -A_c + \Delta]$; notice that $A_0^I = \{-A; A \in A^I\}$.

$(a) \Rightarrow (c)$: The proof is obvious because $\bar{A}_z \in A^I$ for each $z \in Y$.

$(c) \Rightarrow (b)$: Let $z \in Y$. Because $\bar{A}_z$ is symmetric and stable, it follows that all its eigenvalues are negative; hence, the symmetric matrix

$$-\bar{A}_z = -A_c - T_z\Delta T_z$$

has all eigenvalues positive, so that it is positive definite [5]. But $-\bar{A}_z$ is just the matrix A_z for the interval matrix A_0^I; hence, A_0^I is positive definite by the assertion (c) of Theorem 2.

$(b) \Rightarrow (a)$: Let A_0^I be positive definite. Consider an eigenvalue λ of a matrix $A \in A^I$. Due to the Bendixson theorem ([15], p. 395), we have

$$\operatorname{Re}\lambda \leqq \lambda_{\max}(\tfrac{1}{2}(A + A^T)),$$

where the matrix $\tilde{A} = \frac{1}{2}(A + A^T)$ is symmetric and belongs to A^I; hence, $-\tilde{A} \in A_0^I$. Thus, $-\tilde{A}$ is positive definite so that all eigenvalues of $\tilde{A}$ are negative, which gives that $\operatorname{Re}\lambda \leqq \lambda_{\max}(\tilde{A}) < 0$. Hence, A is stable, and because it was chosen arbitrarily, A^I is also stable. □

There are several previous results relevant to the equivalence $(a) \Leftrightarrow (c)$. First, let us recall that a matrix $A \in A^I$ is called a vertex matrix of A^I if for each $i, j \in \{1, \ldots, n\}$, either $A_{ij} = (A_c - \Delta)_{ij}$ or $A_{ij} = (A_c + \Delta)_{ij}$ holds. Thus, there are exactly 2^{n^2} vertex matrices in the most disadvantageous case of $\Delta > 0$. Clearly, $\bar{A}_z$ is a vertex matrix for each $z \in Y$. The first attempt to use vertex matrices for characterizing stability was made by Białas [3], who proved that a general interval matrix A^I is stable if and only if all its vertex matrices are stable. His result was shown, however, to be erroneous by Karl, Greschak, and Verghese [7] and independently by Barmish and Hollot [1]. Soh [14] proved in 1990 that the conjecture is true for *symmetric* interval matrices in this form: if all the symmetric vertex matrices of A^I are stable, then each symmetric $A \in A^I$ is stable. This result required testing $2^{n(n+1)/2}$ vertex matrices for stability. This bound has been essentially improved recently by Hertz [6], who proved (using another notation) that if all the matrices $\bar{A}_z$ are stable, then each *symmetric* $A \in A^I$ is stable; this reduced the number of test matrices from $2^{n(n+1)/2}$ to 2^{n-1}. Theorem 6 shows that under the Hertz assumption *each* matrix $A \in A^I$ is already stable.

In Theorem 2 we showed that positive (semi)definiteness of a general interval matrix can be equivalently formulated in terms of the associated symmetric interval matrix A_s^I. Unfortunately, this nice property does not hold for stability, where only one implication is true.

THEOREM 7. *If A_s^I is stable, then A^I is also stable.*

Proof. Let λ be an eigenvalue of a matrix $A \in A^I$. Then by the Bendixson theorem we have $\operatorname{Re} \lambda \leqq \lambda_{\max}(\frac{1}{2}(A + A^T)) < 0$ because the symmetric matrix $\frac{1}{2}(A + A^T)$ belongs to A_s^I and thus has all eigenvalues negative. This proves that A^I is stable. □

The converse implication is generally not valid. Consider the interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ with

$$A_c = \begin{pmatrix} 1 & 7 \\ -1 & -2 \end{pmatrix}$$

and $\Delta = 0$. Here A^I is stable because A_c is stable ($\operatorname{Re} \lambda_1 = \operatorname{Re} \lambda_2 = -\frac{1}{2}$), but A_s^I is not because $\lambda_{\max}(A_c') = (\sqrt{45} - 1)/2 = 2.85\cdots$.

Finally, we give the respective versions of Theorems 3 and 4 for the case of stability. The reformulations are direct consequences of the equivalence $(a) \Leftrightarrow (b)$ of Theorem 6.

THEOREM 8. *A symmetric interval matrix A^I is stable if and only if it is regular and contains at least one stable symmetric matrix.*

Proof. The "only if" part follows from the fact that each stable matrix is nonsingular. Conversely, if A^I is regular and contains a stable symmetric matrix $\tilde{A}$, then $A_0^I = [-A_c - \Delta, -A_c + \Delta] = \{-A;\ A \in A^I\}$ is also regular and contains a positive definite matrix $-\tilde{A}$; hence, A_0^I is positive definite by Theorem 3 and A^I is stable by Theorem 6. □

The last result of this section follows from Theorem 4 applied to the interval matrix $[-A_c - \Delta, -A_c + \Delta]$ and its straightforward proof is omitted.

THEOREM 9. *Let $A^I = [A_c - \Delta, A_c + \Delta]$ be a symmetric interval matrix such that A_c is stable and*

$$\rho(|A_c^{-1}|\Delta) < 1$$

holds. Then A^I is stable.

For a practical verification, the results of this section can be used in the following way. Given an interval matrix A^I, first form the symmetric interval matrix A_s^I and test it for stability using Theorem 9. If the test is successful, then A^I is stable (Theorem 7).

This procedure will, however, fail whenever A^I is stable, whereas A_s^I is not, as, e.g., in the example following Theorem 7. In such a case another condition must be tried (cf. Mansour [8] for further results).

Example. Consider the interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ with

$$A_c = \begin{pmatrix} -1 & -1 & 2 \\ 3 & -2 & -5 \\ -2 & 1 & -5 \end{pmatrix}$$

and $\Delta_{ij} = 0.03$ for each i, j. Then for the associated symmetric interval matrix $A_s^I = [A_c' - \Delta', A_c' + \Delta']$, we have

$$A_c' = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -2 & -2 \\ 0 & -2 & -5 \end{pmatrix}$$

and $\Delta' = \Delta$. Because A_c' is stable and $\rho(|(A_c')^{-1}|\Delta') = 0.9 < 1$, Theorems 7 and 9 imply that A^I is stable.

4. Schur stability of interval matrices. A square matrix A is called Schur stable if $\rho(A) < 1$, i.e., if $|\lambda| < 1$ for each eigenvalue λ of A. We shall consider here Schur stability of symmetric matrices only to avoid complex eigenvalues that seemingly cannot be easily handled by the method used. Therefore, we shall say that a symmetric interval matrix A^I is Schur stable if each *symmetric* matrix $A \in A^I$ is Schur stable; hence, we do not take into account the nonsymmetric matrices contained in A^I. This definition is in accordance with the approach employed in [14] or [6].

A necessary and sufficient condition for Schur stability has been recently given by Hertz [6], who proved that a symmetric interval matrix A^I is Schur stable if and only if all the matrices $A_z, \bar{A}_z, z \in Y$ are Schur stable. In Theorem 11 below we formulate another necessary and sufficient condition based on the following result that links Schur stability to Hurwitz stability.

THEOREM 10. *A symmetric interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ is Schur stable if and only if the symmetric interval matrices*

$$[(A_c - I) - \Delta, (A_c - I) + \Delta] \tag{4}$$

and

$$[(-A_c - I) - \Delta, (-A_c - I) + \Delta] \tag{5}$$

are stable, where I is the unit matrix.

Proof. Only if: Denote the interval matrix (4) by $\tilde{A}^I$ and let $\tilde{A}_z = (A_c - I) + T_z \Delta T_z = \bar{A}_z - I$ for $z \in Y$. Because $\bar{A}_z$ is symmetric and Schur stable, it has all eigenvalues in $(-1, 1)$; therefore, all the eigenvalues of $\tilde{A}_z$ belong to $(-2, 0)$; hence, $\tilde{A}_z$ is stable. In view of Theorem 6 this implies that $\tilde{A}^I$ is stable. Stability of (5) can be proved in a similar way if we consider the matrices $\tilde{\tilde{A}}_z = -A_z - I$, $z \in Y$.

If: Let $A \in A^I$ be symmetric and let λ be an eigenvalue of A. Then $\lambda - 1$ is an eigenvalue of the matrix $A - I$ that belongs to (4) and hence is stable, which gives $\lambda - 1 < 0$. In a similar way, stability of (5) implies $-\lambda - 1 < 0$. Hence, $|\lambda| < 1$; thus, A^I is Schur stable. □

Now we have this criterion that is again formulated along the lines of Theorems 3 and 8.

THEOREM 11. *A symmetric interval matrix A^I is Schur stable if and only if it contains at least one Schur stable symmetric matrix and both the interval matrices* (4) *and* (5) *are regular.*

Proof. Only if: If A^I is Schur stable, then both (4) and (5) are stable by Theorem 10; hence, regular. If: Let some symmetric $A_0 \in A^I$ be Schur stable and let (4) and (5) be regular. Then $A_0 - I$ is symmetric, stable, and belongs to (4); hence, (4) is stable by Theorem 8. Similarly, stability of (5) can be established by considering the matrix $-A_0 - I$. Then Theorem 10 gives that A^I is Schur stable. □

Again, using sufficient regularity condition, we obtain the following.

THEOREM 12. *Let $A^I = [A_c - \Delta, A_c + \Delta]$ be a symmetric interval matrix such that A_c is Schur stable and the conditions*

$$\rho(|A_c - I|^{-1}\Delta) < 1 \tag{6}$$

$$\rho(|A_c + I|^{-1}\Delta) < 1 \tag{7}$$

are satisfied. Then A^I is Schur stable.

Proof. This is a direct consequence of Theorem 11 because (6) and (7) are the Beeck sufficient regularity conditions [2] for the interval matrices (4) and (5). □

5. Final remark. In Theorems 4, 9, and 12 we formulated verifiable sufficient conditions for positive definiteness, Hurwitz and Schur stability of symmetric interval matrices. Each of them involved the sufficient condition (2) for regularity of an interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$. This condition may be seen to be inappropriate for practical computations because the inverse matrix computed on a computer is usually afflicted with roundoff errors. Therefore, for practical purposes we propose a modified condition

$$\rho(|I - QA_c| + |Q|\Delta) < 1 \tag{8}$$

involving an arbitrary square matrix Q, because we have: *if* (8) *holds for some Q, then A^I is regular*. In fact, for an arbitrary $A \in A^I$, we have

$$QA = I - (I - QA_c + Q(A_c - A))$$

and because

$$\rho(I - QA_c + Q(A_c - A)) \leqq \rho(|I - QA_c| + |Q|\Delta) < 1,$$

it follows that QA is nonsingular; hence, A is nonsingular. Notice that (2) is a special case of (8) for $Q = A_c^{-1}$. In practical computations we recommend to set Q equal to the *computed* value of A_c^{-1}.

Acknowledgment. The author wishes to thank two anonymous referees for helpful suggestions.

REFERENCES

[1] B. R. BARMISH AND C. V. HOLLOT, *Counterexample to a recent result on the stability of interval matrices by S. Białas*, Internat. J. Control, 39 (1984), pp. 1103–1104.

[2] H. BEECK, *Zur Problematik der Hüllenbestimmung von Intervallgleichungssystemen*, in Interval Mathematics, K. Nickel, ed., Lecture Notes in Computer Science 29, Springer-Verlag, Berlin, 1975, pp. 150–159.

[3] S. BIAŁAS, *A necessary and sufficient condition for the stability of interval matrices*, Internat. J. Control, 37 (1983), pp. 717–722.

[4] S. BIAŁAS AND J. GARLOFF, *Intervals of P-matrices and related matrices*, Linear Algebra Appl., 58 (1984), pp. 33–41.

[5] M. FIEDLER, *Special Matrices and Their Use in Numerical Analysis*, SNTL Publishing House, Prague, 1986.

[6] D. HERTZ, *The extreme eigenvalues and stability of symmetric interval matrices*, IEEE Trans. Automat. Control, to appear.

[7] W. C. KARL, J. P. GRESCHAK, AND G. C. VERGHESE, *Comments on "A necessary and sufficient condition for the stability of interval matrices,"* Internat. J. Control, 39 (1984), pp. 849–851.
[8] M. MANSOUR, *Robust stability of interval matrices*, Proc. 28th Conf. Decision and Control, Tampa, FL, 1989, pp. 46–51.
[9] A. NEUMAIER, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.
[10] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[11] J. ROHN, *Inverse-positive interval matrices*, Z. Angew. Math. Mech., 67 (1987), pp. T492–T493.
[12] ———, *Systems of linear interval equations*, Linear Algebra Appl., 126 (1989), pp. 39–78.
[13] Z. C. SHI AND W. B. GAO, *A necessary and sufficient condition for the positive-definiteness of interval symmetric matrices*, Internat. J. Control, 43 (1986), pp. 325–328.
[14] C. B. SOH, *Necessary and sufficient conditions for stability of symmetric interval matrices*, Internat. J. Control, 51 (1990), pp. 243–248.
[15] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer-Verlag, Berlin, 1980.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 185–193, January 1994

015

ESPRIT DIRECTION-OF-ARRIVAL ESTIMATION IN THE PRESENCE OF SPATIALLY CORRELATED NOISE*

HAESUN PARK†

Abstract. An algorithm is presented for ESPRIT (estimation of signal parameters via rotational invariance techniques) direction of arrival estimation when the expected value of the covariance of the measurement noise is different from a constant multiple of an identity matrix. The algorithm is based on a modification of the generalized singular value decomposition (GSVD) of two data matrices and requires only unitary transformations. The modification results in a significant simplification of the GSVD-based ESPRIT algorithm and produces more accurate solutions than prewhitening.

Key words. correlated noise, direction of arrival, generalized singular value decomposition

AMS subject classifications. 15A18, 15A23

1. Introduction. ESPRIT is a recently developed method for signal parameter estimation with applications in array signal processing such as direction of arrival estimation [7], [8]. The method is based on a sensor array consisting of two identical subarrays separated by a known displacement vector. It provides estimates of the signal parameters by exploiting the eigenstructure of the underlying rotational invariance. For other applications where similar structures arise, see [8]. We briefly describe the array geometry and signal model assumed in ESPRIT and show how the algorithm is formulated. Since the goal of this paper is to develop an efficient and numerically robust algorithm to estimate the directions of arrival according to the ESPRIT method, we will state the problem in a matrix-oriented language. For the detailed description and assumptions on which the formulation is based, see [7] and [8]. Throughout this paper, the superscript asterisk (*) of a matrix denotes the complex conjugate transpose of the matrix and i denotes the pure imaginary number $\sqrt{-1}$.

The array considered in ESPRIT consists of m sensor pairs. The sensor array is grouped into two subarrays, X and Y, that are assumed to be displaced by a known translation vector. We denote the number of sources by d, which is unknown, and assume the relation that $m \geqq d$. Let $A \in C^{m \times d}$ be the unknown matrix with rank d, whose columns are the steering vectors associated with the sources, and let $\Phi = \operatorname{diag}(e^{i\gamma_1}, \ldots, e^{i\gamma_d}) \in C^{d \times d}$ be the unitary diagonal matrix that relates the measurements from subarray X to those from subarray Y. The sensor output $x(t)$, $y(t) \in C^{m \times 1}$ is modeled as

$$x(t) = As(t) + n_x(t), \qquad y(t) = A\Phi s(t) + n_y(t), \tag{1.1}$$

where $s(t) \in C^{d \times 1}$ is an unknown vector of impinging signals and $n_x(t)$, $n_y(t) \in C^{m \times 1}$ are unknown noise vectors. Combining the two equations in (1.1), we have

$$z(t) \equiv \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A \\ A\Phi \end{bmatrix} s(t) + n_z(t), \tag{1.2}$$

where

$$n_z(t) = \begin{bmatrix} n_x(t) \\ n_y(t) \end{bmatrix}.$$

* Received by the editors February 25, 1991; accepted for publication (in revised form) June 25, 1992. This work was supported in part by National Science Foundation grant CCR-8813493.

† Computer Science Department, University of Minnesota, Minneapolis, Minnesota 55455 (hpark@cs.umn.edu).

Assuming that the signals are uncorrelated with the noise and that the expected value of $s(t)s(t)^*$, $E(ss^*)$, is a positive definite covariance matrix R_{SS}, we have

$$E(zz^*) = \begin{bmatrix} A \\ A\Phi \end{bmatrix} R_{SS} \begin{bmatrix} A \\ A\Phi \end{bmatrix}^* + E(n_z n_z^*).$$

The problem is to find the unitary diagonal matrix Φ when we have the matrices $H_z \in C^{N_z \times 2m}$ and $H_n \in C^{N_n \times 2m}$ with full column rank that estimate $E(zz^*)$ and $E(n_z n_z^*)$, which is $E(zz^*) \approx H_z^* H_z$ and $E(n_z n_z^*) \approx \sigma^2 \Sigma = \sigma^2 H_n^* H_n$, with

$$H_z^* H_z \approx \begin{bmatrix} A \\ A\Phi \end{bmatrix} R_{SS} \begin{bmatrix} A \\ A\Phi \end{bmatrix}^* + \sigma^2 H_n^* H_n. \tag{1.3}$$

For notational convenience, we replace $\approx$ by $=$ in (1.3) in the following. Once the unitary diagonal matrix Φ is found, the directions of arrival can be derived from Φ by simple arithmetic. For details, see [7] and [8].

There have been many numerical algorithms developed for the ESPRIT direction of arrival methods [7], [8] where the noise is assumed to be white, i.e., the expected value of the covariance of the measurement noise has the form $E(n_z n_z^*) = \sigma^2 I$. In the more general case when $E(n_z n_z^*) = \sigma^2 \Sigma$, where $\Sigma \neq I$ is positive definite, prewhitening of the measurement noise using $\Sigma^{-1/2}$ avoids the generalized eigenvalue problem [8], [11]. However, when prewhitening is used, numerical difficulty can occur. To avoid the difficulty, a solution via a GSVD has been developed [13].

In this paper, we present an algorithm for the ESPRIT direction of arrival estimation when $E(n_z n_z^*) = \sigma^2 \Sigma$ where Σ is positive definite. The new algorithm produces numerically more accurate solutions than the prewhitening method and is more efficient than the existing GSVD-based method [13]. Our method is based on a special form of the GSVD, computation of which relies solely upon unitary transformations applied to the data matrices. The special form of the GSVD simplifies the computations in the ESPRIT by taking advantage of the structure of an intermediate matrix. Further advantages of the new form of the GSVD for ESPRIT will be discussed after the algorithm is introduced.

2. GSVD. The following theorem introduces the GSVD as was originally defined in Van Loan [12].

THEOREM 1. *Suppose that two matrices $H_z \in C^{N_z \times 2m}$ with $N_z \geqq 2m$ and $H_n \in C^{N_n \times 2m}$ are given. Then there exist unitary matrices $U_z \in C^{N_z \times N_z}$ and $U_n \in C^{N_n \times N_n}$ and a nonsingular matrix $X \in C^{2m \times 2m}$ such that*

$$\begin{aligned} U_z^* H_z X &= D_z = \operatorname{diag}(\alpha_1, \ldots, \alpha_{2m}) \in C^{N_z \times 2m}, \\ U_n^* H_n X &= D_n = \operatorname{diag}(\beta_1, \ldots, \beta_p) \in C^{N_n \times 2m}, \end{aligned} \tag{2.1}$$

where $p = \min(N_n, 2m)$, $\alpha_i \geqq 0$ for $1 \leqq i \leqq 2m$, and $\beta_i \geqq 0$ for $1 \leqq i \leqq p$. □

We will assume throughout this paper that $N_z \geqq 2m$ and $N_n \geqq 2m$, thus, $p = 2m$, since enough samples should be taken to provide good estimates for the directions of arrival. It is well known that a generalized eigenvalue problem for finding the scalars μ's that satisfy $\det(H_z^* H_z - \mu H_n^* H_n) = 0$ can be solved by computing the GSVD of H_z and H_n since

$$\det(H_z^* H_z - \mu H_n^* H_n) = \det(D_z^* D_z - \mu D_n^* D_n) \det(X^{-*} X^{-1}).$$

This gives better numerical solutions since the explicit formation of the products $H_z^* H_z$ and $H_n^* H_n$ is avoided.

Although the GSVD can produce numerically more accurate solutions, computing the GSVD is a difficult task when the matrices are ill conditioned [4]–[6]. If the nonsingular matrix X from the GSVD is required, then we can expect numerical difficulties when the data matrices H_z and H_n are ill conditioned. Paige and Saunders [5] suggest an alternative form of the GSVD that has better numerical properties. Paige [4] proposed an algorithm that computes this alternative form for the GSVD. We define yet another form of the GSVD in Theorem 2 upon which the new algorithm for the ESPRIT relies. The new GSVD of (2.2) is identical to the GSVD introduced by Paige and Saunders except that in [5], the matrix L is upper triangular. It will become clear that it is critical to have the lower triangular matrix L for simplifying the computation of the ESPRIT algorithm.

THEOREM 2. *Suppose that two matrices $H_z \in C^{N_z \times 2m}$ and $H_n \in C^{N_n \times 2m}$ with $N_z \geqq 2m$ and $N_n \geqq 2m$ are given. Then there exist unitary matrices $U_z \in C^{N_z \times N_z}$, $U_n \in C^{N_n \times N_n}$, and $Q \in C^{2m \times 2m}$, and lower triangular matrices $L_z \in C^{N_z \times 2m}$ and $L_n \in C^{N_n \times 2m}$, such that*

$$U_z^* H_z Q = L_z \quad \text{and} \quad U_n^* H_n Q = L_n, \tag{2.2}$$

where $L_z = D_z L$ and $L_n = D_n L$ for a nonsingular lower triangular matrix $L \in C^{2m \times 2m}$ and diagonal matrices $D_z = \operatorname{diag}(\alpha_1, \ldots, \alpha_{2m}) \in C^{N_z \times 2m}$ and $D_n = \operatorname{diag}(\beta_1, \ldots, \beta_{2m}) \in C^{N_n \times 2m}$ with $\alpha_i \geqq 0$ and $\beta_i \geqq 0$ for $1 \leqq i \leqq 2m$. Moreover, if each of H_z and H_n has full column rank, then the elements in D_z and D_n can be ordered so that

$$\frac{\alpha_1}{\beta_1} \geqq \frac{\alpha_2}{\beta_2} \geqq \cdots \geqq \frac{\alpha_{2m}}{\beta_{2m}} \tag{2.3}$$

is satisfied.

Proof. Let

$$U_z^* H_z X = D_z \quad \text{and} \quad U_n^* H_n X = D_n \tag{2.4}$$

be the GSVD of H_z and H_n according to Theorem 1. Consider the decomposition of X into a unitary matrix $Q \in C^{2m \times 2m}$ and a lower triangular matrix $L^{-1} \in C^{2m \times 2m}$: $X = QL^{-1}$. Then we have $U_z^* H_z Q = D_z L$ and $U_n^* H_n Q = D_n L$. The second part is obvious since the diagonal elements of D_z and D_n in the decomposition (2.4) can be ordered to satisfy the condition (2.3) via a permutation P, then the above process can be applied to XP. □

Although Theorem 2 introduces a modified form for the GSVD that gives a numerically more robust and efficient algorithm for ESPRIT, our algorithm to compute the GSVD of (2.2) would not follow the procedure given in the proof. A better way to compute the new GSVD is given in Algorithm GSVD_L in § 3, which is essentially the same as the algorithm due to Paige [4], except that a matrix W in each step is chosen to lower triangularize the 2×2 submatrices, whereas Paige upper triangularizes the submatrices.

THEOREM 3. *Suppose that $H_z \in C^{N_z \times 2m}$ and $H_n \in C^{N_n \times 2m}$ have full column rank and that they have the* GSVD *as in* (2.2) *and D_z and D_n are ordered to satisfy* (2.3). *Also, assume that $H_z^* H_z = \begin{bmatrix} A \\ A\Phi \end{bmatrix} R_{SS} \begin{bmatrix} A \\ A\Phi \end{bmatrix}^* + \sigma^2 H_n^* H_n$, where $A \in C^{m \times d}$ has rank d, $\Phi \in C^{d \times d}$ is unitary and diagonal, and R_{SS} is symmetric positive definite. Then*

$$\frac{\alpha_1}{\beta_1} \geqq \cdots \geqq \frac{\alpha_d}{\beta_d} \geqq \frac{\alpha_{d+1}}{\beta_{d+1}} = \cdots = \frac{\alpha_{2m}}{\beta_{2m}} = \sigma \tag{2.5}$$

and

$$\begin{bmatrix} A \\ A\Phi \end{bmatrix} R_{SS} \begin{bmatrix} A \\ A\Phi \end{bmatrix}^* = Q_1 L_{11}^* \Sigma L_{11} Q_1^*, \tag{2.6}$$

where $Q = [Q_1 \quad Q_2]$ *with* $Q_1 \in C^{2m \times d}$, $Q_2 \in C^{2m \times (2m-d)}$,

$$L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}$$

with $L_{11} \in C^{d \times d}$, *and* $\Sigma = \operatorname{diag}(\alpha_1^2 - \sigma^2\beta_1^2, \alpha_2^2 - \sigma^2\beta_2^2, \ldots, \alpha_d^2 - \sigma^2\beta_d^2) \in R^{d \times d}$.

Proof. For the proof of the first part that involves (2.5), see [13].

$$\begin{aligned} \begin{bmatrix} A \\ A\Phi \end{bmatrix} R_{SS} \begin{bmatrix} A \\ A\Phi \end{bmatrix}^* &= H_z^* H_z - \sigma^2 H_n^* H_n = QL^*(D_z^* D_z - \sigma^2 D_n^* D_n) L Q^* \\ &= [Q_1 \quad Q_2] \begin{bmatrix} L_{11}^* & L_{21}^* \\ 0 & L_{22}^* \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} \\ &= Q_1 L_{11}^* \Sigma L_{11} Q_1^*. \qquad \square \end{aligned}$$

Dividing the matrix $Q_1 \in C^{2m \times d}$ further into two parts

$$Q_1 = \begin{bmatrix} Q_x \\ Q_y \end{bmatrix},$$

where $Q_x \in C^{m \times d}$ and $Q_y \in C^{m \times d}$, we have

$$\begin{bmatrix} A \\ A\Phi \end{bmatrix} R_{SS} \begin{bmatrix} A \\ A\Phi \end{bmatrix}^* = \begin{bmatrix} Q_x \\ Q_y \end{bmatrix} L_{11}^* \Sigma L_{11} \begin{bmatrix} Q_x \\ Q_y \end{bmatrix}^*. \tag{2.7}$$

Accordingly,

$$AR_{SS}A^* - \lambda AR_{SS}\Phi^*A^* = AR_{SS}(I - \lambda\Phi^*)A^* = Q_x L_{11}^* \Sigma L_{11}(Q_x^* - \lambda Q_y^*),$$

and rank $(AR_{SS}(I - \lambda\Phi^*)A^*) < d$ if and only if λ is the same as one of the diagonal elements of Φ. That is, to find Φ, we have only to find $\lambda_1, \ldots, \lambda_d$ that make

$$\text{rank}\,(Q_x L_{11}^* \Sigma L_{11}(Q_x^* - \lambda Q_y^*)) < d, \quad \text{i.e., rank}\,(Q_x^* - \lambda Q_y^*) < d. \tag{2.8}$$

The importance of the lower triangular structure of the matrix L in the GSVD of (2.2) is clear from Theorem 3, since only the upper left $d \times d$ submatrix L_{11} of L is required in representing the matrix $[{}^{A}_{A\Phi}]\, R_{SS} [{}^{A}_{A\Phi}]^*$ as shown in (2.7). In fact, none of the elements from L is needed: the lower triangular structure of the matrix L resulted in requiring only Q_x and Q_y for finding the matrix Φ as shown in (2.8). The difficult problem that remains is that the eigenvalues of a rectangular matrix pencil $Q_x^* - \lambda Q_y^* \in C^{d \times m}$ must be computed. We can apply one of two previously developed approaches [8], [13] for reducing the problem to an eigenproblem of a square pencil. From the relation in (2.7),

$$\text{null}\,(Q_x^*) = \text{null}\,(Q_y^*) = \text{null}\,(A^*). \tag{2.9}$$

Let a matrix $J_1 \in C^{m \times (m-d)}$ satisfy span $(J_1) =$ null (Q_x^*) and be expanded to a nonsingular matrix $J = [J_1 \quad J_2] \in C^{m \times m}$. Then

$$\text{rank}\,(Q_x^* - \lambda Q_y^*) = \text{rank}\,((Q_x^* - \lambda Q_y^*)J) < d, \quad \text{i.e., rank}\,(Q_x^* J_2 - \lambda Q_y^* J_2) < d, \tag{2.10}$$

and the values λ are the eigenvalues of a $d \times d$ square pencil. The matrix J can be obtained from the QR decomposition of Q_x [13].

We can also reduce the rectangular matrix pencil into a square pencil by using the total least squares (TLS) technique [2] as in TLS-ESPRIT [8]. Since

$$\text{Range}\,(Q_x) = \text{Range}\,(Q_y) = \text{Range}\,(A) \tag{2.11}$$

from (2.7), there must be a unique nonsingular matrix $T \in C^{d\times d}$ such that

$$Q_x = Q_y T. \tag{2.12}$$

Since

$$\text{rank}\,(Q_x^* - \lambda Q_y^*) = \text{rank}\,(T^* Q_y^* - \lambda Q_y^*) < d, \quad \text{i.e., rank}\,(T^* - \lambda I) < d,$$

we have only to find the eigenvalues of T^*. In practice, neither of the relations (2.9) or (2.11) can hold exactly in finite precision. We estimate the matrix T using the TLS idea as in TLS-ESPRIT, assuming that Q_x and Q_y are equally noisy [2], [8]. The algorithm is summarized in § 3.

3. New algorithm. In general, the matrices $H_z \in C^{N_z\times 2m}$ and $H_n \in C^{N_n\times 2m}$ in ESPRIT satisfy the relations that $N_z \gg 2m$ and $N_n \gg 2m$. The initial factorization of each matrix into a unitary matrix and a lower triangular form (we will call it a QL factorization),

$$H_z = Q_z \begin{bmatrix} L_z \\ 0 \end{bmatrix} \quad \text{and} \quad H_n = Q_n \begin{bmatrix} L_n \\ 0 \end{bmatrix},$$

reduces the computational costs for the GSVD substantially. If the GSVD of L_z and L_n is

$$U_1^* L_z Q = L_1, \qquad U_2^* L_n Q = L_2, \tag{3.1}$$

where $L_1 = D_1 L$ and $L_2 = D_2 L$, then the GSVD of H_z and H_n can be obtained as

$$\left(Q_z \begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix}\right)^* H_z Q = \begin{bmatrix} L_1 \\ 0 \end{bmatrix}, \qquad \left(Q_n \begin{bmatrix} U_2 & 0 \\ 0 & I \end{bmatrix}\right)^* H_n Q = \begin{bmatrix} L_2 \\ 0 \end{bmatrix}. \tag{3.2}$$

Note that the same unitary matrix Q appears in both (3.1) and (3.2), which is the only unitary matrix we need from the decomposition for our ESPRIT algorithm. Thus the problem is reduced to computing the GSVD of two lower triangular matrices L_z, $L_n \in C^{2m\times 2m}$, and there is no need to compute Q_z, Q_n, U_1, or U_2. The Paige algorithm [4] for computing the GSVD of two real matrices is similar to the Jacobi algorithm for computing the SVD. If we start with a triangular matrix and use cyclic-by-rows ordering, the matrix will alternate between upper and lower triangular forms after each sweep. Since the lower triangular structure of L is required in our algorithm, we can finish the iterations after an even number of sweeps when we start with two lower triangular matrices L_z and L_n. For parallel implementation, we can also use the odd-even ordering that preserves the original triangular structure. For details, see [1] and [4].

The algorithm for computing our GSVD is summarized in Algorithm GSVD_L. For a matrix X, X_{pq} denotes the 2×2 submatrix of X:

$$\begin{bmatrix} x_{pp} & x_{pq} \\ x_{qp} & x_{qq} \end{bmatrix}.$$

Similarly, for a 2×2 matrix Y, $Y(p, q)$ denotes a matrix of proper dimension that is like an identity matrix, except that its 2×2 submatrix in the (p, q) plane is Y.

ALGORITHM GSVD_L ($H_z \in C^{N_z \times 2m}$, $H_n \in C^{N_n \times 2m}$)

0. $Q := I_{2m}$.
1. Compute the QL decomposition of H_z and H_n:

$$H_z = Q_z \begin{bmatrix} A \\ 0 \end{bmatrix} \quad \text{and} \quad H_n = Q_n \begin{bmatrix} B \\ 0 \end{bmatrix}.$$

2. Repeat until convergence.
3. Iterate for one sweep using odd-even ordering.
   /* assume that the current index pair is (p, q).
3.1. Determine 2×2 unitary matrices U, V, and J
   so that $U^* A_{pq} J$ and $V^* B_{pq} J$ are lower triangular and
   the second row of $U^* A_{pq} J$ is parallel to the second row of $V^* B_{pq} J$.
3.2. {update the matrices}
   $A := U^*(p, q) A J(p, q)$,
   $B := V^*(p, q) B J(p, q)$,
   $Q := Q J(p, q)$.

The algorithm is terminated when each row of A is parallel to the corresponding row of B. Under the assumption that H_z and H_n have full column rank, the computed generalized singular values α_i / β_i can be obtained from a_{ii} / b_{ii}. For our application, none of the left unitary transformations need to be saved. Thus, the input matrices $H_z \in C^{N_z \times 2m}$ and $H_n \in C^{N_n \times 2m}$ can be overwritten by the lower triangular matrices $L_z \in C^{2m \times 2m}$ and $L_n \in C^{2m \times 2m}$. The discussions from § 2 are summarized in the following algorithm, using the TLS approximation.

ALGORITHM ESPRIT_GSVD_L

1. Form the matrices H_z and H_n from the available measurements.
2. Compute the GSVD of H_z and H_n

$$U_z^* H_z Q = L_z \quad \text{and} \quad U_n^* H_n Q = L_n$$

   according to Algorithm GSVD_L.
3. Estimate the number of sources d from L_z and L_n.
   Let

$$\begin{bmatrix} Q_x \\ Q_y \end{bmatrix}$$

   be the first d columns of Q.
4. Compute the SVD of

$$[Q_x \quad Q_y] = WD \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}^*.$$

5. Compute the eigenvalues of $-V_{12} V_{22}^{-1}$.

Two existing algorithms, one based on the GSVD of form (2.2) due to Van Loan and the other based on the TLS-ESPRIT with prewhitening, are summarized in algorithms ESPRIT_GSVD and ESPRIT_PREWHITENING, respectively. For the definition of the CS decomposition used in Algorithm ESPRIT_GSVD, see Golub and Van Loan [3].

ALGORITHM ESPRIT_GSVD

1. Form the matrices H_z and H_n from the available measurements.
2. Compute the QR decomposition

$$\begin{bmatrix} H_z \\ H_n \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R,$$

   where $Q_1 \in C^{N_z \times 2m}$, $Q_2 \in C^{N_n \times 2m}$, and $R \in C^{2m \times 2m}$.
3. Compute the CS decomposition of Q_1 and Q_2:

$$U_1^* Q_1 Q = C \quad \text{and} \quad U_2^* Q_2 Q = S.$$

4. Estimate the number of sources d from C and S.
   Let

$$\begin{bmatrix} Q_x \\ Q_y \end{bmatrix}$$

   be the first d columns of R^*Q.
5. Compute the SVD of

$$[Q_x \quad Q_y] = WD \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}^*.$$

6. Compute the eigenvalues of $-V_{12}V_{22}^{-1}$.

Note that the matrices Q_1, Q_2, and R from step 2 need to be saved for steps 3 and 4. Thus the storage requirement of Algorithm ESPRIT_GSVD is larger than that of Algorithm ESPRIT_GSVD_L. Assuming that the complexity for computing the GSVD in ESPRIT_GSVD_L and the complexity for the CS decomposition in ESPRIT_GSVD are about the same, ESPRIT_GSVD is more costly and the control is more complicated than that of ESPRIT_GSVD_L since Q_1, Q_2, and R^*Q should be computed.

ALGORITHM ESPRIT_PREWHITENING

1. Form the matrices H_z and H_n from the available measurements.
   Measure Σ such that $E(n_z n_z^*) = \sigma^2 \Sigma$ (or assume that Σ is known).
2. Compute $\Sigma^{-1/2}$.
3. Compute the SVD of $H_z\Sigma^{-1/2}$: $H_z\Sigma^{-1/2} = UDQ^*$.
4. Estimate the number of sources d from D.
   Let

$$\begin{bmatrix} Q_x \\ Q_y \end{bmatrix}$$

   be the first d columns of $\Sigma^{1/2}{}^*Q$.
5. Compute the SVD of

$$[Q_x \quad Q_y] = WD \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}^*.$$

6. Compute the eigenvalues of $-V_{12}V_{22}^{-1}$.

All three algorithms were coded in MATLAB on a SUN 3/80 with the machine precision $\varepsilon = 2.2204E - 16$. Both algorithms ESPRIT_GSVD_L and ESPRIT_GSVD

produced solutions that are essentially the same in accuracy. When the condition number of the noise matrix H_n is large, prewhitening suffered from numerical difficulties. For example, when the condition number of $H_n \approx 1/\sqrt{\varepsilon}$, with the directions of arrival 14° and 15°, and signal-to-noise ratio 10, while two algorithms based on the GSVD produced answers accurate to the ninth digit, prewhitening failed (the eigenvalues produced in step 6 have absolute values far from unit or the algorithm broke down without completion) 25 times out of 30 tests.

Remarks. We have presented a new algorithm for the ESPRIT direction of arrival estimation in the presence of noise of which the expected value of the covariance is $\sigma^2\Sigma$, where Σ is not necessarily an identity matrix. The new algorithm has been shown to be more efficient than the existing GSVD-based algorithm in terms of storage and complexity. Also, it produces more accurate solutions than the prewhitening method. The fact that only the unitary matrix Q is needed would make the systolic array for the new algorithm simple. The implementation of the new algorithm on a two-dimensional mesh-connected array should be straightforward from the results presented in [6].

One difficulty in the parallel implementation of the proposed algorithm is that the generalized singular values need to be ordered in decreasing order. We have modified the Paige algorithm so that the generalized singular values of the 2×2 submatrices in each step are ordered. More research is needed regarding the relation between the ordering of the singular values and the convergence.

Recently, Stewart [9], [10] introduced new decompositions of a matrix based on two-sided unitary transformations, and they are called the URV and ULV decompositions. These can be generalized for the triangularization of a matrix pair analogous to the GSVD. The generalized URV and ULV decompositions have many potential applications, especially where the solutions should adapt to changing statistics. Note that (2.2) can be used as a starting point for such generalization of the URV and ULV decompositions. Rank estimation and recursive updating will need to be incorporated to fully realize the generalized URV and generalized ULV decompositions. The advantage of this approach is that the solution can be recursively updated in an efficient way, which will lead to an adaptive ESPRIT algorithm.

Acknowledgments. The author wishes to thank Professor M. Kaveh and Mr. R. Hamza for a program to generate simulated data for the direction of arrival estimation. She also thanks Mr. B. Drake for introducing her to the problem discussed in this paper.

REFERENCES

[1] P. J. EBERLEIN AND H. PARK, *Efficient implementation of Jacobi algorithms and Jacobi sets on distributed memory architectures*, J. Parallel Dist. Comput., special issue on Algorithms for Hypercube Computers, 8 (1990), pp. 358–366.

[2] 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.

[3] ———, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[4] C. C. PAIGE, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1126–1146.

[5] C. C. PAIGE AND M. A. SAUNDERS, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.

[6] H. PARK AND L. M. EWERBRING, *An Algorithm for the Generalized Singular Value Decomposition on Massively Parallel Computers*, J. Parallel Dist. Comput., 17 (1993), pp. 267–276.

[7] R. ROY, A. PAULRAY, AND T. KAILATH, *ESPRIT-A subspace rotation approach to estimation of parameter of sinusoids in noise*, IEEE Trans. Acoustic Speech Signal Process, 34 (1986), pp. 1340–1342.

[8] R. ROY AND T. KAILATH, *ESPRIT-Estimation of signal parameters via rotational invariance techniques*, IEEE Trans. Acoustic Speech Signal Process, 37 (1989), pp. 984–995.

[9] G. W. STEWART, *An Updating Algorithm for Subspace Tracking*, IEEE Trans. Signal Proc., 40 (1992), pp. 1535–1541.

[10] ———, *Updating a Rank-Revealing ULV Decomposition*, Tech. Report, Dept. of Computer Science, Univ. of Maryland, College Park, CS-TR-2627, 1991.

[11] H. L. VAN TREES, *Detection, Estimation, and Modulation Theory*, John Wiley and Sons, New York, 1971.

[12] C. F. VAN LOAN, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.

[13] ———, *A Unitary Method for the ESPRIT Direction-of-Arrival Estimation Algorithm*, Tech. Report 87-862, Dept. of Computer Science, Cornell University, Ithaca, NY, 1987.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 194–204, January 1994

016

FINDING THE BEST REGRESSION SUBSET BY REDUCTION IN NONFULL-RANK CASES*

ALAN H. FEIVESON†

Abstract. The computational problem of finding the best fitting subset of independent variables in least-squares regression with a fixed subset size is addressed, especially in the context of the nonfull-rank case with more variables than observations. For the full-rank case, the most efficient widely used methods work by finding the complementary subset with minimum reduction to the total regression sum of squares; a task that can usually be accomplished with far less computation than exhaustive evaluation of all subsets. Here, a method using Cholesky-type factorizations (Algorithm 2) has been developed, which also takes advantage of the computational savings offered by the "reduction" approach, but which can be used in nonfull-rank cases where existing methods are not applicable. Algorithm 2 is derived by examining the asymptotic properties of a full-rank procedure (Algorithm 1) used on a "ridge" perturbation of the cross-product matrix. In the course of testing, it was discovered that Algorithm 1, with the appropriate ridge parameter, usually selected the best subset with less computation than Algorithm 2; however, if one requires mathematical certitude, use of Algorithm 2 is indicated. Also, some new approaches are proposed for developing efficient methods of identifying the best subset directly, rather than by complement to the minimum-reduction subset.

Key words. regression, subset selection, ridge regression

AMS subject classifications. 62J05, 05A05, 65U05, 62J07

1. Introduction.

Consider the standard linear regression model form

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e}, \tag{1.1}$$

where $\mathbf{y}$ is an $n \times 1$ vector of observations, $\mathbf{X}$ is an $n \times p$ matrix of n given values of p independent variables $\{\mathbf{x}_j;\ j = 1, \ldots, p\}$, β is a $p \times 1$ vector of unknown coefficients, and $\mathbf{e}$ is an error term with unspecified distribution. In the regression subset selection problem, given an integer $k < p$, we wish to find the subset S^* that maximizes $R(S, k)$, the regression sum of squares obtained by least-squares fitting of y to the k independent variables $\{\mathbf{x}_j | j \in S\}$, where S is a k-subset of the integers $1, 2, \ldots, p$.

In this paper, we are concerned only with the computational problem of finding S^*. Reasons for subset selection are discussed extensively in the literature, many involving minimizing estimation or prediction errors associated with a "true" underlying model, e.g., [3], [8], and [5]. In the estimation/prediction scenario, in addition to $R(S, k)$, the stability of the k-variate estimate of β should also be considered; cf. [2, p. 414]. Simulation results given by J. Hoerl, Schuenemeyer, and A. Hoerl [7] showed a tendency for subset models to be inferior to ridge regression for purposes of estimating β. A more efficacious application for subset selection by maximization of $R(S, k)$ is in the area of empirical modeling, where we observe $\mathbf{y}$ and $\mathbf{x}_1, \ldots, \mathbf{x}_{p^*}$, where $p^* < p$. No exact physical model of how $\mathbf{y}$ relates to $\mathbf{x}_1, \ldots, \mathbf{x}_{p^*}$ is known, but the goal is to interpolate the response between the experimental values of $\mathbf{x}_1, \ldots, \mathbf{x}_{p^*}$. This is done by generating $p - p^*$ *new* independent variables as functions of the originals (e.g., powers, ratios, etc.), with the intent of approximately representing the unknown response function by a linear combination of $\mathbf{x}_1, \ldots, \mathbf{x}_p$. No pretense is made about an unbiased estimation of β, since no actual underlying model of the form (1.1) is assumed. Indeed, if $\mathbf{y}$ is measured with no error, $\mathbf{e}$ may well be nonrandom, representing only model misspecification. The

* Received by the editors October 28, 1991; accepted for publication (in revised form) June 23, 1992.

† National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Houston, Texas 77058 (feiveson@astro.jsc.nasa.gov).

interpolating function should be reasonable in that (1) it fits the experimental data, and (2) it does not have excess "wiggles." The latter requirement can be generally satisfied by including only a limited number of terms (i.e., k) in the interpolating model; hence the selection problem.

Let $\mathbf{d} = \mathbf{X}'\mathbf{y}$ and $\mathbf{Q} = \mathbf{X}'\mathbf{X}$. For S a k-subset of $\{1, \ldots, p\}$, let $\mathbf{Q}_S$ be the principal submatrix (PSM) of $\mathbf{Q}$ corresponding to S. We shall assume $\mathbf{Q}_S$ to be nonsingular although $\mathbf{Q}$ may not be. Then $R(S, k) = \mathbf{d}_S'\mathbf{Q}_S^{-1}\mathbf{d}_S$, where $\mathbf{d}_S$ is the $k \times 1$ subvector of $\mathbf{d}$ also corresponding to S. Clearly, for large p and most values of k, it is not practical, especially on personal computers, to compute all $\binom{p}{k}$ values of $R(S, k)$ to find the maximum. Let $RTOT$ be the regression sum of squares on all p variables. We can ease the computational problem by searching for the complementary subset $\bar{S}$ of size $p - k$, which minimizes $L(\bar{S}, p - k) = RTOT - R(S, k)$, the reduction in the regression sum of squares due to removing $\{\mathbf{x}_j \mid j \in \bar{S}\}$. It has been shown, e.g., [4], that by solving this dual problem, we can avoid evaluating all subsets, because if a complementary subset of size less than $p - k$ exhibits a larger reduction than the candidate for the minimum, all subsets containing the smaller one will produce at least as large a reduction and hence will be ineligible to improve on the current candidate. In the remainder of this paper, this fact will be referred to as the monotone-reduction property.

When $\mathbf{Q}$ has full rank, it is well known that

$$L(\bar{S}, p - k) = \mathbf{b}_{\bar{S}}'\mathbf{C}_{\bar{S}}^{-1}\mathbf{b}_{\bar{S}}, \tag{1.2}$$

where $\mathbf{b}_{\bar{S}}$ and $\mathbf{C}_{\bar{S}}$ are the respective $(p - k) \times 1$ subvector and $(p - k) \times (p - k)$ submatrix of $\mathbf{b} = \mathbf{Q}^{-1}\mathbf{d}$ and $\mathbf{C} = \mathbf{Q}^{-1}$, corresponding to the elements of $\bar{S}$. Methods for computing (1.2) through sequential modification of $\mathbf{C}_{\bar{S}}^{-1}$ have further reduced computation. The algorithm of Furnival and Wilson [1], which is more general in that it finds the best regressions for all values of k, is based on the above reasoning, and is perhaps the most widely used procedure for subset selection today.

In the empirical-modeling problem there is no limit to the number of transformations of the original independent variables that can be made to define new ones. As a consequence, p could exceed n, making $\mathbf{Q}$ singular and the reduction as given by (1.2) undefined. In this paper, a procedure (Algorithm 2) is developed for comparing $L(\bar{S}, p - k)$ to a specified value that lends itself to sequential modification with changes in $\bar{S}$ when $\mathbf{Q}$ is singular. This allows relatively efficient identification of the best subset without evaluation of all values of $L(\bar{S}, p - k)$ or $R(S, k)$. The method is derived by examining the asymptotic properties of a full-rank procedure (Algorithm 1) used with ridge regression.

2. Ridge selection. In the singular case, $L(\bar{S}, p - k)$ cannot be evaluated by (1.2); however, borrowing from Hoerl and Kennard [6], we may replace $\mathbf{Q}$ with a *ridge* adjustment $\mathbf{Q} + \varepsilon\mathbf{M}$, where $\mathbf{M}$ is symmetric and otherwise arbitrary as long as $\mathbf{Q}(\varepsilon) = \mathbf{Q} + \varepsilon\mathbf{M}$ is positive definite for all $\varepsilon > 0$. We shall refer to $\mathbf{M}$ as the *ridge matrix*. For $\varepsilon > 0$, let $\mathbf{C}(\varepsilon) = \mathbf{Q}^{-1}(\varepsilon)$ and $\mathbf{b}(\varepsilon) = \mathbf{Q}^{-1}(\varepsilon)\mathbf{d}$. Define $L(\bar{S}, p - k; \mathbf{M}, \varepsilon)$ to be (1.2) with $\mathbf{C}_{\bar{S}}(\varepsilon)$ and $\mathbf{b}_{\bar{S}}(\varepsilon)$, the corresponding submatrix of $\mathbf{C}(\varepsilon)$ and subvector of $\mathbf{b}(\varepsilon)$ replacing $\mathbf{C}_{\bar{S}}$ and $\mathbf{b}_{\bar{S}}$, respectively. We use the term *ridge selection* to mean finding the subset that minimizes $L(\bar{S}, p - k; \mathbf{M}, \varepsilon)$.

Let q be the rank of $\mathbf{Q}$. Then there exists a $p \times q$ matrix $\mathbf{V}$ such that $\mathbf{V}'\mathbf{V} = \mathbf{I}_q$ and $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'$, where $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_q)$, λ_j being the jth positive eigenvalue of $\mathbf{Q}$. Let $\mathbf{H} = \mathbf{I}_p - \mathbf{V}\mathbf{V}'$. Then for $\varepsilon > 0$, $\mathbf{Q} + \varepsilon\mathbf{H}$ has eigenvalues $\lambda_1, \ldots, \lambda_q, \varepsilon, \ldots, \varepsilon$ and is thus positive definite. It will be shown by Theorem 2 that for ε sufficiently small, the subset chosen by ridge selection with $\mathbf{M} = \mathbf{H}$ is S^*. To prove Theorem 2, we first need Theorem 1.

THEOREM 1. *Given* $\mathbf{H}$ *defined as above,* $\lim_{\varepsilon \to 0+} L(\bar{S}, p-k; \mathbf{H}, \varepsilon) = L(\bar{S}, p-k)$ for all $(p-k)$-subsets $\bar{S} \subset \{1, 2, \ldots, p\}$.

Proof. For $\varepsilon > 0$, the relationship between regression and reduction sums of squares can be expressed by

$$\mathbf{d}'(\mathbf{Q} + \varepsilon\mathbf{H})^{-1}\mathbf{d} = \mathbf{d}'_{\bar{S}}(\mathbf{Q}_{\bar{S}} + \varepsilon\mathbf{H})^{-1}\mathbf{d}_{\bar{S}} + \mathbf{b}'_{\bar{S}}(\varepsilon)[\mathbf{C}_{\bar{S}}(\varepsilon)]^{-1}\mathbf{b}_{\bar{S}}(\varepsilon). \tag{2.1}$$

Denoting the terms in (2.1) by $a_1(\varepsilon)$, $a_2(\varepsilon)$, and $a_3(\varepsilon)$, respectively, it will be shown that $a_1(\varepsilon) = RTOT$ and that $\lim_{\varepsilon \to 0+} a_2(\varepsilon) = R(S, k)$. It follows that $\lim_{\varepsilon \to 0+} a_3(\varepsilon)$ must be equal to $RTOT - R(S, k) = L(\bar{S}, p-k)$.

Value of $a_1(\varepsilon)$. Recall $\mathbf{Q} = \mathbf{V}\Lambda\mathbf{V}'$ and $\mathbf{H} = \mathbf{I}_p - \mathbf{V}\mathbf{V}'$. The matrix $\mathbf{H}$ is symmetric, idempotent and satisfies $\mathbf{V}'\mathbf{H} = \mathbf{0}$; hence it is easily verified that

$$\mathbf{Q}^{-1}(\varepsilon) = [\mathbf{Q} + \varepsilon\mathbf{H}]^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{V}' + \left(\frac{1}{\varepsilon}\right)\mathbf{H}. \tag{2.2}$$

Since $\mathbf{d} = \mathbf{X}'\mathbf{y}$, there exists a $p \times 1$ vector α such that $\mathbf{d} = \mathbf{X}'\mathbf{X}\alpha = \mathbf{V}\Lambda\mathbf{V}'\alpha$. ($\alpha$ is a solution to the normal equations; e.g., see Searle [9].) Using $\mathbf{V}'\mathbf{V} = \mathbf{I}_q$, $\mathbf{V}'\mathbf{H} = \mathbf{0}$, and (2.2) we obtain $\mathbf{d}'\mathbf{Q}^{-1}(\varepsilon)\mathbf{d} = \alpha'\mathbf{V}\Lambda\mathbf{V}'\alpha = \alpha'\mathbf{Q}\alpha$, the total regression sum of squares, or $RTOT$, which does not depend on ε and is unique even though α is not.

Limit of $a_2(\varepsilon)$. Since $\mathbf{Q}_{\bar{S}}$ is positive definite, $\lim_{\varepsilon \to 0+} \mathbf{d}'_{\bar{S}}(\mathbf{Q}_{\bar{S}} + \varepsilon\mathbf{H})^{-1}\mathbf{d}_{\bar{S}} = \mathbf{d}'_{\bar{S}}\mathbf{Q}_{\bar{S}}^{-1}\mathbf{d}_{\bar{S}} = R(S, k)$. □

Using Theorem 1, we can now prove Theorem 2, which states that ridge selection with $\mathbf{M} = \mathbf{H}$ is asymptotically correct.

THEOREM 2. *Let* $\bar{S}_\varepsilon$ *be the subset that minimizes* $L(\bar{S}, p-k; \mathbf{H}, \varepsilon)$. *Then for* ε *sufficiently small,* $\bar{S}_\varepsilon$ *minimizes* $L(\bar{S}, p-k)$.

Proof. Let $\bar{S}^*$ be the $(p-k)$-subset that minimizes $L(\bar{S}, p-k)$, and let $\delta = L(\bar{S}^*, p-k) - L(\bar{S}', p-k)$, where $L(\bar{S}', p-k)$ is the second-lowest reduction sum of squares. From Theorem 1, $\varepsilon > 0$ can be found such that $|L(\bar{S}, p-k; \mathbf{H}, \varepsilon) - L(\bar{S}, p-k)| < \frac{\delta}{3}$ for all $(p-k)$-subsets $\bar{S} \subset \{1, 2, \ldots, p\}$. Then

$$\begin{aligned}
L(\bar{S}_\varepsilon, p-k) &< L(\bar{S}_\varepsilon, p-k; \mathbf{H}, \varepsilon) + \frac{\delta}{3} \text{ (from Theorem 1)} \\
&\leqq L(\bar{S}^*, p-k; \mathbf{H}, \varepsilon) + \frac{\delta}{3} \text{ (by definition of } \bar{S}_\varepsilon) \\
&< \left[L(\bar{S}^*, p-k) + \frac{\delta}{3}\right] + \frac{\delta}{3} \text{ (from Theorem 1)} \\
&= L(\bar{S}^*, p-k) + \frac{2\delta}{3} = L(\bar{S}', p-k) - \frac{\delta}{3}.
\end{aligned}$$

Thus $\bar{S}_\varepsilon$ must be $\bar{S}^*$. □

Remark. If the ridge matrix is the $p \times p$ identity, it can be shown that Theorem 1 and hence Theorem 2 with $\mathbf{I}$ replacing $\mathbf{H}$ still hold. In this case, $[\mathbf{Q} + \varepsilon\mathbf{I}_p]^{-1} = \mathbf{V}(\Lambda + \varepsilon\mathbf{I}_q)^{-1}\mathbf{V}' + (\frac{1}{\varepsilon})\mathbf{H}$; hence $a_1(\varepsilon) = \alpha'\mathbf{V}\Lambda(\Lambda + \varepsilon\mathbf{I}_q)^{-1}\Lambda\mathbf{V}'\alpha$, which has the limit $\alpha'\mathbf{V}\Lambda\mathbf{V}'\alpha = RTOT$ as $\varepsilon \to 0$. It follows that $\lim_{\varepsilon \to 0+} a_3(\varepsilon)$ must still be equal to $L(\bar{S}, p-k)$.

Although ridge selection with $\mathbf{M} = \mathbf{H}$ is used later as a device to derive the main result (Algorithm 2), it was found that using $\mathbf{M} = \mathbf{I}$ for appropriate values of ε gives surprisingly good results by itself. Ridge selection with $\mathbf{M} = \mathbf{I}$ can be easily implemented by appending $\varepsilon^{1/2}$ times the $p \times p$ identity matrix as additional rows of $\mathbf{X}$ and p zeros as

additional elements of $\mathbf{y}$, then using conventional software to find the best subset. The degree of success of this subterfuge depends on how well calculations stand up on the ill-conditioned matrix $\mathbf{Q} + \varepsilon\mathbf{I}$. This type of ridge selection was tested using Algorithm 1 in § 3 on several data sets with more variables than observations (see § 7). With $\mathbf{X}'\mathbf{X}$ in correlation form, it was found that ε on the order of 10^{-4} worked well in that the best subset was usually identified.

3. Comparing $L(\bar{S}, p - k)$ to a specified value. Despite the knowledge that for "small" ε, ridge selection theoretically identifies the best subset, we can never be assured in practice that ε was small enough or that numerical accuracy problems caused by small values of ε did not give spurious results. We therefore seek an exact method that does not suffer these shortcomings.

For the moment, let us revert to the full-rank case. We construct a procedure (Algorithm 1) that finds the best subset by comparing $L(\bar{S}, p - k)$ with trial minimum values. In § 4, this procedure is applied to $\mathbf{Q}(\varepsilon)$ with ridge matrix $\mathbf{H}$ and the limiting form as $\varepsilon \to 0$ is derived. Because of Theorem 2, we are assured that the limiting form of the procedure will find the best subset. To develop Algorithm 1 we need Theorem 3.

THEOREM 3. *For $L^* > 0$, let $\mathbf{D} = \mathbf{C} - (1/L^*)\mathbf{b}\mathbf{b}'$, where $\mathbf{C} = \mathbf{Q}^{-1}$ and $\mathbf{b} = \mathbf{Q}^{-1}\mathbf{d}$ as before. Let $\mathbf{D}_{\bar{S}}$* be the PSM of $\mathbf{D}$ *corresponding to the elements of a reduction subset $\bar{S}$ of size $m = p - k$. Then $L(\bar{S}, m) < L^*$ if and only if $\mathbf{D}_{\bar{S}}$* is positive definite.

Proof.

$$
\begin{aligned}
&\mathbf{D}_{\bar{S}} = \mathbf{C}_{\bar{S}} - \left(\frac{1}{L^*}\right)\mathbf{b}_{\bar{S}}\mathbf{b}'_{\bar{S}}, \\
\Rightarrow\ &|\mathbf{D}_{\bar{S}}| = |\mathbf{C}_{\bar{S}}|\left|I_m - \left(\frac{1}{L^*}\right)\mathbf{C}_{\bar{S}}^{-1}\mathbf{b}_{\bar{S}}\mathbf{b}'_{\bar{S}}\right|, \\
&\qquad = |\mathbf{C}_{\bar{S}}|\left[1 - \left(\frac{1}{L^*}\right)\mathbf{b}'_{\bar{S}}\mathbf{C}_{\bar{S}}^{-1}\mathbf{b}_{\bar{S}}\right], \\
&\qquad = |\mathbf{C}_{\bar{S}}|\left[1 - \frac{L(\bar{S}, m)}{L^*}\right].
\end{aligned}
\tag{3.1}
$$

The matrix $\mathbf{D}_{\bar{S}}$ is a rank-one perturbation of the positive-definite matrix $\mathbf{C}_{\bar{S}}$. It can thus be shown (e.g., see [2, p. 270]) that $\mathbf{D}_{\bar{S}}$ has at most one negative eigenvalue (of multiplicity not exceeding one). It follows that $|\mathbf{D}_{\bar{S}}| > 0$ if and only if $\mathbf{D}_{\bar{S}}$ is positive definite. Since $|\mathbf{C}_{\bar{S}}| > 0$, it can be seen from (3.1) that $L(\bar{S}, m) < L^*$ if and only if $\mathbf{D}_{\bar{S}}$ is positive definite. □

Theorem 3 suggests the following algorithm to find the best subset.

ALGORITHM 1.

Step 1. Number the m-subsets lexicographically: $\bar{S}_1 = \{1, 2, \ldots, m\}$, $\bar{S}_2 = \{1, 2, \ldots, m-1, m+1\}, \ldots, \bar{S}_M = \{p - m + 1, \ldots, p\}$, where $M = \binom{p}{m}$.

Step 2. Find an initial candidate $\bar{S}^*$ for the best subset, say, by stepwise regression. Set $i = 1$.

Step 3. Let $L^* = L(\bar{S}^*, m)$ and $\mathbf{D} = \mathbf{C} - (1/L^*)\mathbf{b}\mathbf{b}'$.

Step 4. Check $\mathbf{D}_{\bar{S}_i}$ for positive definiteness by attempting the Cholesky factorization $\mathbf{D}_{\bar{S}_i} = \mathbf{B}_{\bar{S}_i}(\mathbf{B}_{\bar{S}_i})'$, where $\mathbf{B}_{\bar{S}_i}$ is lower triangular. If the factorization fails, increment i; if $i = M + 1$, go to Step 6, otherwise repeat Step 4. If the factorization succeeds, go to Step 5.

Step 5. $\mathbf{D}_{\bar{S}_i}$ is positive definite; hence $L(\bar{S}_i, m) < L(\bar{S}^*, m)$. Set $\bar{S}^* = \bar{S}_i$, increment i by 1. If $i = M + 1$, go to Step 6, otherwise go back to Step 3.

Step 6. Stop. The best subset is $\bar{S}^*$.

Algorithm 1 can be made to take advantage of the monotone-reduction property by proper incrementation of i in Step 4. Suppose that $\bar{S}_i = \{s_1, s_2, \ldots, s_m\}$ and that factorization failed on the fth row of $\mathbf{D}_{\bar{S}_i}$. Then any subset containing $\{s_1, s_2, \ldots, s_f\}$ will also have a higher reduction than $L(\bar{S}^*, m)$ and need not be checked. Therefore, in Step 4, we should increment i by c, where $\bar{S}_{i+c}$ is the first reduction subset after $\bar{S}_i$ that does not contain $\{s_1, s_2, \ldots, s_f\}$. Suppose that the first g members of $\bar{S}_{i+c}$ are identical to the first g members of $\bar{S}_i$. Then the first g rows of $\mathbf{B}_{\bar{S}_{i+c}}$ are the same as the first g rows of $\mathbf{B}_{\bar{S}_i}$ and need not be recomputed.

4. The singular case: Algorithm 2. If $\mathbf{Q}$ is not of full rank, we can accept the possibility of error and implement ridge selection; e.g., with $\mathbf{M} = \mathbf{I}$ for some $\varepsilon > 0$. As an alternative, we consider the behavior of ridge selection when the ridge matrix is $\mathbf{H}$ and $\varepsilon \to 0$. By Theorem 2, for ε sufficiently small (say, $\varepsilon < \varepsilon_1$), this ridge selection will result in the proper identification of S^*. Let $\mathbf{Q}(\varepsilon) = \mathbf{Q} + \varepsilon\mathbf{H}$, and let $\mathbf{C}(\varepsilon)$, $\mathbf{b}(\varepsilon)$, and $L(\bar{S}, m; \mathbf{H}, \varepsilon)$ be defined as in § 2, where $\bar{S}$ is a reduction subset of size $m = p - k$. For $L^* > 0$, let $\mathbf{D}(\varepsilon) = \mathbf{C}(\varepsilon) - (1/L^*)\mathbf{b}(\varepsilon)\mathbf{b}'(\varepsilon)$, and let $\mathbf{D}_{\bar{S}}(\varepsilon)$ be the PSM of $\mathbf{D}(\varepsilon)$ corresponding to $\bar{S}$. It will be shown (Theorem 4) that there exists $\varepsilon_2 > 0$ and a matrix $\mathbf{B}$ that depends on $\bar{S}$ but not on ε such that all matrices $\mathbf{D}_{\bar{S}}(\varepsilon)$ for $0 < \varepsilon < \varepsilon_2$ are or are not positive definite depending on whether $\mathbf{B}$ is or is not positive definite. In particular, the above relationship holds for $\varepsilon < \min(\varepsilon_1, \varepsilon_2)$. It follows from Theorem 3 that we may ascertain whether $L(\bar{S}, m) < L^*$ by checking for the positive definiteness of $\mathbf{B}$.

For any real symmetric matrix $\mathbf{A}$, let the notation $\mathbf{A} > 0$ mean that $\mathbf{A}$ is positive definite and let $\mathbf{A} \not\geqslant 0$ mean that $\mathbf{A}$ has at least one negative eigenvalue. Consider the following theorem.

THEOREM 4. *For $\varepsilon > 0$, let $\mathbf{D}$ be an $m \times m$ matrix of the form*

$$\mathbf{D} = \left(\frac{1}{\varepsilon}\right)\mathbf{G}\mathbf{G}' + \mathbf{A}, \quad \textit{where } \mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}$$

is symmetric and $\mathbf{G} = (\mathbf{G}_1', \mathbf{G}_2')'$ with $\mathbf{A}_{11}$ and $\mathbf{G}_1$ having dimensions $r \times r$, and with $\mathbf{G}_2$ being $(m - r) \times r$. In addition, let $\mathbf{G}_1$ be nonsingular and let $\mathbf{T} = (\mathbf{T}_1', \mathbf{T}_2')'$ be partitioned as $\mathbf{G}$ and satisfy

$$\mathbf{G}\mathbf{T}' + \mathbf{T}\mathbf{G}' = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{U} \end{pmatrix},$$

where $\mathbf{U}$ is arbitrary. Let $\mathbf{B} = \mathbf{A}_{22} - U$. Then for ε sufficiently small, $\mathbf{B} > 0 \Rightarrow \mathbf{D} > 0$ and $\mathbf{B} \not\geqslant 0 \Rightarrow \mathbf{D} \not\geqslant 0$.

Proof. The proof of Theorem 4 is given in the Appendix. It depends on showing (1) $\mathbf{B} > 0 \Rightarrow \mathbf{y}'\mathbf{D}\mathbf{y} > 0$ for all m-vectors $\mathbf{y} \neq 0$, and (2) $\mathbf{B} \not\geqslant 0 \Rightarrow$ there exists $\mathbf{y} \ni \mathbf{y}'\mathbf{D}\mathbf{y} < 0$.

Using the fact that $(Q + \varepsilon\mathbf{H})^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{V}' + (\frac{1}{\varepsilon})\mathbf{H}$, we obtain $\mathbf{D}(\varepsilon) = \mathbf{A} + (\frac{1}{\varepsilon})\mathbf{H}$, where $\mathbf{A} = \mathbf{V}(\Lambda^{-1} - (1/L^*)\mathbf{u}\mathbf{u}')\mathbf{V}'$ and $\mathbf{u} = \Lambda^{-1}\mathbf{V}'\mathbf{d}$. Thus the $m \times m$ submatrix $\mathbf{D}_{\bar{S}}(\varepsilon)$ is equal to $\mathbf{A}_{\bar{S}} + (\frac{1}{\varepsilon})\mathbf{H}_{\bar{S}}$, where $\mathbf{A}_{\bar{S}}$ and $\mathbf{H}_{\bar{S}}$ are corresponding submatrices of $\mathbf{A}$ and $\mathbf{H}$. Since $\mathbf{H}$ is positive semidefinite, so at least is $\mathbf{H}_{\bar{S}}$. Let r be the rank of $\mathbf{H}_{\bar{S}}$ and write $\mathbf{H}_{\bar{S}}$ in the form

$$\mathbf{H}_{\bar{S}} = \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{pmatrix},$$

where $\mathbf{H}_{11}$ is $r \times r$ and nonsingular. Then there exists a matrix $\mathbf{G}$ of the form $\mathbf{G} =$

$(\mathbf{G}_2', \mathbf{G}_2')'$ such that $\mathbf{GG}' = \mathbf{H}_{\bar{S}}$, where $\mathbf{G}_1$ is $r \times r$, nonsingular lower triangular, and $\mathbf{G}_2$ is $(m - r) \times r$. Writing

$$\mathbf{A}_{\bar{S}} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix},$$

where $\mathbf{A}_{11}$ is $r \times r$, we can find a matrix $\mathbf{T} = (\mathbf{T}_1', \mathbf{T}_2')'$ partitioned as $\mathbf{G}$, such that

$$\mathbf{GT}' + \mathbf{TG}' = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & U \end{pmatrix}$$

(see the following paragraph), where $\mathbf{U} = \mathbf{G}_2\mathbf{T}_2' + \mathbf{T}_2\mathbf{G}_2'$. Letting $\mathbf{B} = \mathbf{A}_{22} - \mathbf{U}$ (which does not depend on ε) and applying Theorem 4, it is seen that for ε sufficiently small (say, $\varepsilon < \varepsilon_2$), $\mathbf{B} > 0 \Rightarrow \mathbf{D}(\varepsilon) > 0$ and $\mathbf{B} \ngeq 0 \Rightarrow \mathbf{D}(\varepsilon) \ngeq 0$.

From the above result, an exact algorithm is developed that can be used when $\mathbf{Q}$ is singular and has the same sequential computational properties as Algorithm 1. Let $(gt)_{ij}$ denote the i,jth element of $\mathbf{GT}' + \mathbf{TG}'$ and let $\mathbf{A} = (a_{ij})$. By letting $\mathbf{T}_1$ be lower triangular, the elements of $\mathbf{T}$ can be easily obtained by sequential solution of $(gt)_{ij} = a_{ij}$ for $(i, j) = (1, 1), (2, 1), (2, 2), \ldots, (r, r), (r + 1, 1), \ldots, (r + 1, r), (r + 2, 1), \ldots, (r + 2, r), \ldots, (m, r)$. For convenience, we call this method Procedure T. Thus we have Algorithm 2.

ALGORITHM 2.

Step 1. Number the m-subsets lexicographically: $\bar{S}_1 = \{1, 2, \ldots, m\}$, $\bar{S}_2 = \{1, 2, \ldots, m - 1, m + 1\}, \ldots, \bar{S}_M = \{p - m + 1, \ldots, p\}$, where $M = \binom{p}{m}$.

Step 2. Find an initial candidate $\bar{S}^*$ for the best subset, say, by stepwise regression. Find $\mathbf{V}$, $\mathbf{\Lambda}^{-1}$, $\mathbf{H}$, and $\mathbf{u}$. Set $i = 1$.

Step 3. Let $L^* = L(\bar{S}^*, m)$ and $\mathbf{A} = \mathbf{V}[\mathbf{\Lambda}^{-1} - (1/L^*)\mathbf{uu}']\mathbf{V}'$.

Step 4. Use Cholesky factorization of $\mathbf{H}_{\bar{S}_i}$ to obtain r and $\mathbf{G}_1$; use Procedure T to obtain $\mathbf{T}_1$.

Step 5. For $j > r$, continue to use Cholesky factorization and Procedure T to find the $(j - r)$th rows of $\mathbf{G}_2$, $\mathbf{T}_2$, and $\mathbf{B}$. As each row of $\mathbf{B}$ is obtained, check for $\mathbf{B} > 0$ by another Cholesky factorization $\mathbf{B} = \mathbf{WW}'$. If the factorization fails, increment i; if $i = M + 1$, go to Step 7, otherwise repeat Step 4. If the factorization succeeds, go to Step 6.

Step 6. $\mathbf{D}_{\bar{S}_i}$ is positive definite; hence $L(\bar{S}_i, m) < L(\bar{S}^*, m)$. Set $\bar{S}^* = \bar{S}_i$, increment i by 1. If $i = M + 1$, go to Step 7; otherwise go back to Step 3.

Step 7. Stop. The best subset is $\bar{S}^*$.

As with Algorithm 1, i can be incremented by a number c to avoid checking subsets known to be inferior to $\bar{S}^*$. Again, let g be the last common component of $\bar{S}_i$ and $\bar{S}_{i+c}$. If $g \geqq r$, then $\mathbf{G}_1$ and $\mathbf{T}_1$ remain the same for $\bar{S}_{i+c}$, and only the last $m - (g - r)$ rows of $\mathbf{G}_2$, $\mathbf{T}_2$, and $\mathbf{W}$ need to be recomputed. If $g < r$, then the last $(r - g)$ rows of $\mathbf{G}_1$ and $\mathbf{T}_1$, as well as all of $\mathbf{G}_2$ and $\mathbf{T}_2$, must be recomputed.

5. Efficiency. If the singularity in $\mathbf{Q}$ is caused by $n < p$, as opposed to $\mathbf{X}$ having a special structure as in analysis-of-variance models, the rank of $\mathbf{H}$ is $p - n$; hence r, the rank of $\mathbf{H}_{\bar{S}}$ is at most $p - n$. At least $r + 1$ variables must be removed from the regression before the reduction sum of squares exceeds zero; hence a lower bound of $\binom{p}{r}$ subsets of size $r + 1$ or greater must be checked before the one with minimum reduction is found. In terms of Algorithm 2, this means all $\binom{p}{r}$ $\mathbf{G}_1$ and $\mathbf{T}_1$ matrices must be computed. This makes the method less efficient as p is increased for fixed n, thus restricting the number of transformations in the empirical-modeling application. On the other hand, if p exceeds n by only a small amount, the method is almost as efficient as a full-rank reduction

search. The only way to circumvent this problem is to search for the best k-subset directly, not its m-dimensional complement by reduction. A possible approach for doing so is described in § 6.

In Algorithm 2, it is necessary to find $\mathbf{H}_{11}$ which is $r \times r$ and has rank r. Experiments on randomly generated data sets with more variables than observations have shown that not all $(p - n) \times (p - n)$ PSMs of $\mathbf{H}_{\bar{S}}$ have rank $p - n$; some may have rank $p - n - 1$. It is therefore possible to have $r = p - n$, but $\mathbf{H}_{11}$ consists of rows and columns $\{1, 2, \ldots, p - n - 1, x\}$ where $x > p - n$. In this case, some efficiency is lost in keeping track of the index x. The value of r can be calculated during the Cholesky factorization of $\mathbf{H}_{\bar{S}}$. If the $(p - n)$th diagonal element of $\mathbf{G}_1$ is less than some threshold, assume it is zero and test subsequent rows of $\mathbf{H}_{\bar{S}}$ to see if one is linearly independent of the first $p - n - 1$ rows. If no such row is found then take $r = p - n - 1$.

6. Direct search for the best regression. When $p - n$ is large, it has been noted in § 5 that any reduction search is inefficient. Let $\mathscr{P}_k$ be the set of all real symmetric matrices $\mathbf{A}$, such that all $k \times k$ PSMs of $\mathbf{A}$ are positive definite or semidefinite. One may use Theorem 3 to show that no k-variate regression sum of squares exceeds R^* if and only if $\mathbf{K} \in \mathscr{P}_k$, where $\mathbf{K} = \mathbf{Q} - (1/R^*)\mathbf{dd}'$. Thus, any PSM of $\mathbf{K}$ having a negative eigenvalue corresponds to a subset with a regression sum of squares that exceeds R^*. For small values of k, attempted factorization of all $k \times k$ PSMs of $\mathbf{K}$ might actually be more efficient than any reduction procedure; however, in most cases better methods need to be developed for checking if $\mathbf{K} \in \mathscr{P}_k$ without examining all possible PSMs.

A sufficient condition for $\mathbf{K} \in \mathscr{P}_k$ is that every $k \times k$ PSM is diagonally dominant (kDD). This condition is easy to check; merely sum the $k - 1$ largest absolute off-diagonal elements on each row of $\mathbf{K}$ and compare to the diagonal element. Unfortunately, $\mathbf{K}$ being kDD is not necessary for $\mathbf{K} \in \mathscr{P}_k$; however, it might be possible to make a series of transformations $\mathbf{K}_i = \mathbf{T}_i\{\mathbf{K}_{i-1}\}$, $i = 1, 2, \ldots$, such that $\mathbf{T}_i^{-1}$ preserves membership in $\mathscr{P}_k$ and for some i, $\mathbf{K}_i$ is kDD. Two such transformations are $\mathbf{T}\{\mathbf{K}\} = \mathbf{DKD}$, where $\mathbf{D}$ is diagonal nonsingular; or $\mathbf{T}\{\mathbf{K}\} = \mathbf{K} - \mathbf{P}$, where $\mathbf{P}$ is positive definite. Another approach is to search for any vector $\mathbf{x}$ with $n - k$ zero elements such that $\mathbf{x}'\mathbf{K}\mathbf{x} < 0$. If such a vector is found, then the position of the nonzero elements defines a regression subset that beats R^*.

7. Numerical results. Algorithm 1, with ridge selection ($\mathbf{M} = \mathbf{I}$), and Algorithm 2 were tested on four groups of randomly generated data, where the ith group consisted of N_i independent regression sets of $(\mathbf{X}, \mathbf{y})$-pairs with $n = n_i$, $p = p_i$, and $k = k_i$, as shown in Table 1.

For each regression set, the rows of the $n \times (p + 1)$ matrix $(\mathbf{X} \mid \mathbf{y})$ were generated independently as $N_{p+1}(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ was itself generated as a $(p + 1) \times (p + 1)$ sample covariance matrix based on $p + 1$ observations of $N(0, \mathbf{I}_{p+1})$ vectors. This two-stage process introduces many near-collinearities in the columns of $\mathbf{X}$, thus producing excellent

TABLE 1
Test groups.

Group (i)	n_i	p_i	k_i	N_i
1	10	15	6	24
2	14	18	6	10
3	18	24	12	10
4	24	28	8	10

TABLE 2(a)
Simulation results: $n = 10$, $p = 15$, $k = 6$.

Efficiency ratios (F)			R^2-values of best subset found			
Ridge selection[1]		Exact[2]		Ridge selection[1]		Exact[2]
$\varepsilon = .001$	$\varepsilon = .0001$	method	Step-up	$\varepsilon = .001$	$\varepsilon = .0001$	method
0.08623	0.10986	0.27451	0.91224	0.95552	0.95552	0.95552
0.09072	0.15309	0.40192	0.94257	0.97945*	0.98152	0.98152
0.07873	0.09756	0.26024	0.94882	0.99485	0.99485	0.99485
0.02793	0.07027	0.25124	0.98841	0.99876	0.99876	0.99876
0.05828	0.11073	0.32055	0.99550	0.99550*	0.99604	0.99604
0.10735	0.11635	0.28401	0.94004	0.99151*	0.99457	0.99457
0.10873	0.15565	0.40116	0.95395	0.99687*	0.99704	0.99704
0.06058	0.10153	0.27425	0.98226	0.99677	0.99677	0.99677
0.05755	0.09597	0.25443	0.97451	0.99075	0.99075	0.99075
0.10753	0.13643	0.33272	0.97810	0.99061*	0.99523	0.99523
0.07199	0.12952	0.36188	0.97657	0.99742	0.99742	0.99742
0.03637	0.06452	0.24260	0.99250	0.99929	0.99920*	0.99929
0.05211	0.09935	0.29982	0.99332	0.99757	0.99757	0.99757
0.02991	0.08943	0.25866	0.99381	0.99883	0.99883	0.99883
0.02011	0.04577	0.21638	0.98897	0.99956*	0.99965	0.99965
0.08636	0.11156	0.30204	0.95600	0.99513	0.99513	0.99513
0.06052	0.10179	0.29149	0.95561	0.99651	0.99651	0.99651
0.13624	0.16054	0.39524	0.87658	0.98880	0.98880	0.98880
0.09253	0.11278	0.27141	0.98256	0.99232*	0.99287	0.99287
0.10768	0.11142	0.30007	0.98923	0.99331*	0.99649	0.99649
0.09452	0.12845	0.32134	0.95980	0.99629	0.99629	0.99629
0.07180	0.09586	0.24483	0.98200	0.99622	0.99622	0.99622
0.05091	0.10170	0.26620	0.93971	0.98937	0.98937	0.98937
0.15752	0.17089	0.38740	0.97799	0.99743	0.99743	0.99743

* Failed to select best subset.
[1] Ridge selection using Algorithm 1 with $\mathbf{M} = \mathbf{I}$.
[2] Algorithm 2.

TABLE 2(b)
Simulation results: $n = 14$, $p = 18$, $k = 6$.

Efficiency ratios (F)			R^2-values of best subset found			
Ridge selection[1]		Exact[2]		Ridge selection[1]		Exact[2]
$\varepsilon = .001$	$\varepsilon = .0001$	method	Step-up	$\varepsilon = .001$	$\varepsilon = .0001$	method
0.05746	0.07228	0.20735	0.96333	0.98478	0.98478	0.98478
0.04602	0.05129	0.14633	0.87845	0.91219	0.91219	0.91219
0.08098	0.08859	0.26968	0.84224	0.90962	0.90962	0.90962
0.12106	0.11241	0.34823	0.81274	0.93069	0.93069	0.93069
0.11620	0.11678	0.36293	0.89084	0.96111	0.96111	0.96111
0.06767	0.07534	0.21400	0.91288	0.92999	0.92999	0.92999
0.08026	0.08255	0.25763	0.91985	0.96124	0.96124	0.96124
0.03713	0.04353	0.12372	0.94692	0.95210	0.95210	0.95210
0.12280	0.13929	0.43548	0.91392	0.94192	0.94192	0.94192
0.08019	0.08541	0.26164	0.89249	0.95354	0.95354	0.95354

[1] Ridge selection using Algorithm 1 with $\mathbf{M} = \mathbf{I}$.
[2] Algorithm 2.

TABLE 2(c)
Simulation results: $n = 18$, $p = 24$, $k = 12$.

Efficiency ratio (F)			R^2-values of best subset found			
Ridge selection[1]		Exact[2]		Ridge selection[1]		Exact[2]
$\varepsilon = .001$	$\varepsilon = .0001$	method	Step-up	$\varepsilon = .001$	$\varepsilon = .0001$	method
0.00078	0.00285	0.01008	0.98910	0.99863*	0.99863*	0.99866
0.00115	0.00238	0.00896	0.98912	0.99931*	0.99955	0.99955
0.00034	0.00221	0.00911	0.99392	0.99819*	0.99841*	0.99873
0.00149	0.00382	0.00991	0.99227	0.99898*	0.99979	0.99979
0.00047	0.00205	0.00899	0.99224	0.99759*	0.99901	0.99901
0.00088	0.00332	0.01003	0.99394	0.99787	0.99787	0.99787
0.00263	0.00425	0.01342	0.95633	0.99527	0.99527	0.99527
0.00067	0.00383	0.01527	0.99312	0.99585*	0.99784	0.99784
0.00203	0.00317	0.01165	0.98580	0.99642*	0.99838	0.99838
0.00100	0.00270	0.01028	0.98034	0.99672*	0.99711	0.99711

* Failed to select best subset.
[1] Ridge selection using Algorithm 1 with $\mathbf{M} = \mathbf{I}$.
[2] Algorithm 2.

"nasty" test cases. All data was normalized so that $\mathbf{X}'\mathbf{X}$ was in correlation form with $\mathbf{y}'\mathbf{y} = 1$. Regressions were calculated without intercept terms.

The primary standard of efficiency was taken as the number of elements in the auxiliary matrices ($\mathbf{B}$ for Algorithm 1, $\mathbf{G}$, $\mathbf{T}$, and $\mathbf{B}$ for Algorithm 2) calculated in the process of finding the best subset. In Tables 2(a)–2(d), this number was expressed as a ratio F to the number of elements that would have had to be computed to find the best subset by "brute force" factorization of all $\binom{p}{k}$ submatrices of $\mathbf{K}$ in § 6. Algorithm 1 was tested with $\varepsilon = .001$ and .0001. Cases for which it failed to find the best subset are marked with an asterisk (*). Results for Algorithm 2 are listed under the heading Exact. The

TABLE 2(d)
Simulation results: $n = 24$, $p = 28$, $k = 8$.

Efficiency ratios (F)			R^2-values of best subset found			
Ridge selection[1]		Exact[2]		Ridge selection[1]		Exact[2]
$\varepsilon = .001$	$\varepsilon = .0001$	method	Step-up	$\varepsilon = .001$	$\varepsilon = .0001$	method
0.00288	0.00356	0.01462	0.94783	0.94783	0.94783	0.94783
0.02022	0.02327	0.10229	0.89189	0.92462	0.92462	0.92462
0.02297	0.02516	0.12104	0.93876	0.93876	0.93876	0.93876
0.01285	0.01427	0.06227	0.84248	0.84248*	0.84274	0.84274
0.00820	0.00937	0.04126	0.90499	0.94406	0.94406	0.94406
0.01255	0.01450	0.06364	0.80411	0.90515	0.90515	0.90515
0.07629	0.08135	0.41797	0.92954	0.95515	0.95515	0.95515
0.01452	0.01590	0.06916	0.95050	0.96230	0.96230	0.96230
0.01348	0.01522	0.07278	0.78169	0.89145	0.89145	0.89145
0.02299	0.02616	0.11613	0.93577	0.94562	0.94562	0.94562

* Failed to select best subset.
[1] Ridge selection using Algorithm 1 with $\mathbf{M} = \mathbf{I}$.
[2] Algorithm 2.

value of $R(S, k)$ for the subset S found to be "best" is given in the three right-hand columns. Theoretically, the values of $R(S, k)$ for Algorithm 2 should be the true maxima. For the two smaller groups ($p = 15$ and $p = 18$), these values were verified by the brute force method. For comparison purposes, the value of the step-up regression sum of squares is also given.

It can be seen that, in most cases, ridge selection found the best subset, especially with $\varepsilon = .0001$. Generally, the efficiency of Algorithm 1 was three to ten times better than Algorithm 2. Some experimentation with $\varepsilon = .00001$ resulted in numerous errors that led to spurious identification of "better" subsets. When ridge selection failed to find the best subset, it generally came close in terms of the corresponding regression sum of squares. For the values of n, p, and k shown here, it can be seen that if exact results are desired, Algorithm 2 is still considerably more efficient than brute force. As discussed previously, when $k < p - n$, reduction-based methods are less efficient than examination of all k-subsets.

8. Summary. The computational problem of finding the best-fitting subset of independent variables in least-squares regression with a fixed subset size has been addressed, especially in the context of the nonfull-rank case with more variables than observations. Generally, there is a range of n, p, and k ($n < p$ and $k > p - n$) for which existing reduction-based methods cannot be used and where Algorithm 2 is able to find the best subset more efficiently than can exhaustive enumeration. In the development of Algorithm 2, it was discovered that ridge selection using a ridge matrix of $\mathbf{I}$ as implemented in Algorithm 1, usually finds the best subset, and that it is more efficient than Algorithm 2. The most likely circumstances of the application of Algorithm 2 are for empirical modeling situations where measurement or other random model errors are small or nonexistent. The region of n, p, k-space for which k is relatively small (regardless of whether n exceeds p), remains the most challenging for developing better selection methods. Possible approaches were put forth in § 6 as starting points for further investigation.

Appendix. Proof of Theorem 4. Let $\mathbf{y} = (\mathbf{y}_1', \mathbf{y}_2')'$ be any $m \times 1$ vector with $\|\mathbf{y}\|^2 = 1$, where $\mathbf{y}_1$ is $r \times 1$ and $\mathbf{y}_2$ is $(m - r) \times 1$. It will first be shown that $\mathbf{B} > 0 \Rightarrow \mathbf{y}'\mathbf{D}\mathbf{y} > 0$. We make use of the following lemma.

LEMMA. *Let γ_r be the minimum eigenvalue of $\mathbf{G}_1\mathbf{G}_1'$ and γ_2 be the maximum eigenvalue of $\mathbf{G}_2\mathbf{G}_2'$. Then $\|\mathbf{G}'\mathbf{y}\|^2 < c \Rightarrow \|\mathbf{y}_2\|^2 \geqq (\gamma_r - c)/(\gamma_r + \gamma_2)$.*

Proof. Suppose that $\|\mathbf{G}'\mathbf{y}\|^2 < c$. We have $\mathbf{G}_1'\mathbf{y}_1 = \mathbf{G}'\mathbf{y} - \mathbf{G}_2'\mathbf{y}_2$

$$\Rightarrow \|\mathbf{G}_1'\mathbf{y}_1\|^2 \leqq \|\mathbf{G}'\mathbf{y}\|^2 + \|\mathbf{G}_2'\mathbf{y}_2\|^2 < c + \|\mathbf{G}_2'\mathbf{y}_2\|^2$$

$$\Rightarrow \gamma_r\|\mathbf{y}_1\|^2 < c + \gamma_2\|\mathbf{y}_2\|^2$$

$$\Rightarrow \gamma_r(1 - \|\mathbf{y}_2\|^2) < c + \gamma_2\|\mathbf{y}_2\|^2$$

$$\Rightarrow \|\mathbf{y}_2\|^2 > \frac{(\gamma_r - c)}{(\gamma_r + \gamma_2)}. \qquad \square$$

Now, from the definitions of $\mathbf{G}$, $\mathbf{T}$, and $\mathbf{B}$, we have

$$\varepsilon\mathbf{y}'\mathbf{D}\mathbf{y} = \mathbf{y}'(\mathbf{G} + \varepsilon\mathbf{T})(\mathbf{G} + \varepsilon\mathbf{T})'\mathbf{y} + \varepsilon\mathbf{y}_2'\mathbf{B}\mathbf{y}_2 - \varepsilon^2\mathbf{y}'\mathbf{T}\mathbf{T}'\mathbf{y} \tag{A.1}$$

$$\geqq \varepsilon\beta\|\mathbf{y}_2\|^2 - \varepsilon^2\tau, \tag{A.2}$$

where β is the minimum eigenvalue of $\mathbf{B}$ and τ is the maximum eigenvalue of $\mathbf{T}\mathbf{T}'$.

Suppose that $\|\mathbf{G}'\mathbf{y}\|^2 < \gamma_r/2$. Then by the above lemma, it holds that

$$\|\mathbf{y}_2\|^2 > \left(\frac{1}{2}\right)\gamma_r \Big/ (\gamma_r + \gamma_2). \tag{A.3}$$

Let $\theta = (\frac{1}{2})\gamma_r/(\gamma_r + \gamma_2)$. Clearly $\theta > 0$, since $\gamma_2 \geqq 0$ and $\gamma_r > 0$, $\mathbf{G}_1$ being nonsingular. If $\mathbf{B} > 0$, then $\beta > 0$; hence by (A.2) and (A.3), we must have $\mathbf{y}'\mathbf{D}\mathbf{y} > 0$ for any $\varepsilon > 0$ if $\tau = 0$, and for $0 < \varepsilon < \beta\theta/\tau$, if $\tau > 0$.

Suppose now that $\|\mathbf{G}'\mathbf{y}\|^2 \geqq \gamma_r/2$. Equation (A.1) may be rewritten

$$\varepsilon\mathbf{y}'\mathbf{D}\mathbf{y} = \mathbf{y}'\mathbf{G}\mathbf{G}'\mathbf{y} + \varepsilon\mathbf{y}_2'\mathbf{B}\mathbf{y}_2 + \varepsilon\mathbf{y}'\mathbf{A}^*\mathbf{y}, \tag{A.4}$$

where $\mathbf{A}^* = \mathbf{G}\mathbf{T}' + \mathbf{T}\mathbf{G}'$. Hence $\varepsilon\mathbf{y}'\mathbf{D}\mathbf{y} > \gamma_r/2 - \varepsilon\alpha^*$, where $-\alpha^*$ is the minimum eigenvalue of $\mathbf{A}^*$. It follows that $\mathbf{y}'\mathbf{D}\mathbf{y} > 0$ for any $\varepsilon > 0$ if $\alpha^* = 0$, and for $0 < \varepsilon < \gamma_r/2\alpha^*$, if $\alpha^* > 0$. This completes the first part of the proof; i.e., $\mathbf{B} > 0 \Rightarrow \mathbf{y}'\mathbf{D}\mathbf{y} > 0$.

Suppose that $\mathbf{B} \ngeqq 0$. Then there exists $\mathbf{z}_2$ such that $\mathbf{z}_2'\mathbf{B}\mathbf{z}_2 < 0$. Let $\mathbf{z}_1 = -\mathbf{G}_1^{-1}\mathbf{G}_2'\mathbf{z}_2$ and $\mathbf{z} = (\mathbf{z}_1', \mathbf{z}_2')'$. Then $\mathbf{G}'\mathbf{z} = 0$; hence from (A.4) with $\mathbf{y} = \mathbf{z}$ and from the definition of $\mathbf{A}^*$, $\varepsilon\mathbf{z}'\mathbf{D}\mathbf{z} = \varepsilon\mathbf{z}_2'\mathbf{B}\mathbf{z}_2 < 0 \Rightarrow \mathbf{z}'\mathbf{D}\mathbf{z} < 0$ for $\varepsilon > 0$. Thus $\mathbf{B} \ngeqq 0 \Rightarrow \mathbf{D} \ngeqq 0$ for any $\varepsilon > 0$ and the theorem is proved. □

REFERENCES

[1] G. W. FURNIVAL AND R. W. WILSON, *Regression by leaps and bounds*, Technometrics, 16 (1974), pp. 499–511.

[2] H. GOLUB AND C. S. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[3] S. S. GUPTA, D. Y. HUANG, AND C. L. CHANG, *Selection procedures for optimal subsets of regression variables*, Design of Experiments: Ranking and Selection, Marcel Dekker, New York, 1984, pp. 67–75.

[4] R. R. HOCKING AND R. N. LESLIE, *Selection of the best subset in regression analysis*, Technometrics, 9 (1967), pp. 531–540.

[5] R. R. HOCKING, *The analysis and selection of variables in linear regression*, Biometrics, 32 (1976), pp. 1–49.

[6] A. E. HOERL AND R. W. KENNARD, *Ridge regression: Biased estimation for non-orthogonal problems*, Technometrics, 12 (1970), pp. 55–63.

[7] R. W. HOERL, J. H. SCHUENEMEYER, AND A. E. HOERL, *A simulation of biased estimation and subset selection techniques*, Technometrics, 28 (1986), pp. 369–380.

[8] A. J. MILLER, *Selection of subsets of regression variables*, (with discussion), J. Roy. Statist. Soc. Ser. A, 147 (1984), pp. 389–425.

[9] S. R. SEARLE, *Linear Models*, John Wiley & Sons, New York, 1971.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 205–214, January 1994

017

NUMERICAL SOLUTION OF THE EIGENPROBLEM FOR BANDED, SYMMETRIC TOEPLITZ MATRICES*

SUSAN L. HANDY† AND JESSE L. BARLOW‡

Abstract. A fast method for calculating the eigensystem of a banded, symmetric Toeplitz matrix is presented. The method is based on the use of rank-one updates and employs deflation to reduce problem size whenever possible.

Key words. eigenproblem, Toeplitz matrices, banded matrices

AMS subject classifications. 65F15, 65F50

1. Introduction. In this paper, we consider the problem of calculating the eigensystem of an $n \times n$ banded, symmetric Toeplitz (BST) matrix with bandwidth k. Recall that the distinguishing characteristic of a Toeplitz matrix is that its entries along any diagonal parallel to the main diagonal are constant. Formally we have the following definition.

DEFINITION 1.1. *A real, symmetric Toeplitz matrix T has the form*

$$T = (t_{ij})_{i,j=1}^{n} = (c_{|i-j|+1})_{i,j=1}^{n}, \tag{1.1}$$

where each c_i is real. Thus any such matrix T is of the form

$$T = \begin{bmatrix} c_1 & c_2 & c_3 & \cdot & \cdot & \cdot & c_n \\ c_2 & c_1 & c_2 & & & & \cdot \\ c_3 & c_2 & \ddots & & & & \cdot \\ \cdot & & & c_1 & & & \cdot \\ \cdot & & & & \ddots & c_2 & c_3 \\ \cdot & & & & c_2 & c_1 & c_2 \\ c_n & \cdot & \cdot & \cdot & c_3 & c_2 & c_1 \end{bmatrix}. \tag{1.2}$$

We can use the notation

$$T = \mathrm{Toep}_n(c_1, c_2, \ldots, c_n) \tag{1.3}$$

to denote the matrix given in (1.1) and (1.2).

DEFINITION 1.2. *A real $n \times n$ Toeplitz matrix T is said to be a* BST *matrix if it is symmetric and has the form*

$$T = \mathrm{Toep}_n(c_1, c_2, \ldots, c_k, 0, \ldots, 0) \tag{1.4}$$

for some index $k < n$.

T is then referred to as a $2k - 1$-diagonal BST matrix or as a BST matrix with bandwidth k. Let $T_{n,k}$ denote the matrix T defined in (1.4).

Another class of matrices whose structure is very similar to that of Toeplitz matrices is the class of Hankel matrices. Hankel matrices will play a significant role in the method that we present for solving the symmetric eigenvalue problem for BST matrices. A matrix

* Received by the editors March 25, 1991; accepted for publication (in revised form) June 29, 1992.

† HRB Systems, Inc., P.O. Box 60 Science Park Road, State College, Pennsylvania 16804 (slh@icf.hrb.com).

‡ Department of Computer Science, The Pennsylvania State University, University Park, Pennsylvania 16802 (barlow@cs.psu.edu). The research of this author was supported by the National Science Foundation grant CCR-90000526.

is said to be Hankel if its entries are constant along each cross-diagonal. Formally we have the following definition.

DEFINITION 1.3. *A real $n \times n$ Hankel matrix has the form*

$$H = (h_{ij})_{i,j=1}^{n} = (d_{i+j-1})_{i,j=1}^{n},$$

where each d_i is real. If H is also persymmetric (i.e., symmetric about the main cross-diagonal) then $d_{2n-j} = d_j$, $1 \leqq j \leqq n$ and H has the form

$$H = \begin{bmatrix} d_1 & d_2 & d_3 & \cdot & \cdot & \cdot & d_n \\ d_2 & d_3 & & & & & \cdot \\ d_3 & & & & \iddots & & \cdot \\ \cdot & & & d_n & & & \cdot \\ \cdot & & & & & & d_3 \\ \cdot & \iddots & & & & d_3 & d_2 \\ d_n & \cdot & \cdot & \cdot & d_3 & d_2 & d_1 \end{bmatrix}. \tag{1.5}$$

As with real, symmetric Toeplitz matrices, the first row of a persymmetric Hankel matrix completely determines the remaining entries. Thus we use the notation

$$H = \mathrm{Hank}_n(d_1, d_2, \ldots, d_n) \tag{1.6}$$

to denote the $n \times n$ real, persymmetric Hankel matrix and $H_{n,k}$ to denote the Hankel matrix

$$H_{n,k} = \mathrm{Hank}_n(d_1, d_2, \ldots, d_k, 0, \ldots, 0).$$

A third class of matrices in which we are strongly interested is the class of "cross-sum" matrices.

DEFINITION 1.4. *C_n is said to be a cross-sum matrix if it is an element of the set*

$$\Theta_n = \{B \equiv (b_{ij}) : b_{ij} \in \mathbf{R} \quad \textit{and} \quad b_{i-1,j} + b_{i+1,j} = b_{i,j-1} + b_{i,j+1},\ i, j = 1, 2, \ldots, n\}$$

with $b_{0,j} = b_{i,0} = b_{i,n+1} = b_{n+1,j} = 0$.

As with the symmetric Toeplitz and persymmetric Hankel matrices, the entries in the first row of a cross-sum matrix are enough to completely determine the remaining entries.

A number of other authors have developed methods for solving the symmetric, eigenvalue problem for banded Toeplitz matrices. Arbenz [2] investigates a technique that embeds the Toeplitz matrix into a higher-order circulant matrix, computes the eigensystem of the circulant, and then solves the Toeplitz eigenproblem as a restricted eigenvalue problem. Trench [12] calculates a formula for the characteristic polynomial of an nth-order Toeplitz matrix T with bandwidth k in terms of the zeroes of a kth degree polynomial whose coefficients are independent of n. Bini and Pan [6] present algorithms for computing $p(\lambda) = \det(T - \lambda I)$ and the ratio $p(\lambda)/p'(\lambda)$ in the case where T is a block BST matrix.

2. Numerical solution of the BST eigenproblem.

2.1. Solution of the rank-one updating problem. In this section, we will briefly summarize the solution from Bunch, Nielsen, and Sorensen to the rank-one eigensystem updating problem. (For complete details, we refer the reader to [7] and [9].) Given an $n \times n$ diagonal matrix $D = \mathrm{diag}(\delta_1, \delta_2, \ldots, \delta_n)$, $z = (z_1, z_2, \ldots, z_n)^T \in \mathbf{R}^n$, and ρ, a

nonzero scalar, we wish to calculate $(\hat{D}, \hat{Q})$ such that

$$D + \rho zz^T = \hat{Q}\hat{D}\hat{Q}^T, \tag{2.1}$$

where $\hat{D} = \text{diag}\,(\hat{\delta}_1, \hat{\delta}_2, \ldots, \hat{\delta}_n)$ and $\hat{Q}$ is orthogonal.

Initially, we make the assumptions that each δ_i is distinct and that no z_i is zero. It is then shown that the roots of

$$f(\lambda) \equiv 1 + \rho \sum_{j=1}^{n} \frac{z_j^2}{\delta_j - \lambda} \tag{2.2}$$

are exactly the eigenvalues $\hat{\delta}_i$ of $D + \rho zz^T$ and that the corresponding eigenvectors $\hat{q}_i$ are given by

$$\hat{q}_i = \gamma_i \Delta_i^{-1} z, \tag{2.3}$$

where

$$\Delta_i = \text{diag}\,(\delta_1 - \hat{\delta}_i, \ldots, \delta_n - \hat{\delta}_i) \tag{2.4}$$

and

$$\gamma_i = \|\Delta_i^{-1} z\|_2. \tag{2.5}$$

For each root $\hat{\delta}_i$, the function $f(\lambda)$ is separated into the sum of positive and negative terms of the form $f(\lambda) = 1 + \phi(\lambda) + \psi(\lambda)$, and the roots of $f(\lambda)$ are calculated iteratively by constructing rational interpolants to ϕ and ψ. The iteration converges quadratically to the roots $\hat{\delta}_i$ given an appropriate starting point. (Further details are available from Bunch, Nielsen, and Sorensen [7].)

In the implementation of this method by Dongarra and Sorensen , the iteration is reformulated so that they solve for the difference between successive iterates $\tau_i = \lambda_i - \lambda_{i-1}$ rather than λ_i itself. The quantities $\delta_j - \lambda_k$ used in (2.4) to calculate the eigenvectors are retained, and the iterative corrections τ_i are applied to these quantities as well as to the eigenvalue approximations. Cancellation in the calculation of these differences is avoided because the corrections decrease in size and are eventually applied to the lowest-order bits. Stringent convergence criteria are applied that ensure a small residual and orthogonality of the eigenvectors to full machine accuracy. The stopping criteria employed are $|f(\lambda)| \leq \mu \cdot \max(|\delta_1|, |\delta_n|)$ and $|\tau| \leqq \mu \cdot \min(|\delta_i - \lambda|, |\delta_{i+1} - \lambda|)$, where λ is the current iterate, τ is the last calculated iterative correction, and μ is a small tolerance close to the machine unit. In his paper on the numerical stability of updating methods for the symmetric eigenvalue problem, Barlow [4] discusses the use of the stopping criterion $|f(\lambda)|/|f'(\lambda)| \leqq \mu \cdot \min(|\delta_i - \lambda|, |\delta_{i+1} - \lambda|)$.

We made two assumptions about the problem originally: all components of z are nonzero and the eigenvalues of D are distinct. If either of these conditions does not hold, we may reduce the problem size. If $z_i = 0$ for some i, then the ith eigenvalue of D and its corresponding eigenvector already form an eigenpair for $D + \rho zz^T$. Numerically, we accept the pair and the problem size reduces by one if $|\rho z_i|$ is small. If $|\delta_i - \delta_{i+1}| \leqq \varepsilon$, then, by applying an appropriately constructed Givens rotation to (2.1), we can reduce this case to the previous one. Again the problem size can be reduced by one. Cuppen [8] used this updating procedure to solve the symmetric tridiagonal eigenproblem.

Improvements to the Dongarra and Sorensen implementation are discussed in Barlow [4] and in Sorensen and Tang [11].

2.2. Existence of the cross-sum representation for any BST matrix. To employ the updating technique above, first we must show that it is always possible for us to split a

Toeplitz matrix into the sum of a cross-sum matrix and a Hankel matrix. To show that the splitting

$$\begin{aligned}&\mathrm{Toep}_n(d_1, d_2, \ldots, d_k, 0, \ldots, 0)\\ &\quad = \mathrm{CrossSum}_n(d_1 - d_3, \ldots, d_{k-2} - d_k, d_{k-1}, d_k, 0, \ldots, 0)\\ &\qquad + \mathrm{Hank}_n(d_3, d_4, \ldots, d_k, 0, \ldots, 0)\end{aligned} \tag{2.6}$$

is valid for any n and for any $3 \leqq k \leqq n$, we must show that

$$\begin{aligned}C = &\mathrm{Toep}_n(d_1, d_2, \ldots, d_k, 0, \ldots, 0)\\ &- \mathrm{Hank}_n(d_3, d_4, \ldots, d_k, 0, \ldots, 0)\end{aligned} \tag{2.7}$$

satisfies the cross-sum property. Assume that d_r is defined as follows:

$$d_r = \begin{cases} 0, & k+1 \leqq r \leqq n,\\ d_{2n+r-4}, & n+1 \leqq r \leqq 2n+1.\end{cases}$$

By definition of the elements of a symmetric Toeplitz matrix and a persymmetric Hankel matrix, we have

$$c_{ij} = d_{|i+j|+1} - d_{i+j+1}, \qquad 0 \leqq i, j \leqq n+1. \tag{2.8}$$

THEOREM 2.1. *The matrix C as defined in* (2.8) *is an element of the set of cross-sum matrices.*

Proof. For C to be a cross-sum matrix, we require that c_{ij} be 0 for i or j equal to 0 or $n+1$ and that $c_{i-1,j} + c_{i+1,j} = c_{i,j-1} + c_{i,j+1}$. It is straightforward to verify that $c_{ij} = 0$ if either i or j is 0 or $n+1$. Define Δ_{ij} as

$$\Delta_{ij} = c_{i-1,j} + c_{i+1,j} - c_{i,j-1} - c_{i,j+1}.$$

Then we have

$$\begin{aligned}\Delta_{ij} = &(d_{|i-j-1|+1} - d_{i+j}) + (d_{|i-j+1|+1} - d_{i+j+2}) - (d_{|i-j+1|+1} - d_{i+j})\\ &- (d_{|i-j-1|+1} - d_{i+j+2}) = 0\end{aligned}$$

for $1 \leqq i, j \leqq n$. Thus C is a cross-sum matrix. This completes the proof. □

The class of cross-sum matrices is valuable in the context of our problem because solution of the eigenproblem for a cross-sum matrix is fast. From Bini and Capovani [5], it is known that the class of cross-sum matrices Θ_n, defined in the previous section, is equivalent to the algebra generated over $\mathbf{R}$ by the matrix

$$T = \mathrm{Toep}_n(0, 1, 0, \ldots, 0) = \begin{bmatrix} 0 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ & 1 & 0 & 1 & & & \\ & & & \ddots & & & \\ & & & 1 & 0 & 1 & \\ & & & & 1 & 0 & \end{bmatrix}.$$

Thus any A in Θ_n can be expressed as

$$A = \sum_{i=0}^{n-1} c_{i+1} T^i.$$

From Proposition 2.2 in [5, p. 102], it is shown that we may solve for the values $c_1, c_2, \ldots, c_n$ by solving an upper triangular system of the form $Sc = a$, where a is the first row of A in Θ_n and the entries of $S = [s_{ij}]$ are given by the equations

$$\begin{aligned} s_{ij} &= s_{i-1,j-1} + s_{i+1,j-1}, && 1 \leqq i \leqq j \leqq n, \\ s_{ij} &= 0, && 0 \leqq j < i \leqq n, \\ s_{0,j} &= 0, && 1 \leqq j \leqq n, \\ s_{0,0} &= 1. \end{aligned}$$

From [5], the eigendecomposition of $T = U^T D U$ is known to be

$$D = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

where

$$\lambda_j = 2 \cos\left(\frac{\pi j}{n+1}\right) \qquad j = 1, 2, \ldots, n$$

and

$$U = [u_{ij}] \qquad i, j = 1, 2, \ldots, n,$$

where

$$u_{ij} = \sqrt{\frac{2}{n+1}} \sin\left(\frac{\pi ij}{n+1}\right).$$

Thus the eigenvalues of A in Θ_n are

$$\lambda_i(A) = \sum_{j=0}^{n-1} [\lambda_i(T)]^j \cdot c_{j+1}$$

as $i = 1, 2, \ldots, n$. The eigenvectors of A are exactly those of T, which are the column vectors of U.

We have shown that we can express a BST matrix as the sum of a cross-sum matrix (whose eigenvalues and eigenvectors can be easily calculated) and a persymmetric, "offset" Hankel matrix. Thus we have

$$T_{n,k} = C_{n,k} + H_{n,k-2}, \tag{2.9}$$

where

$$C_{n,k} = \operatorname{CrossSum}_n(c_1 - c_3, c_2 - c_4, \ldots, c_{k-2} - c_k, c_{k-1}, c_k, 0, \ldots, 0) \tag{2.10}$$

and

$$H_{n,k-2} = \operatorname{Hank}_n(c_3, c_4, \ldots, c_k, 0, \ldots, 0). \tag{2.11}$$

If the spectral decomposition of $C_{n,k}$ is given by

$$C_{n,k} = UDU^T,$$

then we have

$$T_{n,k} = UDU^T + H_{n,k-2} = U(D + U^T H_{n,k-2} U)U^T.$$

The problem of calculating the eigensystem of $T_{n,k}$ reduces to that of finding the eigensystem of

$$D + U^T H_{n,k-2} U, \tag{2.12}$$

where

$$UDU^T = \text{CrossSum}_n (c_1 - c_3, c_2 - c_4, \ldots, c_{k-2} - c_k, c_{k-1}, c_k, 0, \ldots, 0)$$

and

$$H_{n,k-2} = \text{Hank}_n (c_3, c_4, \ldots, c_k, 0, \ldots, 0).$$

The key to applying the updating procedure described above to (2.12) is in expressing the matrix $H_{n,k-2}$ as a sum of rank-one updates of the form $\rho z z^T$. Note that $H_{n,k-2}$, which has the form

$$H_{n,k-2} = \begin{bmatrix} a_3 & a_4 & \cdots & a_k & & \\ a_4 & a_5 & \iddots & & & \\ & & a_k & & & a_k \\ \vdots & \iddots & 0 & & \iddots & \\ a_k & & & a_k & & \vdots \\ & & \iddots & & a_5 & a_4 \\ & & a_k & \cdots & a_4 & a_3 \end{bmatrix}, \tag{2.13}$$

will be quite sparse for values of k that are small with respect to n. We also note that $H_{n,k-2}$ is a block diagonal matrix if we assume that $k \leqq n/2$. A logical way of expressing $H_{n,k-2}$ as a sum of matrices of the form $\rho z z^T$ is to calculate the eigensystem for the two small $k \times k$ blocks appearing in the upper left and lower right corners of (2.13). We use the EISPACK routines to solve this problem in $O(k^3)$ steps.

We express the complete process formally as Algorithm 2.1.

ALGORITHM 2.1.

(1) Form

$$v = [a_1 - a_3, a_2 - a_4, \ldots, a_{k-2} - a_k, a_{k-1}, a_k, 0, \ldots, 0]^T.$$

(2) Solve

$$Sc = v$$

for the vector $c \in \mathbf{R}^n$, where $S = [s_{ij}]_{i,j=1}^n$ such that

$$\begin{aligned} s_{ij} &= s_{i-1,j-1} + s_{i+1,j-1}, && 1 \leqq i \leqq j \leqq n, \\ s_{ij} &= 0, && 0 \leqq j < i \leqq n, \\ s_{0,j} &= 0, && 1 \leqq j \leqq n, \\ s_{0,0} &= 1. \end{aligned}$$

(3) Form the matrix Λ of eigenvalues of the matrix $\text{CrossSum}_n (v)$ as follows:

$$\Lambda = \text{diag} (\lambda_1, \lambda_2, \ldots, \lambda_n),$$

where

$$\lambda_i = \sum_{j=0}^{n-1} \left[2 \cos \left(\frac{i\pi}{n+1} \right) \right]^j \cdot c_{j+1} \qquad i = 1, 2, \ldots, n.$$

The eigenvectors of the cross-sum matrix are then the columns of the matrix $U = [u_{ij}]$ given by

$$u_{ij} = \left(\frac{2}{n+1}\right)^{1/2} \sin\left(\frac{\pi ij}{n+1}\right)$$

for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$.

(4) Set $M = U^T$ and $D = \Lambda$.

(5) Calculate the eigensystem of the $k \times k$ matrix

$$B_1 = \begin{bmatrix} a_3 & a_4 & \cdots & a_k \\ a_4 & a_5 & & \iddots \\ & & a_k & \\ \vdots & \iddots & & \\ a_k & & & \end{bmatrix}.$$

So we have B_1 expressed as

$$B_1 = \sum_{i=1}^{k} \lambda_i q_i q_i^T.$$

Then the lower right corner of $H_{n,k-2}$ given by

$$B_2 = \begin{bmatrix} & & & a_k \\ & & \iddots & \vdots \\ & a_k & & \\ \iddots & & a_5 & a_4 \\ a_k & \cdots & a_4 & a_3 \end{bmatrix},$$

and we have

$$B_2 = \sum_{i=1}^{k} \lambda_i (q_i)^R (q_i^T)^R$$

as B_2 is related to B_1 by a permutation.

(6) For $j = 1$ to $2k$:

(a) Form $z_j = (q_j, 0, 0, \ldots, 0) \in \mathbf{R}^n$ if $j \leqq k$ or $z_j = (0, 0, \ldots, 0, q_i^R) \in \mathbf{R}^n$ if $j > k$.

(b) Form $v_j = Mz_j$.

(c) Calculate the eigensystem $\bar{Q}_j \bar{D}_j \bar{Q}_j^T$ of

$$D + v_j v_j^T$$

by the updating procedure.

(d) Update the eigenvector matrix M, which equals $\bar{Q}_j^T M$.

(e) Set $D = D_j$.

(7) The eigensystem of $T_{n,k}$ is then given by

$$M^T D M = T_{n,k}.$$

We have not as yet discussed the stability of the updating procedure used in Algorithm 2.1. Barlow [4] provides a backward error analysis of Cuppen's method for solving the symmetric tridiagonal eigenproblem based on using the implementation by Dongarra and Sorensen with some additional assumptions made about the details of the implementation. The analysis shows the method to be stable in the classical sense. Barlow extends these results to general updating strategies for the symmetric eigenvalue problem

and particularly to the technique presented in this paper. He obtains the following bound: Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the exact eigenvalues of a BST matrix $T_{n,k}$, and let $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_n$ be the computed eigenvalues obtained for $T_{n,k}$, then we have

$$\sqrt{\sum_{i=1}^{n} (\lambda_i - \hat{\lambda}_i)^2} \leqq [n^{1/2}\mu + 2(n+1)^{1/2}u] \cdot \|T_{n,k}\|_F + O(u^2 + \mu^2),$$

where u is the machine unit and μ is the tolerance for the iteration. In general, μ should be approximately equal to the machine unit. The numerical results in § 3 support these conclusions.

3. Numerical results and conclusions. We compare the performance of Algorithm 2.1 with that of the EISPACK routines BANDR and TQL2 that calculate the complete set of eigenvalues and eigenvectors of a real, symmetric, banded matrix. For k small with respect to n, Algorithm 2.1 compares very favorably to the EISPACK routines. Our tests were run on a Sun 4 under the UNIX operating system. All code was written in FORTRAN. The updating routine was obtained from D. Sorensen.

First we generated a random set of 50 banded Toeplitz matrices with dimension between 8 and 100, bandwidth between 3 and 7, and matrix entries in the range (−1000.0, 1000.0). For this set of problems, the average error between the answer obtained by EISPACK and the answer obtained by the technique presented here was 2.52E−11. The average time required by BANDR and TQL2 was 7.45 seconds. The average time required by the update method was 4.90 seconds.

Next, we generated a set of test cases that were very sparse. In most cases, the only nonzero entries were in the main diagonal and the kth subdiagonal and superdiagonal. There were 25 problems in this set with dimension between 8 and 100 and bandwidth between 3 and 7. The matrix entries were in the range (−100.0, 100.0). For this set, the average error was 9.20E−13. The average time for the BANDR and TQL2 routines was 3.50 seconds and the average time used by the update method was 1.97 seconds.

We now consider some small examples showing how performance of the algorithms changes as k becomes small with respect to n. All times in the examples are given in seconds. Each ratio entry is the ratio of the time required by the updating method to the time required by the EISPACK routines. The error column contains the two-norm of the absolute error between the vectors of eigenvalues obtained by the two methods.

Example 3.1. $T_{n,k} = \text{Toep}_n(10.0, 5.0, 6.0, 0.0, \ldots, 0.0)$.

Dim	k	Time for BANDR/TQL2	Time for UPDATE	Ratio	Error
10	3	0.04	0.03	0.75	1.83E−14
20	3	0.20	0.11	0.55	6.68E−14
30	3	0.65	0.24	0.37	6.81E−14
40	3	1.47	0.49	0.33	7.52E−14

Example 3.2. $T_{n,k} = \text{Toep}_n(-1.0, 2.0, -3.0, 4.0, 0.0, \ldots, 0.0)$.

Dim	k	Time for BANDR/TQL2	Time for UPDATE	Ratio	Error
10	4	0.03	0.06	2.00	2.91E−14
14	4	0.10	0.08	0.80	2.49E−14
18	4	0.17	0.15	0.88	4.79E−14
22	4	0.30	0.24	0.80	8.02E−14
26	4	0.47	0.34	0.72	3.48E−14
30	4	0.73	0.47	0.64	4.46E−14

Example 3.3. $T_{n,k} = \text{Toep}_n(20.0, 1.0, 3.0, 5.0, 7.0, 9.0, 0.0, \ldots, 0.0)$.

Dim	k	Time for BANDR/TQL2	Time for UPDATE	Ratio	Error
20	6	0.26	0.35	1.35	1.18E−13
40	6	1.66	1.62	0.98	1.99E−13
60	6	5.19	4.46	0.86	3.22E−13
80	6	11.97	9.40	0.79	4.02E−13
100	6	22.15	17.79	0.80	6.19E−13
120	6	39.45	31.28	0.79	5.54E−13
140	6	62.61	49.45	0.79	7.47E−13
160	6	90.88	72.65	0.80	7.92E−13
180	6	129.39	101.53	0.79	8.00E−13
200	6	179.44	137.81	0.77	1.15E−12

Example 3.4. $T_{n,k} = \text{Toep}_n(13.0, 0.0, 0.0, 0.0, -6.0, 0.0, \ldots, 0.0)$.

Dim	k	Time for BANDR/TQL2	Time for UPDATE	Ratio	Error
20	5	0.06	0.08	1.33	3.04E−14
40	5	0.56	0.34	0.61	5.08E−14
60	5	1.78	0.85	0.48	1.09E−13
80	5	3.87	1.73	0.45	7.63E−14
100	5	7.87	2.98	0.38	1.24E−13

Finally, we give a worst-case complexity analysis of Algorithm 2.1. Assuming that no deflation in update size occurs, we obtain the following operation counts.

Step	Operation Count
1	n
2	$n^2/2$
3	$4n^2$
4	$n^2 + n$
5	$5k^3$
6	$2(k-2)9n^2$

The total operation count is approximately $18kn^2$. It is clear that k must be rather small with respect to n for Algorithm 2.1 to compete with the EISPACK routines. If deflation occurs, it will further reduce the operation count for calculating the update's eigenvalues.

4. Conclusions and open questions. In this section, we summarize the previous results and examine areas in which further work can be done. The Bini and Capovani splitting proved quite useful in developing a numerical method for calculating the eigensystem of BST matrices. The new method was based on a divide-and-conquer method developed by Cuppen [8] and further studied by Dongarra and Sorensen [9]. The Cuppen method employs a procedure due to Bunch, Nielsen, and Sorensen [7] that calculates the eigensystem of a matrix obtained from adding a rank-one update to a diagonal matrix. The method we derived is quite fast for BST matrices whose bandwidth is small relative to their dimension. Other methods exist for calculating eigenvalues of this type of update. For example, two different techniques are discussed in Chapter 12 of Golub and Van Loan [10]. These updating methods may yield faster performance or improved accuracy and should be explored.

It may also be possible to apply the divide-and-conquer algorithms of Cuppen to other types of eigenvalue problems. Barlow and Demmel [3] obtained error bounds on the computed solutions to certain matrix pencil problems that are much better than those derived from standard algorithms. They have shown that algorithms exist that achieve the relative accuracy promised by these bounds. One example of such an algorithm is bisection followed by inverse iteration. It may be possible to implement the divide and conquer approaches used here in such a way as to satisfy these new error bounds as well.

A number of problems in this area have been suggested by Arbenz and Golub [1]. Further research in this direction should prove to be quite interesting.

Acknowledgments. The authors would like to thank Jim Demmel for some useful insights and Danny Sorensen for lending us the software for the updating method in [9]. The authors would also like to thank the referees whose suggestions greatly improved the paper.

REFERENCES

[1] P. ARBENZ AND G. H. GOLUB, *On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[2] P. ARBENZ, *Computing eigenvalues of banded symmetric Toeplitz matrices*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 743–754.

[3] J. BARLOW AND J. DEMMEL, *Computing accurate eigensystems of scaled diagonally dominant matrices*, SIAM J. Numer. Anal., 27 (1990), pp. 762–791.

[4] J. L. BARLOW, *Error Analysis of Update Methods for the Symmetric Eigenvalue Problem*, Tech. Report CS-91-09, Dept. of Computer Science, The Pennsylvania State University, University Park, PA, 1991; SIAM J. Matrix Anal. Appl., 14 (1993), pp. 598–618.

[5] D. BINI AND M. CAPOVANI, *Spectral and computational properties of band symmetric Toeplitz matrices*, Linear Algebra Appl., 52/53 (1983), pp. 98–126.

[6] D. BINI AND V. PAN, *Efficient algorithms for the evaluation of the eigenvalues of (block) banded Toeplitz matrices*, Math. Comp., 50 (1988), pp. 431–448.

[7] J. R. BUNCH, C. P. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[8] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

[9] J. J. DONGARRA AND D. C. SORENSEN, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. s139–s154.

[10] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[11] D. C. SORENSEN AND P. T. TANG, *On the orthogonality of eigenvectors computed by the divide-and-conquer techniques*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.

[12] W. TRENCH, *On the eigenvalue problem for Toeplitz band matrices*, Linear Algebra Appl., 64 (1985), pp. 199–214.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 215–218, January 1994

018

A NOTE ON JACOBI BEING MORE ACCURATE THAN *QR**

WALTER F. MASCARENHAS†

Abstract. In [*SIAM J. Matrix Anal. Appl.*, 13 (1992) pp. 1204–1245], Demmel and Veselić present a theoretical and experimental analysis to show that the Jacobi method is more accurate than the *QR* method when computing the eigenvalues of positive definite matrices. They show that the error caused by the Jacobi method depends on the size of a factor ρ, which is related to the singular values of certain matrices associated with the Jacobi iterates. Their experiments suggest that $\rho = O(1)$. However, in this note a family of matrices and orderings is presented for which $\rho = O(N)$, where N is the dimension of the matrix.

Key words. eigenvalues, Jacobi method, accuracy

AMS subject classifications. 65F15, 65G05, 65H15

1. Introduction. In [DV], Demmel and Veselić present strong evidence that the Jacobi method for computing eigenvalues of symmetric matrices is optimally accurate when applied to a positive definite matrix. Their work starts by showing that if H is positive definite and P is such that $|P_{ij}| \leqq \eta |H_{ij}|$, then

$$\max_i \frac{|\lambda_i(H) - \lambda_i(H+P)|}{\lambda_i(H)} \leqq \frac{N\eta}{\sigma(H)}, \tag{1.1}$$

where λ_i denotes the ith eigenvalue and $\sigma(H)$ is the smallest singular value of the matrix $A = D^{-1}HD^{-1}$, where D is a diagonal matrix and $A_{ii} = 1$. We note that although Demmel and Veselić use the condition number A instead of $1/\sigma(H)$ in (1.1), a more careful look at their analysis shows that the sharper bound in (1.1) holds.

The bound in (1.1) is better than what can be expected from the usual perturbation theory. The usual analysis leads to a bound of the form (1.1), but with $\sigma(H)$ replaced by the smallest singular value of H, which can be much smaller than $\sigma(H)$.

The bound (1.1) is sharp, in the sense that there exists P for which equality is almost achieved in (1.1). Therefore, the best accuracy we can hope for when using a finite arithmetic with rounding ε is obtained by replacing η by ε in (1.1).

To show that the Jacobi method has optimal accuracy, Demmel and Veselić obtain a bound

$$\max_i \frac{|\lambda_i(H) - \bar{\lambda}_i(H)|}{\lambda_i(H)} \leqq \frac{cMN\varepsilon}{\theta(H, O)}, \tag{1.2}$$

for the relative error in the eigenvalues $\bar{\lambda}$ computed by the Jacobi method using an ordering O, i.e., the order in which the rotations are performed. In this last equation c is a constant, M is the number of rotations performed until convergence, and

$$\theta(H, O) = \min_k \sigma(H^{(k)}), \tag{1.3}$$

where $H^{(k)}$ are the iterates produced by the Jacobi method with ordering O to H.

Therefore, the ratio

$$\rho(H, O) = \frac{\sigma(H)}{\theta(H, O)}$$

* Received by the editors March 19, 1992; accepted for publication June 29, 1992.

† Institute for Mathematics and Its Applications, University of Minnesota, Minneapolis, Minnesota 55455 (walterm@dcc.unicamp.br).

gives an idea of how the accuracy obtained by the Jacobi method compares to the best accuracy we can expect to obtain in finite arithmetic. Demmel and Veselić conjecture that ρ is small. In fact, their numerical experiments suggest that $\rho = O(1)$ for the ordering by rows. The purpose of this note is to present matrices and orderings for which $\rho = O(N)$, where N is the dimension of the matrix. More formally, at the end of the next section we prove the following result.

THEOREM 1. *Given $N = 2^n$ and α, $0 < \alpha < \frac{1}{2}$, there exists an $N \times N$ symmetric positive definite matrix $H = H(N, \alpha)$ and a family of orderings O such that $\rho(H, O) \geqq \frac{N}{4}$.*

We emphasize, however, that a ρ of order N is not enough to invalidate the claims in [DV]. If ρ can be at most $O(N)$, then Jacobi is more accurate than QR. Our search for matrices and orderings with bigger ρ's was unsuccessful and leads us to believe that the growth above is maximal. As it was the case with Gaussian elimination, the use of clever optimization techniques can lead to bigger values of ρ. Even if such examples are found, they are likely to be complicated, and we believe that our example gives a nice and simple explanation of how ρ can grow.

Finally, in the appendix we present a very short Matlab program that simulates the behavior of the Jacobi method for the orderings and matrices from Theorem 1. The experiments show that the maximum relative error in this case is asymptotically equal to $\frac{N\varepsilon}{2}$, where N is the dimension of the matrix and ε is machine precision.

2. Description of H and O. We start this section by presenting the family $H(N, \alpha)$ and the orderings O. Then we present a lemma and finally a proof of Theorem 1.

The matrices $H = H(N, \alpha)$ are rather simple. They have ones on the diagonal and $1 - \alpha$ in all the other entries. In other words

$$H_{ii} = 1, \qquad H_{ij} = 1 - \alpha \text{ if } i \neq j.$$

These H are positive definite and $\lambda_{\min}(H) = \alpha$ if $\alpha < 1$.

We will say that an ordering O for applying the Jacobi method is *acceptable* if it can be obtained by the following recursive procedure:

- if $N = 2$, then pivot $(1, 2)$.
- If $N = 2^n$, $n > 1$, partition the matrix as

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{pmatrix},$$

where the H_{ij} are $2^{n-1} \times 2^{n-1}$ blocks, pivot the entries on the main diagonal of H_{12}, then rotate the remaining entries of H_{12} and the entries in H_{11} in any ordering. Finally, apply an acceptable ordering to H_{22}.

The iterates of the Jacobi method applied according to an acceptable ordering to the matrices $H(N, \alpha)$ above can be simply described, provided we assume that in the ambiguous case of repeated diagonal entries we choose the Jacobi rotation J as did, for example, Golub and Van Loan, that leads to the example below

$$J\begin{pmatrix} 1 & b \\ b & 1 \end{pmatrix}J^T = \begin{pmatrix} 1-b & 0 \\ 0 & 1+b \end{pmatrix}. \tag{2.1}$$

Let us call $\tilde{H}$ the matrix obtained after applying the rotations in the $\frac{N}{2} \times \frac{N}{2}$ blocks H_{12} and H_{11} of H. We now have the following lemma.

LEMMA 1. *If the ordering O is acceptable and, in the case of repeated diagonal entries, we rotate as in* (2.1), *then*

$$\tilde{H}(N, \alpha) = \begin{pmatrix} \alpha I_{N/2} & 0 \\ 0 & (2-\alpha)H(\frac{N}{2}, \frac{\alpha}{2-\alpha}) \end{pmatrix},$$

where I_k is the $k \times k$ identity matrix.

Proof of Lemma 1. We analyze here the case $N = 4$. The general case is analogous and is left to the reader. In this case we have

$$H = H(N, \alpha) = \begin{pmatrix} 1 & 1-\alpha & 1-\alpha & 1-\alpha \\ & 1 & 1-\alpha & 1-\alpha \\ & & 1 & 1-\alpha \\ & & & 1 \end{pmatrix}.$$

The first rotation is at (1, 3) and leads to the matrix

$$H^{(1)} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ & 1 & \frac{\sqrt{2}(1-\alpha)}{2} & 1-\alpha \\ & & 2-\alpha & \frac{\sqrt{2}(1-\alpha)}{2} \\ & & & 1 \end{pmatrix}.$$

Then we pivot at (2, 4), getting

$$H^{(2)} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ & \alpha & 0 & 0 \\ & & 2-\alpha & 2(1-\alpha) \\ & & & 2-\alpha \end{pmatrix}.$$

The rest of the pivots in the block H_{12}, (1, 4) and (2, 3), are zero, so we do not rotate. The only pivot in the block H_{11}, (1, 2), is also zero and again no rotation is performed. To complete the proof, note that

$$\begin{pmatrix} 2-\alpha & 2(1-\alpha) \\ & 2-\alpha \end{pmatrix} = (2-\alpha)\begin{pmatrix} 1 & \frac{2(1-\alpha)}{2-\alpha} \\ & 1 \end{pmatrix} = (2-\alpha)\begin{pmatrix} 1 & 1-\frac{\alpha}{2-\alpha} \\ & 1 \end{pmatrix}.$$

Proof of Theorem 1. Take O to be an acceptable ordering. Lemma 1 shows that applying the Jacobi method to $H(N, \alpha)$ according to O reduces to applying the Jacobi method to $(2-\alpha)H(\frac{N}{2}, \frac{\alpha}{2-\alpha})$, according to another acceptable ordering O'. Since the Jacobi method, and the smallest singular value of $A = D^{-1}HD^{-1}$, are scale invariant, this implies that

$$\theta(H(N, \alpha), 0) \leqq \theta\left(H\left(\frac{N}{2}, \frac{\alpha}{2-\alpha}\right), O'\right). \tag{2.2}$$

Therefore, if α_k is given by the recurrence relation

$$\alpha_0 = \alpha, \tag{2.3}$$

$$\alpha_{k+1} = \frac{\alpha_k}{2-\alpha_k}, \tag{2.4}$$

we have

$$\theta(H(N, a), O) \leqq \theta\left(H\left(\frac{N}{2^k}, \alpha_k\right), O'\right) \leqq \theta(H(2, \alpha_{\log N-1}), O'') = \alpha_{\log N-1}.$$

Fortunately, the recurrence relation (23) has a simple closed form solution:

$$\alpha_k = \frac{\alpha}{2^k(1-\alpha)+\alpha}.$$

Thus,

$$\theta(H(N,\alpha), O) = \frac{\alpha}{2^{\log N-1}(1-\alpha)+\alpha} = \frac{\alpha}{\frac{N}{2}(1-\alpha)+\alpha}.$$

Since we are assuming $0 < \alpha < \frac{1}{2}$, this implies that

$$\theta(H(N,\alpha), O) \leqq \frac{4\alpha}{N}.$$

Therefore, since $\sigma(H(N,\alpha)) = \alpha$,

$$\rho(H(N,\alpha), O) \geqq \frac{N}{4},$$

and the proof of Theorem 1 is complete.

3. Appendix. In this appendix we present the Matlab code to simulate the operations realized by the Jacobi method when applied to the matrices H and orderings O in Theorem 1. Given the special structure of these matrices and orderings, the operations performed by the Jacobi method can be described exactly by the Matlab program.

```
1 a=.5; d=1; ev=[]; % d = diagonal, ev = eigenvalues.
2 for i=1:n,        % N = 2^n is the dimension, to be given.
3   ev(i)=d-a;
4   d=d+a;
5   a=a*sqrt(.5)*sqrt(.5)*4; % or a = 2*a*(1+eps);
6 end
7   max(abs((ev-.5)./.5))/(eps*2^(n-1))
```

In this program

$$a = a_k = (1-\alpha_k)\prod_{i=0}^{k-1}(2-\alpha_i)$$

is the nonzero off-diagonal element of the matrices H, computed using the recursion in Lemma 1, and d is the diagonal entry of H.

The expression in line 5 of the program above is used to update a because the exact expression, `a = 2 a`, would be computed exactly in most computers, and we would not be able to see the effects of rounding. Line 5 is a good simulation of the rounding errors that will happen if we apply a usual implementation of the Jacobi method for computing the eigenvalues of H.

REFERENCES

[DV] J. DEMMEL AND K. VESELIĆ, *Jacobi's method is more accurate than QR*, in SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 219–227, January 1994

019

LARGE LEAST SQUARES PROBLEMS INVOLVING KRONECKER PRODUCTS*

DONALD W. FAUSETT† AND CHARLES T. FULTON†

Abstract. The general problem considered here is the least squares solution of $(A \otimes B)x = t$, where A and B are full rank, rectangular matrices, and $A \otimes B$ is the Kronecker product of A and B. Equations of this form arise in areas such as digital image and signal processing, photogrammetry, finite elements, and multidimensional approximation. An efficient method of solution is based on QR factorizations of the original matrices A and B. It is demonstrated how these factorizations can be used to obtain the Cholesky factorization of the least squares coefficient matrix without explicitly forming the normal equations. A similar approach based on singular value decomposition (SVD) factorizations also is indicated for the rank-deficient case.

Key words. Kronecker product, overdetermined least squares, QR factorization, SVD factorization, matrix algorithms

AMS subject classifications. 15A23, 65F05, 65F20, 65F30

1. Introduction.

In this paper we consider primarily the least squares problem of full rank

$$(A \otimes B)x = t, \tag{1.1}$$

where A and B are real matrices that are $m \times p$, $m > p$, and $n \times q$, $n > q$, respectively, with rank $(A) = p$, rank $(B) = q$, and where the Kronecker product [7], [12], [14] (also tensor or direct product) is defined by

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1p}B \\ a_{21}B & a_{22}B & \cdots & a_{2p}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mp}B \end{pmatrix}.$$

Technically this definition is for a right Kronecker product; the left Kronecker product would have the matrix A multiplied by elements of B in each block [17], [18]. Here we make use of the notation $\mathbb{R}^k$ for the usual Euclidean vector space over the real field and $M_{k,l}$ for the vector space of $k \times l$ matrices over the real field, with $\mathbb{R}$ denoting the real numbers. Thus we have $A \in M_{m,p}$, $B \in M_{n,q}$, $A \otimes B \in M_{mn,pq}$, $x \in \mathbb{R}^{pq}$, and $t \in \mathbb{R}^{mn}$.

Least squares problems involving Kronecker products of the type (1.1) arise in several areas of application including signal and image processing [18], photogrammetry [17], fast transform algorithms [1], multidimensional approximation [15], the Lyapunov approach to stability theory [7], circuits and systems [4], and stochastic matrices [2]. In image processing applications, particularly where data collected by satellites or spacecraft are involved, the size of the matrices A and B can be very large, resulting in huge systems of linear equations of the form (1.1) involving literally hundreds of thousands, or even millions, of variables. A problem of Heap and Lindler [10], for example, on the algebraic restoration of astronomical images involves a linear model that gives rise to a system on the order of 250,000 linear equations with 250,000 unknowns. Similar problems arise rather commonly in photogrammetric applications, cf. Rauhala [17].

* Received by the editors November 5, 1991; accepted for publication (in revised form) July 22, 1992. This work was supported by National Science Foundation grant ASC-8821626, New Technologies Program, Division of Advanced Scientific Computing.

† Florida Institute of Technology, Department of Applied Mathematics, 150 West University Boulevard, Melbourne, Florida 32901 (dfausett@zach.fit.edu, fulton@zach.fit.edu).

The most common approach to computing the least squares solution of (1.1) is to form the normal equations

$$(A \otimes B)^T(A \otimes B)x = (A \otimes B)^T t. \tag{1.2}$$

Then the least squares solution of (1.1) (assuming $A \otimes B$ has full rank) is the exact solution of (1.2). Using standard properties of the Kronecker product [2], [7], [11], [12], [16], [17], the exact solution of (1.2) may be obtained in the form

$$x = (A^+ \otimes B^+)t \tag{1.3}$$

or the equivalent matrix form

$$X = B^+T(A^+)^T, \tag{1.4}$$

where

$$A^+ = (A^TA)^{-1}A^T \tag{1.5}$$

and

$$B^+ = (B^TB)^{-1}B^T$$

are the Moore–Penrose pseudoinverses of A and B, and the matrices T and X are related to the vectors t and x by

$$X = (x^{(1)}, x^{(2)}, \ldots, x^{(p)}) = \begin{pmatrix} x_1 & x_{q+1} & \cdots & x_{(p-1)q+1} \\ \vdots & \vdots & & \vdots \\ x_q & x_{2q} & \cdots & x_{pq} \end{pmatrix} \tag{1.6}$$

and

$$T = (t^{(1)}, t^{(2)}, \ldots, t^{(m)}) = \begin{pmatrix} t_1 & t_{n+1} & \cdots & t_{(m-1)n+1} \\ \vdots & \vdots & & \vdots \\ t_n & t_{2n} & \cdots & t_{mn} \end{pmatrix}. \tag{1.7}$$

For the details leading from (1.2) to (1.4) we refer to Rao and Mitra [16], Horn and Johnson [12], or Rauhala [17].

Since the dimensions of the matrices A, B, X, T are small compared to the dimensions of $A \otimes B$, the operation counts for the solution (1.4) are an order of magnitude smaller than the operation counts required for the explicit formation and solution of (1.2). In fact, it is this tremendous reduction in the amount of work in using (1.4) vis-à-vis (1.2) (or (1.3)) that forms the basis for Rauhala's concept of "array algebra" [17] and its generalizations.

The main drawback of the solution formula (1.4) is the instability associated with the explicit formation of A^TA and B^TB (which squares the condition number, cf. [6]). In this paper we propose a new method for solving (1.1) which also completely avoids the explicit formation of $A \otimes B$, and at the same time ensures a numerically stable solution. It is based on forming QR decompositions of A and B, which, upon algebraic simplification, give rise to a block diagonal system consisting of p blocks of upper triangular square $q \times q$ systems of equations. This has the particularly nice feature of being intrinsically parallel since the $q \times q$ blocks are mutually independent. Parallel implementations of the solution method described below are currently in progress.

2. QR factorization of the Kronecker product. Suppose that the full-rank matrices A and B have each been QR decomposed with column pivoting [6], so that

$$BP_1 = Q_1R_1 = Q_1\begin{pmatrix} R^{(1)} \\ 0^{(1)} \end{pmatrix} \tag{2.1}$$

and

$$AP_2 = Q_2R_2 = Q_2\begin{pmatrix} R^{(2)} \\ 0^{(2)} \end{pmatrix}, \tag{2.2}$$

where

$Q_1 \in M_{n,n}$	and	$Q_2 \in M_{m,m}$	are orthogonal matrices,
$R_1 \in M_{n,q}$	and	$R_2 \in M_{m,p}$	are upper triangular matrices,
$R^{(1)} \in M_{q,q}$	and	$R^{(2)} \in M_{p,p}$	are square, upper triangular matrices,
$0^{(1)} \in M_{n-q,q}$	and	$0^{(2)} \in M_{m-p,p}$	are zero matrices,

and

$P_1 \in M_{q,q}$ and $P_2 \in M_{p,p}$ are permutation matrices arising from the column pivoting.

Then

$$R^{(1)} = \begin{pmatrix} r_{11}^{(1)} & r_{12}^{(1)} & \cdots & r_{1q}^{(1)} \\ 0 & r_{22}^{(1)} & \cdots & r_{2q}^{(1)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{q,q}^{(1)} \end{pmatrix} \text{ and } R^{(2)} = \begin{pmatrix} r_{11}^{(2)} & r_{12}^{(2)} & \cdots & r_{1p}^{(2)} \\ 0 & r_{22}^{(2)} & \cdots & r_{2p}^{(2)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{p,p}^{(2)} \end{pmatrix}, \tag{2.3}$$

where the diagonal elements of $R^{(1)}$ and $R^{(2)}$ are nonzero. The permutation matrices P_1 and P_2 are used to keep the diagonal elements as far away from zero as possible.

For the following development, we need two basic properties of Kronecker products [12].

(2.4) Property 1. $(A \otimes B)^T = A^T \otimes B^T$ for any two rectangular matrices.

(2.5) Property 2. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$,

where the matrices must be of proper dimensions to be conformable for the matrix multiplications indicated.

Also, the following two lemmas are readily established.

LEMMA 1. *The Kronecker product of two permutation matrices is a permutation matrix.*

LEMMA 2. *The Kronecker product of two orthogonal matrices is an orthogonal matrix.*

Our solution technique is based on the following two theorems.

THEOREM 1. *If A and B have the permuted* QR *factorizations* (2.1)–(2.2), *then $A \otimes B$ has the permuted* QR *factorization*

$$(AP_2) \otimes (BP_1) = [(Q_2 \otimes Q_1)P_3^T][P_3(R_2 \otimes R_1)], \tag{2.6}$$

where $P_3 \in M_{mn,mn}$ is the permutation matrix defined by the requirement

$$P_3(R_2 \otimes R_1) = \begin{pmatrix} R^{(2)} \otimes R^{(1)} \\ 0^{(3)} \end{pmatrix}, \tag{2.7}$$

with the zero matrix $0^{(3)} \in M_{mn-pq,pq}$.

Proof. By property (2.5) we have

$$(AP_2) \otimes (BP_1) = (Q_2R_2) \otimes (Q_1R_1)$$
$$= (Q_2 \otimes Q_1)(R_2 \times R_1).$$

By Lemma 2, $Q_2 \otimes Q_1$ is orthogonal. Now,

$$R_2 \otimes R_1 = \begin{pmatrix} R^{(2)} \\ 0^{(2)} \end{pmatrix} \otimes \begin{pmatrix} R^{(1)} \\ 0^{(1)} \end{pmatrix} = \begin{pmatrix} r_{11}^{(2)}R_1 & r_{12}^{(2)}R_1 & \cdots & r_{1p}^{(2)}R_1 \\ 0 & r_{22}^{(2)}R_1 & \cdots & r_{2p}^{(2)}R_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{pp}^{(2)}R_1 \\ 0 & & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & & \cdots & 0 \end{pmatrix}.$$

In each of the first p block rows, the last $n - q$ rows are zero because the last $n - q$ rows of R_1 are zero as indicated in (2.1). Therefore, $R_2 \otimes R_1$ does not have the required upper triangular form for a QR decomposition. But this defect has an easy remedy; we introduce a permutation matrix as in (2.7), which performs the necessary row interchanges to move the last $n - q$ rows of each block row below the first pq rows of the permuted matrix. Then, inserting $I = P_3^TP_3$ between the $Q_2 \otimes Q_1$ and $R_2 \otimes R_1$ factors yields (2.6), where $P_3(R_2 \otimes R_1)$ is an upper triangular matrix and $(Q_2 \otimes Q_1)P_3^T$ is an orthogonal matrix (being the product of two orthogonal matrices). □

THEOREM 2. *Let*

$$R = R^{(2)} \otimes R^{(1)} \tag{2.8}$$

be the $pq \times pq$ upper triangular part of $P_3(R_2 \otimes R_1)$ as defined in (2.7). *Then*

$$R^TR = [(AP_2) \otimes (BP_1)]^T[(AP_2) \otimes BP_1)];$$

i.e., R^TR is the Cholesky factorization of $[(AP_2) \otimes (BP_1)]^T[(AP_2) \otimes (BP_1)]$.

Proof. Using properties (2.4)–(2.5), Lemma 2, and Theorem 1, a straightforward calculation yields

$$[(AP_2) \otimes (BP_1)]^T[(AP_2) \otimes (BP_1)] = R^TR. \quad \square$$

3. Application to the least squares problem. We now make use of the results of the previous section to transform the rectangular least squares problem (1.1) into a square block upper triangular system of dimension $pq \times pq$ whose exact solution is the least squares solution of (1.1). This is the analog of the QR method for least squares for a standard overdetermined system (cf. [9, p. 224]).

Inserting $I = (P_2 \otimes P_1)(P_2 \otimes P_1)^T$ into (1.1) and using property (2.5), we have

$$(A \otimes B)(P_2 \otimes P_1)(P_2 \otimes P_1)^Tx = (AP_2 \otimes BP_1)(P_2^T \otimes P_1^T)x = t. \tag{3.1}$$

Multiplying on the left by $(AP_2 \otimes BP_1)^T$, and making use of the Cholesky factorization in Theorem 2, we obtain the normal equations associated with (3.1) as

$$R^TRy = (AP_2 \otimes BP_1)^Tt$$
$$= [P_3(R_2 \otimes R_1)]^T[P_3(Q_2^T \otimes Q_1^T)]t \tag{3.2}$$
$$= [R^T \vdots (0^{(3)})^T][P_3(Q_2^T \otimes Q_1^T)]t,$$

where

$$y = (P_2^T \otimes P_1^T)x \tag{3.3}$$

is a permutation of the vector x. Here the last two lines make use of (2.6)–(2.7). Since $R = R^{(2)} \otimes R^{(1)}$ is $pq \times pq$ and $[0^{(3)}]^T$ is $pq \times (mn - pq)$, it follows that the last $mn - pq$ components of the $mn \times 1$ vector $[P_3(Q_2^T \otimes Q_1^T)]t$ do not contribute to the right-hand side vector. We therefore define

$$[P_3(Q_2^T \otimes Q_1^T)t]_{pq} \in \mathbb{R}^{pq} \tag{3.3}$$

to be the vector containing the first pq components of $P_3(Q_2^T \otimes Q_1^T)t$. Then the normal equations become

$$R^T R y = R^T [P_3(Q_2^T \otimes Q_1^T)]_{pq}, \tag{3.4}$$

where R is the square matrix in (2.8). Since A and B were assumed full rank, all diagonal elements of $R^{(1)}$ and $R^{(2)}$ are nonzero, so it follows that R is nonsingular. Multiplying by $(R^T)^{-1}$ in (3.4) thus yields the block upper triangular system

$$Ry = \begin{pmatrix} r_{11}^{(2)} R^{(1)} & r_{12}^{(2)} R^{(1)} & \cdots & r_{1p}^{(2)} R^{(1)} \\ 0 & r_{22}^{(2)} R^{(1)} & \cdots & r_{2p}^{(2)} R^{(1)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{pp}^{(2)} R^{(1)} \end{pmatrix} \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(p)} \end{pmatrix} = \begin{pmatrix} h^{(1)} \\ h^{(2)} \\ \vdots \\ h^{(p)} \end{pmatrix} = h, \tag{3.5}$$

where

$$y = (P_2^T \otimes P_1^T)x = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(p)} \end{pmatrix}, \quad \text{with } y^{(i)} = \begin{pmatrix} y_1^{(i)} \\ y_2^{(i)} \\ \vdots \\ y_q^{(i)} \end{pmatrix} \quad \text{for } i = 1, \ldots, p; \tag{3.6}$$

and

$$h = [P_3(Q_2^T \otimes Q_1^T)t]_{pq} = \begin{pmatrix} h^{(1)} \\ h^{(2)} \\ \vdots \\ h^{(p)} \end{pmatrix}, \quad \text{with } h^{(i)} = \begin{pmatrix} h_1^{(i)} \\ h_2^{(i)} \\ \vdots \\ h_q^{(i)} \end{pmatrix} \quad \text{for } i = 1, \ldots, p. \tag{3.7}$$

Since $R^{(2)} \otimes R^{(1)}$ is upper triangular, (3.5) can be solved by block back substitution to obtain y. The least squares solution x to the original, unpermuted problem is then obtained by premultiplication by $P_2 \otimes P_1$:

$$(P_2 \otimes P_1)y = (P_2 \otimes P_1)(P_2^T \otimes P_1^T)x = (P_2 P_2^T) \otimes (P_1 P_1^T)x = x.$$

The right-hand-side vector h can be computed in a block manner as follows. Observe that

$$(Q_2^T \otimes Q_1^T)t = (Q_2^T \otimes I_1) \begin{pmatrix} Q_1^T t^{(1)} \\ Q_1^T t^{(2)} \\ \vdots \\ Q_1^T t^{(m)} \end{pmatrix}, \tag{3.8}$$

where the vector $t \in \mathbb{R}^{mn}$ has been partitioned into m subvectors, each of length n as indicated in (1.7). Recall that P_3 is the permutation matrix such that $P_3(R_2 \otimes R_1)$ is

upper triangular. In the right-hand-side vector h, P_3 performs the same interchanges on $(Q_2^T \otimes Q_1^T)t$. Therefore, the components of $h = [P_3(Q_2^T \otimes Q_1^T)t]_{pq}$ are precisely the same as those of $(Q_2^T \otimes Q_1^T)t$, which are located in rows in which $R_2 \otimes R_1$ has (some) nonzero elements. Since we know in advance in which rows those nonzero elements will appear, the permutation matrix P_3 need not be formed. For example, we can compute h by taking the first q components of

$$Q_1^T \sum_{k=1}^{m} q_{kj}^{(2)} t^{(k)} \quad \text{for } j = 1, \ldots, p.$$

The rows of $R_2 \otimes R_1$ that are zero, and hence are omitted from $R^{(2)} \otimes R^{(1)}$, along with the corresponding components of $(Q_2^T \otimes Q_1^T)t$ on the right-hand side, represent $p(n-q) + n(m-p)$ equations in the original linear system $(A \otimes B)x = t$. In general, those equations will not be satisfied by the least squares solution x. The omitted components of $(Q_2^T \otimes Q_1^T)t$ (as defined by (3.3)) constitute the "irreducible residual" of the least squares problem (cf. [5, p. 132]).

4. Reduction to block diagonal form. The upper block triangular system (3.5) can be brought into block diagonal form by computation of $(R^{(2)})^{-1}$. This can be done in parallel by back solving the triangular systems $R^{(2)}v_i = e_i$ for $i = 1, \ldots, p$ on different processors, where e_i is the ith column of the $p \times p$ identity matrix I_2. Since more computational effort is required to solve the triangular systems for larger values of i, $1 \le i \le p$, more of the systems corresponding to smaller values of i can be assigned to the same processor than for larger values. A scheme of this sort allows for the computation of $(R^{(2)})^{-1}$ with a reasonably well-balanced distribution of the computational load among the processors.

Once the inverse of $R^{(2)}$ has been obtained, the transformation to block diagonal form proceeds in the following manner. Starting from (3.5), we have

$$\begin{aligned} & (R^{(2)} \otimes R^{(1)})y = h, \\ \Leftrightarrow\ & (R^{(2)} I_2) \otimes (I_1 R^{(1)})y = h, \\ \Leftrightarrow\ & (R^{(2)} \otimes I_1)(I_2 \otimes R^{(1)})y = h, \end{aligned} \tag{4.1}$$

where I_1 is a $q \times q$ identity matrix.

Now $(R^{(2)} \otimes I_1)^{-1} = (R^{(2)})^{-1} \otimes I_1$, so we can multiply both sides of (4.1) by $(R^{(2)})^{-1} \otimes I_1$ to obtain

$$(I_2 \otimes R^{(1)})y = ((R^{(2)})^{-1} \otimes I_1)h \equiv \tilde{h}, \tag{4.2}$$

which has the block diagonal form

$$\begin{pmatrix} R^{(1)} & 0 & \cdots & 0 \\ 0 & R^{(1)} & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & R^{(1)} \end{pmatrix} \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(p)} \end{pmatrix} = \begin{pmatrix} \tilde{h}^{(1)} \\ \tilde{h}^{(2)} \\ \vdots \\ \tilde{h}^{(p)} \end{pmatrix}, \tag{4.3}$$

where

$$\tilde{h} = ((R^{(2)})^{-1} \otimes I_1)h. \tag{4.4}$$

The resulting block diagonal system (4.3) is perfectly parallel, so the upper triangular systems in each block row, $R^{(1)}y^{(j)} = \tilde{h}^{(j)}$, may be solved simultaneously. On a distributed memory computer the solution procedure for different blocks can be assigned to different processors and the computations performed independently. If the problem matrices are large, then each processor may have more than one block computation to perform.

5. Computational complexities. The QR decompositions of A and B require $2p^2(m - p/3)$ flops and $2q^2(n - q/3)$ flops, respectively, cf. [6]. The formation of h requires $p[n(2m - 1) + q(2n - 1)]$ flops. The computation of $(R^{(2)})^{-1}$ requires $p + p(p^2 - 1)/3$ flops; and the matrix multiplications to generate $\tilde{h}$ require p^2q flops.

Finally, back substitution for each block requires q^2 flops; therefore the total flops for all p blocks is pq^2. Thus the total work required to solve the least squares problem is $p^2[2(m - \frac{p}{3}) + q] + q^2[2(n - \frac{q}{3}) + p] + p[n(2m - 1) + q(2n - 1) + (p^2 - 1)/3 + 1] = O(p^2[2m + q - \frac{p}{3}] + q^2[2n + p - \frac{2q}{3}] + 2np(m + q))$ flops.

By way of comparison, the computational complexity for implementing the normal equations in the form (1.4), using Gaussian elimination for the inverses, is

$$p[4mp - (m + p)] + q[4nq - (n + q)] + 5(p^3 + q^3)/3 + p(p - 1)(2p - 1)/6 + q(q - 1)(2q - 1)/6 + mq(2n - 1) + pq(2m - 1) = O(2[p^2(2m + p) + q^2(2n + q) + mq(n + p)]) \text{ flops;}$$

while the computational complexity for implementing the solution formula (1.3) by explicitly forming $A^+ \otimes B^+$ is much greater, namely,

$$p[4mp - (m + p)] + q[4nq - (n + q)] + 5(p^3 + q^3)/3 + p(p - 1)(2p - 1)/6 + q(q - 1)(2q - 1)/6 + pm[q(2n - 1) + n] = O(2[p^2(2m + p) + q^2(2n + q) + mnp(q + \tfrac{1}{2})]).$$

For any $p > 0$ and $q > 0$, $p^3 + q^3 \geq pq(p + q)$. Therefore, the only term in the operations count for the QR approach that could possibly be larger than the corresponding term for (1.4) is the last one. Comparison of these terms shows that $2np[m + q] > 2mq[n + p]$ can occur only when $p > q$ or $n > m$ (or both). It would be very unusual for this term to dominate over the effects of the other terms in which the QR approach has the advantage. Thus the QR approach offers computational efficiency as well as computational stability.

6. SVD factorization of the Kronecker product. We briefly consider the case in which one or both of the matrices A and B is not full rank. In this case, the QR approach must be modified if it is to work at all so as to take account of rank (A) and rank (B) being smaller than p and q, respectively, and it therefore becomes necessary to compute the rank using a technique like that of Bischof [3], for example. Here we outline an alternative procedure using the SVD.

Assume that the matrices A and B already have been decomposed as in (2.1) and (2.2). It is computationally more efficient to proceed with complete orthogonal decompositions of R_1 and R_2 than it is to return to the original matrices A and B. Now we suppose that R_1 and R_2 have each been SVD decomposed [6], so that

$$R_1 = U_1 \Sigma_1 V_1^T \tag{6.1}$$

and

$$R_2 = U_2 \Sigma_2 V_2^T, \tag{6.2}$$

where $U_1 \in M_{n,n}$, $U_2 \in M_{m,m}$, $V_1 \in M_{q,q}$, and $V_2 \in M_{p,p}$ are orthogonal matrices, and $\Sigma_1 \in M_{n,p}$ and $\Sigma_2 \in M_{n,p}$ are diagonal matrices.

Then [12, p. 246]

$$R_2 \otimes R_1 = (U_2 \otimes U_1)(\Sigma_2 \otimes \Sigma_1)(V_2^T \otimes V_1^T); \tag{6.3}$$

and from (2.6) we have

$$(AP_2) \otimes (BP_1) = [(Q_1U_2) \otimes (Q_1U_1)](\Sigma_2 \otimes \Sigma_1)(V_2^T \otimes V_1^T), \tag{6.4}$$

which we premultiply on both sides by $[(Q_2U_2) \otimes (Q_1U_1)]^T$ to obtain

$$(\Sigma_2 \otimes \Sigma_1)(V_2^T \otimes V_1^T)[(P_2^T \otimes P_1^T)x] = [(U_2^TQ_2^T) \otimes (U_1^TQ_1^T)]t. \tag{6.5}$$

Let k_1 = rank (B) and k_2 = rank (A); and let $\zeta_i^{(1)}$, $i = 1, \ldots, k_1$; and $\zeta_i^{(2)}$, $i = 1, \ldots, k_2$ denote the first k_1 and k_2 diagonal elements of Σ_1 and Σ_2, respectively. None of those elements are zero, and all other elements of Σ_1 and Σ_2 are zero. Now we introduce the permutation matrices P_3 and P_4 such that $D = P_3(\Sigma_2 \otimes \Sigma_1)P_4$ is a diagonal matrix with its first k_1k_2 diagonal elements nonzero and the vector y defined by

$$y = P_4^T(V_2^T \otimes V_1^T)(P_2^T \otimes P_1^T)x.$$

Premultiplying both sides of (6.5) by P_3 gives

$$Dy = P_3(U_2^TQ_2^T) \otimes (U_1^TQ_1^T)t. \tag{6.6}$$

Finally, we introduce the vector

$$h = [P_3(U_2^TQ_2^T) \otimes (U_1^TQ_1^T)t]_{k_1k_2}$$

containing the first k_1k_2 components of the right-hand side of (6.6).

Since D is diagonal, (6.6) can be solved for each component independently of the minimum-norm least squares solution y:

$$y_j = \begin{cases} h_j/d_j & \text{for } j = 1, \ldots, k_1k_2, \\ 0 & \text{for } j > k_1k_2. \end{cases}$$

The least squares solution to the original problem is obtained from

$$x = (P_2 \otimes P_1)(V_2 \otimes V_1)P_4y.$$

For the usual rank deficient problem $Cx = b$, it is well known that the minimum-norm least squares solution may be represented by

$$x = C^+b. \tag{6.7}$$

In the case of the Kronecker product least squares problem (1.1), it can be shown [13, p. 474] that

$$(A \otimes B)^+ = A^+ \otimes B^+, \tag{6.8}$$

whether A and/or B are full rank or rank deficient. As a consequence an alternative approach, which can serve as a comparison for the algorithm of this paper as well as for the above-described SVD or QR methods for the rank-deficient case, would be to use schemes such as that of Greville ([8], [13, pp. 222–224]), a direct method, or Sen and Prabhu ([19], [13, pp. 247–254]), an iterative method, to compute A^+ and B^+, and then obtain the minimum-norm least squares solution as

$$x = (A^+ \otimes B^+)t. \tag{6.9}$$

This, of course, could also be written in the compact form of (1.4) to save operation counts even for rank-deficient problems.

7. Conclusion. We have presented an efficient method for the solution of large least squares problems involving Kronecker products, based on QR factorization of the matrices that occur in the Kronecker product. This method combines the desirable stability prop-

erties of the QR approach to least squares with a computational scheme that requires minimal computer storage. Since the coefficient matrix can be very large in some applications, this is an essential consideration.

Acknowledgments. The authors thank Professor James Ortega for suggesting this approach to the problem and Dr. Charles Romine for many helpful comments.

REFERENCES

[1] H. C. ANDREWS AND J. KANE, *Kronecker matrices, computer implementation, and generalized spectra*, J. Assoc. Comput. Mach., 17 (1970), pp. 260–268.

[2] R. BELLMAN, *Introduction to Matrix Analysis*, 2nd ed., McGraw-Hill, New York, 1970.

[3] C. H. BISCHOF, *A parallel* QR *factorization algorithm with controlled local pivoting*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 36–57.

[4] J. W. BREWER, *Kronecker products and matrix calculus in system theory*, IEEE Trans. Circuits and Systems, 25 (1978), pp. 772–781.

[5] G. H. GOLUB AND J. M. ORTEGA, *Scientific Computing and Differential Equations*, Academic Press, New York, 1991.

[6] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations,* 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[7] A. GRAHAM, *Kronecker Products and Matrix Calculus: with Applications*, Halsted Press, John Wiley and Sons, Inc., New York, 1981.

[8] T. N. E. GREVILLE, *The pseudo-inverse of a rectangular or singular matrix and its application to the solution of systems of linear equations*, SIAM Rev., 1 (1959), pp. 38–43.

[9] W. W. HAGER, *Applied Numerical Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[10] S. R. HEAP AND D. J. LINDLER, *Block iterative restoration of astronomical images with the massively parallel processor*, Proc. 1st Aerospace Symp. Massively Parallel Scientific Computation, Sept. 24–25,.1986, pp. 99–109.

[11] H. V. HENDERSON AND S. R. SEARLE, *The vec-permutation matrix, the vec operator and Kronecker products: A review*, Linear and Multilinear Algebra, 9 (1981), pp. 271–288.

[12] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, London, 1991.

[13] E. V. KRISHNAMURTHY AND S. K. SEN, *Numerical Algorithms, Computations in Science and Engineering*, Affiliated East-West Press PVT. LTD., New Delhi, 1986.

[14] J. M. ORTEGA, *Matrix Theory: A Second Course*, Plenum Press, New York, 1987.

[15] V. PEREYRA AND G. SCHERER, *Efficient computer manipulation of tensor products with applications to multidimensional approximation*, Math. Comp., 27 (1973), pp. 595–605.

[16] C. R. RAO AND S. K. MITRA, *Generalized Inverse of Matrices and its Applications*, Wiley, New York, 1971.

[17] U. A. RAUHALA, *Introduction to array algebra*, Photogrammetric Engrg. Remote Sensing, 46 (1980), pp. 177–192.

[18] P. A. REGALIA AND S. K. MITRA, *Kronecker products, unitary matrices and signal processing applications*, SIAM Rev., 31 (1989), pp. 586–613.

[19] S. K. SEN AND S. S. PRABHU, *Optimal iterative schemes for computing Moore–Penrose matrix inverse*, Internat. J. Systems Sci., 8 (1976), pp. 748–753.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 228–272, January 1994

020

A SHIFTED BLOCK LANCZOS ALGORITHM FOR SOLVING SPARSE SYMMETRIC GENERALIZED EIGENPROBLEMS*

ROGER G. GRIMES†, JOHN G. LEWIS†, AND HORST D. SIMON‡

Abstract. An "industrial strength" algorithm for solving sparse symmetric generalized eigenproblems is described. The algorithm has its foundations in known techniques in solving sparse symmetric eigenproblems, notably the spectral transformation of Ericsson and Ruhe and the block Lanczos algorithm. However, the combination of these two techniques is not trivial; there are many pitfalls awaiting the unwary implementor. The focus of this paper is on identifying those pitfalls and avoiding them, leading to a "bomb-proof" algorithm that can live as a black box eigensolver inside a large applications code. The code that results comprises a robust shift selection strategy and a block Lanczos algorithm that is a novel combination of new techniques and extensions of old techniques.

Key words. Lanczos algorithm, sparse eigenvalue problems, structural analysis, symmetric generalized eigenvalue problem, orthogonalization methods

AMS subject classifications. 65F15, 15A18, 65F50, 73K99

1. Introduction.

The Lanczos algorithm [22] is widely appreciated in the numerical analysis community [6]–[9], [14], [15], [17], [23], [29], [30], [32], [35], [37] as a very powerful tool for extracting some of the extreme eigenvalues of a real symmetric matrix H, i.e., to find the largest and/or smallest eigenvalues and vectors of the symmetric eigenvalue problem

$$Hx = \lambda x.$$

It is often believed that the algorithm can be used directly to find the eigenvalues at both ends of the spectrum (both largest and smallest in value). In fact, many applications result in eigenvalue distributions that only allow effectively extracting the eigenvalues at one end of the spectrum. Typical eigenvalue distributions in structural engineering vibration problems have small eigenvalues of order unity with separations $|\lambda_{i+1} - \lambda_i|$ also of order unity, apparently well separated. However, for physical reasons the largest eigenvalues of these problems are very large, say, $\mathcal{O}(10^{10})$. The convergence rates for the eigenvalues is determined by the relative separation $\frac{|\lambda_{i+1}-\lambda_i|}{|\lambda_n-\lambda_1|}$, $\mathcal{O}(10^{-10})$ for the smallest eigenvalues. We expect and find very slow convergence to the small eigenvalues, which are the eigenvalues of interest. The dependence of convergence on *relative* separation between eigenvalues is often ignored.

It is also often believed that the Lanczos algorithm can be applied to the generalized symmetric problem

$$Hx = \lambda Mx$$

by using the naive reduction to standard form [16], [32]: factor M into its Cholesky decomposition $M = LL^T$ and then solve the ordinary eigenproblem $L^{-1}HL^{-T}y =$

* Received by the editors May 23, 1988; accepted for publication (in revised form) March 18, 1992.

† Mathematics and Engineering Analysis, Research and Technology Division, Boeing Computer Services, Mail Stop 7L-22, P.O. Box 24346, Seattle, Washington 98124-0346 (rgrimes@espresso.rt.cs.boeing.com, jglewis@espresso.rt.cs.boeing.com).

‡ Numerical Aerodynamic Simulation (NAS) Systems Division, National Aeronautics and Space Administration Ames Research Center, Mail Stop TO45-1, Moffett Field, California 94035 (simon@nas.nasa.gov). The author is an employee of Computer Sciences Corporation. This work was funded in part through National Aeronautics and Space Administration contract NAS 2-12961.

λy. Suppose that we applied this algorithm to the *vibration* problem of structural engineering,

$$Kx = \lambda Mx, \tag{1}$$

where K is the stiffness matrix and M is the mass matrix. We would fail abysmally for three separate reasons:

- M is very often semidefinite—it may admit no Cholesky factorization.
- Even when M can be factored, the eigenvalues that are desired are often very badly separated.
- The eigenvectors x must be computed by a back transformation $x = L^{-T}y$. When it exists, L is usually poorly conditioned, which can lead to considerable numerical error in the back transformation.

When K is positive definite, the vibration problem can be addressed by applying the usual reduction to the reciprocal problem:

$$Kx = \lambda Mx \Leftrightarrow Mx = \frac{1}{\lambda}Kx \Leftrightarrow L^{-1}ML^{-T}y = \mu y, \tag{2}$$

where L is the Cholesky factor of K and $\mu = \frac{1}{\lambda}$. Often this is sufficient as a cure for the first two problems in (1), because the reciprocals of the eigenvalues are well separated. Eigenanalysis codes in structural engineering packages [24], [27] have been built upon this transformation. But this transformation is still inadequate when:

- the model has rigid body modes—K is positive semidefinite and has no Cholesky decomposition.
- a considerable number of eigenvalues are desired.
- the eigenvalues wanted are not the smallest eigenvalues.

Applications with these characteristics do arise. The stiffness matrix in aerospace applications often has a six-dimensional nullspace of rigid body modes. Detailed analyses of structures may require more than just a few eigenvalues and vectors. One of our test problems is an analysis of a nuclear reactor containment floor, where more than 200 eigenpairs were needed to adequately model the response of the structure to a simulated earthquake. Another problem we analyzed was a model of a large industrial ventilating fan mounted on a large concrete platform, for which we needed good approximations to the eigenvalues near the fan's rotational rate, eigenvalues that are in the interior of the spectrum.

There is a more elaborate transformation of the problem, the *spectral transformation* of Ericsson and Ruhe [14], which treats all of these difficulties. The spectral transformation is discussed in detail in §2, where we discuss an extension of the standard algorithm to buckling as well as to vibration problems. The general idea behind the spectral transformation comes from considering the shifted problem $(K-\sigma M)x = (\lambda-\sigma)Mx$. If we invert $(K-\sigma M)$, we transform the eigenvalues nearest the *shift* σ into the largest and well-separated eigenvalues of the reciprocal problem. Normally we need only to choose a shift σ near the eigenvalues we want. When the number of eigenvalues is large, the reduced convergence rate of the eigenvalues farthest from σ makes it worthwhile to choose additional shifts (and factorizations) in order to search through the spectrum.

Formally we cannot shift at an eigenvalue of the problem, because the shifted operator is singular. In fact, avoiding even near-singularity is an issue for the choice of shifts, particularly the very first shift, because shifts very close to eigenvalues are useful only for computing isolated clusters of eigenvalues.

In general, a well-chosen shift allows us to compute tens of eigenvalues with a single Lanczos run. There is a complicated tradeoff between the cost of a Lanczos run, which increases nonlinearly with increasing numbers of steps, and the cost of computing a new shift and its concomitant factorization. As an example, we consider the oceanography model (matrix PLAT1919 in the Harwell/Boeing sparse matrix collection [11]), with four different paradigms for choosing shifts:

- the heuristic described in this paper;
- a conservative modification of this heuristic;
- an aggressive modification of this heuristic;
- a fixed shift—compute all 200 eigenvalues with a single factorization.

All of these analyses begin with a Lanczos run using the factors of $A - .0001I$ to find the eigenvalues of $(A - .0001I)^{-1}$. Table 1 contains the salient results for these choices, demonstrating the complexity of the tradeoffs and, dramatically, the value of shifting.

TABLE 1
Computing the 200 *lowest eigenvalues in* [.0001, .24] *of* PLAT1919.

Choice of shift	Number of Lanczos runs	Total number of Lanczos steps	Execution cost
normal	8	192	208.1
conservative	13	243	257.4
aggressive	8	209	225.5
fixed shift	1	318	5382.2

(These results were obtained on a Sun 4/690 workstation. The code used a blocksize of three. Execution cost is the sum of central processor (cpu) and input/output (i/o) processor seconds.)

Shifting can provide reliability as well as efficiency. Each factorization provides eigenvalue location information in the form of *matrix inertias* (see §3.1). The collected inertias from a series of well-chosen shifts can provide an independent guarantee on the success of the eigenvalue computation and can be used to drive the choice of further shifts and Lanczos runs to ensure that all of the desired eigenvalues have been computed. Our heuristic strategy for choosing shifts is discussed in §3.

Our goal is a code that can serve as a "black-box" eigenextraction routine in large applications codes. Eigenvalues cannot be assumed to be simple, so our shifting strategy is prepared to continue looking at a small piece of the spectrum until it has determined the full multiplicity of the eigenvalues therein. The shifting scheme and the Lanczos algorithm interact to ensure that we find an orthogonal basis for the invariant subspace for each cluster (see §4.3.3). Most importantly, we use a *block* version of the Lanczos algorithm. The Lanczos algorithm usually will compute the full multiplicities of each cluster without any intervention from the shifting strategy, provided that we have been able to choose a blocksize as large as the largest multiplicity of any cluster we will encounter.

The block Lanczos algorithm also confronts the problem that applications codes often use general representations for their data, even when particular machine architectures would allow or favor alternatives. It is still common for general applications codes to represent their matrices as "out-of-core." The block Lanczos code substitutes, almost on a one-for-one basis, matrix-block multiplies and block solves for matrix-vector products and simple solves. This decreases the i/o cost essentially by the blocksize.

Our production eigenextraction code is a synthesis of the ideas of the spectral transformation and the block Lanczos algorithm. In §2 we begin to address the effects of the generalized problem on the recurrence. We explain what modifications to the Lanczos recurrence result from the use of shifted and inverted operators. With the exception of the development of a spectral transformation for buckling problems, our presentation is quite standard and is provided for the reader not already familiar with these results.

We present our heuristic shifting strategy in §3. There are eight subsections: a discussion of *trust intervals* and matrix inertias, our basic tools for robustness; our heuristic for choosing a shift in a generic case; the idea of *sentinels*, a tool for ensuring orthogonality of invariant subspaces; heuristics for choosing an initial shift; heuristics for determining how to expand the primary trust interval; analysis of a specified finite interval; treatment of various special and pathological cases; and, last, the modifications needed for the buckling problem.

The special characteristics of our block Lanczos algorithm are discussed in §4. This considers the effects due to the spectral transformation. One major problem is that vectors must be orthonormalized with respect to an inner product defined by a positive definite matrix M. We discuss the issues associated with implementing M-orthonormalization of vectors in the basic block Lanczos algorithm, including the further precautions needed to allow cases where M induces only a seminorm, in §4.1.

The block Lanczos recurrence by itself produces only a block tridiagonal matrix T. In §4.2 we describe how to compute eigenvalue and vector approximations, and error bounds on these approximations, from T and the Lanczos vectors. Section 4.3 contains our approach for dealing with the loss of orthogonality in the Lanczos vectors, with a novel combination of various reorthogonalization schemes that work effectively with the unusual distributions of eigenvalues that result from the spectral transformation. Section 4.4 concludes with discussions of when to end and how to start the recurrence. The integration of all of these techniques is a block Lanczos recurrence that will effectively find a limited number of eigenvalues and corresponding eigenvectors of a spectrally transformed operator.

We close with numerical experiments solving a small set of eigenproblems obtained from applications codes.

2. The spectral transformation block Lanczos algorithm. The eigenvalue problem in vibration analysis is given as

$$Kx = \lambda Mx, \tag{3}$$

where K and M are symmetric matrices, and M is positive semidefinite. Usually only the smallest eigenvalues of (3) are wanted, but they typically have very poor relative separation, rarely better than $\mathcal{O}(10^{-6})$. A priori estimates for the rate of convergence predict very slow convergence at the desired end of the spectrum. We can obtain rapid convergence to the desired eigenvalues by using the *spectral transformation* [14], [27] of (3).

2.1. The spectral transformation for vibration problems. Consider the problem

$$M(K - \sigma M)^{-1}Mx = \mu Mx, \tag{4}$$

where σ, the shift, is a real parameter. Assume for the moment that M is positive definite. It is easy to verify that (λ, x) is an eigenpair of (3) if and only if $(\frac{1}{\lambda-\sigma}, x)$ is

an eigenpair of (4). Hence, the transformation of the eigenvalue problem from (3) to (4) does not change the eigenvectors, and the eigenvalues are related by

$$\mu = \frac{1}{\lambda - \sigma}. \tag{5}$$

The form of the spectral transformation is dictated by our need to be able to apply the Lanczos algorithm even when M is semidefinite. Other advantages of this form are well documented in [38].

The main advantage of applying the Lanczos algorithm to (4) instead of to (3) becomes clear when the effect of the spectral transformation on the spectrum is considered. The results in Table 2 demonstrate this in detail. These are the values obtained using the initial shift described in §3.4; the generalized eigenproblem is the model of a nuclear reactor containment floor, given by the stiffness and mass matrices BCSSTK26 and BCSSTM26, respectively, from the Harwell–Boeing sparse matrix collection [11]. (We denote the generalized eigenproblem by BCSST_26.)

Relative separation is affected dramatically by the spectral transformation. The smallest eigenvalues are transformed into eigenvalues with good relative separation, even though their absolute separation is decreased. In addition, eigenvalues far from the shift are transformed to poorly separated values near zero. This spread of the eigenvalues ensures rapid convergence to the eigenvalues near σ. This example clearly demonstrates that the shift does not have to be very close in an absolute sense to work well.

TABLE 2
Vibration spectral transformation of BCSST_26, $\sigma_1 = 385.3$.

			Original		Transformed	
i	λ_i	μ_i	gap	relative gap	gap	relative gap
1	4.6×10^{3}	2.4×10^{-4}	6.4×10^{3}	1.2×10^{-11}	1.4×10^{-4}	6.0×10^{-1}
2	1.1×10^{4}	9.4×10^{-5}	2.5×10^{2}	4.6×10^{-13}	2.2×10^{-6}	9.2×10^{-3}
3	1.1×10^{4}	9.2×10^{-5}	2.5×10^{2}	4.6×10^{-13}	2.2×10^{-6}	9.2×10^{-3}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1920	3.0×10^{14}	3.3×10^{-15}	3.6×10^{11}	6.7×10^{-4}	3.9×10^{-18}	1.7×10^{-14}
1921	3.1×10^{14}	3.3×10^{-15}	3.6×10^{11}	6.7×10^{-4}	3.9×10^{-18}	1.7×10^{-14}
1922	5.4×10^{14}	1.8×10^{-15}	2.4×10^{14}	4.4×10^{-1}	1.4×10^{-15}	6.0×10^{-12}

The primary price for this rapid convergence is the cost of a factorization of $K - \sigma M$. The transformation $M(K - \sigma M)^{-1}M$ is realized implicitly as a sequence of operations in which we compute MQ for a block of vectors Q or solve the linear systems $(K - \sigma M)X = Q$. These operations are usually realized by independent subroutines, which allow tuning the matrix factorization and multiplication routines to the class of problem under consideration.

We must generalize the Lanczos algorithm itself to solve the transformed generalized symmetric eigenproblem. We make this generalization in three steps. We will first consider the ordinary block Lanczos algorithm for a symmetric matrix H. Next we consider a direct generalization of the Lanczos algorithm for an arbitrary generalized symmetric eigenproblem $Hx = \lambda Mx$, where we assume temporarily that M is positive definite. In these first two steps the issue of shifting disappears for the moment. In a third step we consider the much more effective form that results when H is a spectral transformation operator.

2.2. Basic block Lanczos algorithm. Consider first the ordinary eigenvalue problem

$$Hx = \lambda x,$$

where H is a real symmetric linear operator. An important characteristic of the Lanczos algorithm is that H is not required explicitly. All that is required is a subroutine that computes Hy for a given vector y. The block Lanczos iteration with *blocksize* p for an $n \times n$ matrix H is given in Fig. 1.

Initialization:

Set $Q_0 = 0$
Set $B_1 = 0$
Choose R_1 and orthonormalize the columns of R_1 to obtain Q_1

Lanczos Loop:

For $j = 1, 2, 3 \ldots$ do
Set $U_j = HQ_j - Q_{j-1}B_j^T$
Set $A_j = Q_j^T U_j$
Set $R_{j+1} = U_j - Q_j A_j$, the residual
Compute the orthogonal factorization $Q_{j+1}B_{j+1} = R_{j+1}$,
where B_{j+1} is upper triangular and Q_{j+1} is orthogonal
End loop

FIG. 1. *Basic block Lanczos algorithm.*

The matrices Q_j, U_j, R_j for $j = 1, 2, \ldots$ are $n \times p$, whereas A_j and B_j are $p \times p$, with A_j symmetric.

This formulation of the Lanczos loop is the one least susceptible to roundoff errors [31] and is the form that should be used in computation. In exact arithmetic, however, U_j and R_{j+1} can be eliminated from the Lanczos loop and the recurrence becomes

$$Q_{j+1}B_{j+1} = HQ_j - Q_j A_j - Q_{j-1}B_j^T. \tag{6}$$

This three-term recurrence simplifies theoretical discussion. It is shown in [6], [17] that the combined column vectors of the matrices $Q_1, Q_2, \ldots, Q_j$, the so-called *Lanczos vectors*, form an orthonormal set. The computational efficiency of the Lanczos algorithm rests on the fact that these vectors can be computed simply, with a fixed amount of work per iteration step.

The blocks of Lanczos vectors collectively form an $n \times jp$ matrix $\mathcal{Q}_j$, where

$$\mathcal{Q}_j = [Q_1, Q_2, Q_3, \ldots, Q_j].$$

The algorithm also defines a $jp \times jp$ block tridiagonal matrix T_j:

$$T_j = \begin{pmatrix} A_1 & B_2^T & 0 & \ldots & 0 \\ B_2 & A_2 & B_3^T & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & B_{j-1} & A_{j-1} & B_j^T \\ 0 & \ldots & 0 & B_j & A_j \end{pmatrix}.$$

Since the matrices B_j are upper triangular, T_j is a band matrix with half-bandwidth $p+1$ (rather than $2p$, if the B_j were full). The first j instances of formula (6) can be combined into a single formula:

$$HQ_j = Q_j T_j + Q_{j+1} B_{j+1} E_j^T. \tag{7}$$

Here E_j is an $n \times p$ matrix whose last $p \times p$ block is the $p \times p$ identity matrix and which is zero otherwise.

By premultiplying (7) by Q_j^T and using the orthogonality of the Lanczos vectors, we see that $Q_j^T H Q_j = T_j$. Hence T_j is the orthogonal projection of H onto the subspace span(Q_j) spanned by the columns of Q_j. It can be shown by induction that span(Q_j) = span$(Q_1, HQ_1, H^2Q_1, \ldots, H^{j-1}Q_1)$. From a different perspective, the (block) Lanczos algorithm is a method for constructing an orthonormal basis for the (block) Krylov subspace determined by H and Q_1. The orthogonal projection of H onto the (block) Krylov subspace is T_j. Hence the eigenvalues of T_j are the Rayleigh–Ritz approximations from span(Q_j) to the eigenvalues of H. In addition, if s is an eigenvector of T_j, the vector $y = Q_j s$ is an approximate eigenvector of H. Viewed in this form, the Lanczos algorithm replaces a large and difficult eigenvalue problem involving H by a small and easy eigenvalue problem involving the block tridiagonal matrix T_j.

How good are the approximations obtained by solving the block tridiagonal eigenvalue problem involving the matrix T_j? An a posteriori bound on the residual is given by Underwood [17]: Let θ, s be an eigenpair for T_j, i.e., $T_j s = s\theta$, and let $y = Q_j s$, then

$$\|Hy - y\theta\|_2 = \|B_{j+1} s_j\|_2, \tag{8}$$

where s_j are the last p components of the eigenvector s. The quantity $\|B_{j+1} s_j\|_2$ can be computed without computing the approximate eigenvector y. Hence, with some modifications described in §4.2, (8) provides an inexpensive a posteriori error bound.

Formula (8), however, does not guarantee that good approximations to eigenpairs will appear quickly. Such a priori estimates are provided by the Kaniel–Paige–Saad theory. Parlett [32] gives the most detailed discussion for the single vector case ($p = 1$). The generalizations to the block case were originally derived by Underwood [17]. Extensions to both of these presentations can be found in [36].

2.3. The spectral transformation block Lanczos algorithm. The next step is to consider the generalized symmetric eigenproblem $Hx = \lambda Mx$. Were we to reduce the problem to standard form by factoring M, the three-term recurrence (6) would become

$$Q_{j+1} B_{j+1} = M^{-1/2} H M^{-1/2} Q_j - Q_j A_j - Q_{j-1} B_j^T. \tag{9}$$

If we premultiply (9) by $M^{1/2}$ and make the transformation of variables $\hat{Q}_j = M^{-1/2} Q_j$, (9) becomes

$$\begin{aligned} M\hat{Q}_{j+1} B_{j+1} &= M^{1/2} M^{-1/2} H \hat{Q}_j - M\hat{Q}_j A_j - M\hat{Q}_{j-1} B_j^T \\ &= H\hat{Q}_j - M\hat{Q}_j A_j - M\hat{Q}_{j-1} B_j^T. \end{aligned} \tag{10}$$

The matrices $\hat{Q}_j$ are now M-orthogonal, since $Q_j^T Q_j = I$ implies $\hat{Q}_j^T M \hat{Q}_j = I$. This is also a property of the eigenvectors X of this generalized eigenproblem. The approximate eigenvectors will eventually be computed in the subspace span$(\hat{Q})$, regardless

For $j = 1, 2, 3 \ldots$ do
  Set $U_j \quad = H\hat{Q}_j - M\hat{Q}_{j-1}B_j^T$
  Set $A_j \quad = \hat{Q}_j^T M U_j$
  Set $W_{j+1} \quad = U_j - M\hat{Q}_j A_j$
  Solve $MR_{j+1} = W_{j+1}$
  Compute the M-orthogonal factorization $\hat{Q}_{j+1}B_{j+1} = R_{j+1}$
End loop

FIG. 2. *Inner loop of generalized symmetric block Lanczos algorithm.*

of the form used for the Lanczos recurrence. The inner loop of Lanczos recurrence in this subspace is given in Fig. 2.

The matrix M appears in several instances to assure the M-orthogonality of the Lanczos vectors. In particular, the last step requires computing the M-orthogonal factorization of R_{j+1}. Standard derivations of the orthogonality of the Lanczos vectors easily generalize to show that these vectors are M-orthonormal. It appears that $M^{-1/2}$ has disappeared from the standard recurrence, only to reappear at the penultimate step in disguise as a solution operation. Indeed, (10) applied to the original problem $Kx = \lambda Mx$ is merely an implicit form of the explicit reduction to standard form. This is not the case when H is taken as the operator in the spectral transformation. Substituting $M(K - \sigma M)^{-1}M$ for H gives:

$$M\hat{Q}_{j+1}B_{j+1} = M(K - \sigma M)^{-1}M\hat{Q}_j - M\hat{Q}_j A_j - M\hat{Q}_{j-1}B_j^T. \tag{11}$$

M now appears in *all* of the terms in the recurrence. Formally we can premultiply (11) by M^{-1} to obtain a recurrence

$$\hat{Q}_{j+1}B_{j+1} = (K - \sigma M)^{-1}M\hat{Q}_j - \hat{Q}_j A_j - \hat{Q}_{j-1}B_j^T \tag{12}$$

in which M^{-1} does not appear. This allows us to apply the same recurrence even when M is semidefinite. The justification for doing so appears later in §2.4.

At this point we shall no longer put "hats" on the matrices. The actual Lanczos recurrence for solving (4) is given in Fig. 3.

Assuming the matrix MQ_{j+1} is actually stored (at least temporarily), the algorithm as written requires only one multiplication by M per step and no factorization of M is required. The last step of the Lanczos loop, the *M-orthogonalization* of a set of p vectors, is discussed in §4.1.

Our next goal is to generalize the standard eigenvalue approximation results to the spectral transformation block Lanczos algorithm. As before, combining all j instances of (12) into one equation yields

$$(K - \sigma M)^{-1}MQ_j = Q_j T_j + Q_{j+1}B_{j+1}E_j^T, \tag{13}$$

where Q_j, T_j, and E_j are defined as in (7). Premultiplying (13) by $Q_j^T M$ and using the M-orthogonality of the Lanczos vectors, it follows that

$$Q_j^T M(K - \sigma M)^{-1}MQ_j = T_j.$$

Hence, T_j is the M-orthogonal projection of $(K - \sigma M)^{-1}$ onto the block Krylov subspace spanned by the columns of Q_j. The eigenvalues of T_j will approximate the

Initialization:

Set $Q_0 = 0$
Set $B_1 = 0$
Choose R_1 and orthonormalize the columns of R_1 to obtain Q_1 with $Q_1^T(MQ_1) = I_p$

Lanczos Loop:

For $j = 1, 2, 3 \ldots$ do
Set $U_j = (K - \sigma M)^{-1}(MQ_j) - Q_{j-1}B_j^T$
Set $A_j = U_j^T(MQ_j)$
Set $R_{j+1} = U_j - Q_jA_j$
Compute Q_{j+1} and (MQ_{j+1}) such that
a) $Q_{j+1}B_{j+1} = R_{j+1}$
b) $Q_{j+1}^T(MQ_{j+1}) = I_p$
End loop

FIG. 3. *Block Lanczos algorithm for the vibration problem.*

eigenvalues of (4). If (s, θ) is an eigenpair of T_j, i.e., $T_j s = s\theta$, then $(y = Q_j s, \nu = \sigma + \frac{1}{\theta})$ will be an approximate eigenpair of (3).

The generalization of the a posteriori residual bound (8) is

$$(K - \sigma M)^{-1}My - y\theta = Q_{j+1}B_{j+1}E_j^T s. \tag{14}$$

For $\theta \neq 0$ it follows that

$$(K - \nu M)y = -\frac{1}{\theta}(K - \sigma M)Q_{j+1}B_{j+1}E_j^T s.$$

The quantity on the right is computable without explicitly computing the eigenvector y, but only at the cost of a multiplication by $K - \sigma M$, which is not desirable. In §4.2 we present a better way to obtain a residual bound. (Note that $\theta = 0$ corresponds to an infinite eigenvalue of (3), which should not appear in T, as discussed below. Very small θ's correspond to eigenvalues far from the shift. These converge slowly—the division by θ in the residual bounds reflects their relative inaccuracy.)

2.4. Semidefiniteness in the matrix M. Throughout the discussion above, we assumed that M was a positive definite matrix. The formulation of the block Lanczos algorithm for the vibration problem does not require the factorization of M. Hence the spectral transformation Lanczos algorithm can be applied formally when M is semidefinite without further modifications. However, the eigenproblem (3) has infinite eigenvalues. Fortunately, we need only to make the obvious block modification of the analysis in [29] to remove the infinite eigenpairs from the recurrence. Following Nour-Omid et al., the starting block for the Lanczos algorithm should be computed as in Fig. 4.

The eigenvectors of $Kx = \lambda Mx$ corresponding to finite eigenvalues consist of a component orthogonal to the null vectors of M and a component in the nullspace of M. Ericsson [13] shows that the second, nullspace component is determined by an algebraic constraint from the non-nullspace component. The constraint expresses the

Choose	$\tilde{R}_1$
Compute	$R_1 = (K - \sigma M)^{-1} M \tilde{R}_1$
M-orthogonalize	$R_1 = Q_1 B_0$

FIG. 4. *Computation of the starting block.*

fact that all of these eigenvectors lie in the range of $(K - \sigma M)^{-1}M$. It is shown in [13], [29] that all of the Lanczos vectors lie in this subspace when the starting vectors are chosen in this subspace, as above. With this choice of starting block, infinite eigenvalues have no influence on the block Lanczos algorithm in exact arithmetic. In §4.2 we add a final postprocessing step to purge the approximate eigenvectors of components not satisfying the constraint in finite precision arithmetic.

2.5. A spectral transformation for buckling problems. The final point of this section is the spectral transformation for the buckling problem

$$Kx = \lambda K_\delta x, \tag{15}$$

where K is the symmetric positive semidefinite stiffness matrix and K_δ is the symmetric differential or geometric stiffness matrix. Typically only a few eigenvalues closest to zero are wanted. A simple approach would be to interchange the roles of K and K_δ and to compute the largest eigenvalues of the problem

$$K_\delta x = \mu K x, \tag{16}$$

with $\mu = \frac{1}{\lambda}$ by applying the simple Lanczos algorithm without shifts [21]. This reciprocal approach has the same drawbacks as (2). However, it is often effective when K is positive definite because the number of eigenvalues sought is rarely large.

Shifting and particularly the semidefinite K case require an alternative form of the spectral transformation [19]. The shifted and inverted problem

$$K(K - \sigma K_\delta)^{-1} K x = \mu K x \tag{17}$$

is solved instead of the original problem (15). The Lanczos recurrence is carried out using K-orthogonality among the Lanczos vectors. Each multiplication by the mass matrix M in the vibration case is replaced with a multiplication by the stiffness matrix K in the buckling case; the rest of the recurrence remains the same.

In the buckling spectral transformation (λ, x) is an eigenpair of (15) if and only if $(\frac{\lambda}{\lambda - \sigma}, x)$ is an eigenpair of (17). Hence the *buckling spectral transformation* does not change the eigenvectors, and the eigenvalues are related by $\mu = \frac{\lambda}{\lambda - \sigma}$. These results can be obtained directly, or by applying the vibration spectral transformation with reciprocated shifts to the reciprocal problem (16).

The advantages of the buckling spectral transformation are essentially the same as those of the vibration spectral transformation. Large eigenvalues of the buckling problem are transformed to a cluster of eigenvalues near unity. Eigenvalues near the shift σ are transformed into well-separated eigenvalues, which are easily computed by the Lanczos algorithm. The major difference is that a shift at $\sigma = 0$ is not allowed, since all eigenvalues would be transformed to one. This singularity in the transformation also affects shifts close to zero; very small shifts should not be taken in this form of the transformation. Table 3 gives details for the eigenproblem BCSST_28, treated as if it were a buckling problem. The initial shift is negative because we ordinarily

expect the first negative eigenvalue to be the eigenvalue of most interest in buckling problems (see §3.8). Just as in the case of the vibration spectral transformation, we see that the shift does not need to be close to the desired eigenvalues in any absolute sense. Indeed, in this case the shift is on the wrong side of the origin and yet still has the desired effect on relative separation.

TABLE 3
Buckling spectral transformation of BCSST_26, $\sigma_1 = -385.3$.

			Original		Transformed	
i	λ_i	μ_i	gap	relative gap	gap	relative gap
1	4.6×10^{3}	9.23×10^{-1}	6.4×10^{3}	1.2×10^{-11}	4.3×10^{-2}	5.6×10^{-1}
2	1.1×10^{4}	9.66×10^{-1}	2.5×10^{2}	4.6×10^{-13}	7.3×10^{-4}	9.5×10^{-3}
3	1.1×10^{4}	9.67×10^{-1}	2.5×10^{2}	4.6×10^{-13}	7.3×10^{-4}	9.5×10^{-3}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1920	3.0×10^{14}	1.00×10^{0}	3.6×10^{11}	6.7×10^{-4}	1.6×10^{-15}	2.0×10^{-14}
1921	3.1×10^{14}	1.00×10^{0}	3.6×10^{11}	6.7×10^{-4}	1.6×10^{-15}	2.0×10^{-14}
1922	5.4×10^{14}	1.00×10^{0}	2.4×10^{14}	4.4×10^{-1}	5.5×10^{-13}	7.1×10^{-12}

Except for the different role of the stiffness matrix K, all implementation details are the same for vibration and buckling analysis. Issues involving the M-orthogonality of the Lanczos vectors apply equally to the K-orthogonal Lanczos vectors in the buckling case. Since the stiffness matrix K is used in the initialization phase in the same way as M in the vibration case, the sequence of Lanczos vectors will be orthogonal to the space spanned by the eigenvectors corresponding to zero eigenvalues of K. Hence T_j will contain no approximations to the exactly zero eigenvalues of K, which are also zero eigenvalues of (15), which is desirable.

The eigenvalues of T_j approximate the eigenvalues of (17). Hence, if (s, θ) is an eigenpair of T_j, that is, $T_j s = s\theta$, then $(\frac{\sigma\theta}{\theta - 1}, Q_j s)$ is an approximate eigenpair of (15). The approximate eigenvectors form a K-orthonormal set. Bounds on the residuals of approximate eigenpairs are derived in §4.2.

3. A strategy for choosing shifts. Let us try to find some of the eigenvalues and eigenvectors of $KX = MX\Lambda$ or $KX = K_\delta X\Lambda$. We emphasize the fact that we want some, not all, of the eigenvalues, because the eigenvector matrix X is almost always dense. The problem can be written in its general form as:

- find the p eigenvalues of smallest magnitude in $[a, b]$ and their eigenvectors; or
- find the p eigenvalues of largest magnitude in $[a, b]$ and their eigenvectors; or
- find the p eigenvalues in $[a, b]$ closest to ξ and their eigenvectors; or
- find all eigenvalues and eigenvectors in $[a, b]$.

Here $[a, b]$ is the *computational interval*, which can be finite (both a and b finite), semi-infinite (only one of a and b finite), or infinite (no restrictions at all). Note that the problem of finding the algebraically least eigenvalues in an interval can be transformed into one of finding the eigenvalues of smallest magnitude by a suitable shift of origin.

The purpose of the spectral transformation is to transform the original problem into one whose dominant eigenvalues represent some of the desired eigenvalues. The dominant eigenvalues of the transformed problem correspond to the eigenvalues of the original problem nearest σ. There are two major goals that drive our strategy for choosing shifts. One is efficiency—we would like to choose a sequence of shifts $\sigma_1, \sigma_2, \ldots, \sigma_s$ so that the total cost, including the cost of the s factorizations and

the costs of the individual Lanczos runs, is minimized. Our heuristic approach to measuring and reducing cost is described in §§3.2 and 4.4. The second goal of our shift selection is robustness. A paramount objective for our design was a code that would be able to compute all of the desired eigenpairs accurately, except under extreme, pathological conditions. Furthermore, we wanted a code that could diagnose and report any failures. The tools we use to create robustness, trust intervals, and matrix inertias, are an appropriate place to begin the detailed discussion of our choices of shifts.

3.1. Trust intervals, matrix factorizations, and inertias. Suppose that during the course of eigenanalysis, we have computed a set of eigenvalues lying between two shifts σ_1 and σ_2. We would like to confirm that these are, in fact, all the eigenvalues in this interval.

Suppose that C is a real symmetric matrix, which has been decomposed as $C = LDL^T$, where D is diagonal. The *inertia* of C is the triple (π, ν, ζ) of integers, where π is the number of positive eigenvalues, ν the number of negative eigenvalues, and ζ the number of zero eigenvalues. Sylvester's Inertia Theorem [32, p. 10] states that the inertia of F^TCF is the same as that of C. Sylvester's theorem with $F = L^{-T}$ implies that the number of negative entries in D is the number of negative eigenvalues from C. The number of negative terms in D from the LDL^T decomposition of $C - \sigma I$ gives the number of eigenvalues smaller than σ. Frequently $\nu(C - \sigma I)$ is called the Sturm sequence number in engineering references.

It is easy to see that $\nu(C - \sigma_2 I) - \nu(C - \sigma_1 I)$ is the number of eigenvalues in the interval $[\sigma_1, \sigma_2]$ (assuming $\sigma_1 < \sigma_2$ and the two factorizations are nonsingular). When the number of eigenvalues expected in the interval agree with the number actually computed, we say that the interval $[\sigma_1, \sigma_2]$ is a *trust interval.* We want our shifting strategy to establish a trust interval around all of the desired eigenvalues.

However, applying these Sturm sequence results to generalized eigenproblems requires a transformation from the ordinary eigenvalue problem $CX = X\Lambda$ to the generalized problem $KX = MX\Lambda$. In order to guarantee that the generalized eigenvalue problems have real solutions, we assume that the pencils are definite; a positive definite linear combination of K and M must exist. In our code we assume that M or K is positive semidefinite. We compute $K - \sigma M = LDL^T$ (or $K - \sigma K_\delta = LDL^T$), and we want to draw conclusions from $\nu(LDL^T)$. The interpretation of $\nu(LDL^T)$ is given in Table 4; proofs are found in Appendix A. The major surprise in this table of the appearance of the null space dimension $\dim(\mathcal{N}(\cdot))$ when the matrix used as a norm is only a seminorm. This term corresponds to an assignment of signs to the infinite eigenvalues in the vibration case and the zero eigenvalues in the buckling case. We note that in most common vibration cases the term $\dim(\mathcal{N}(M))$ does not appear, because K is positive semidefinite. When it does appear, it is because the infinite eigenvalues have negative signs, which adds a serious complication to the problem of finding the algebraically smallest eigenvalues (the infinite eigenvalues are the algebraically smallest, but cannot be computed by the recurrence as written). However, the problem of finding the eigenvalues of smallest magnitude is only slightly more difficult in this case.

Semidefiniteness in buckling analysis is more significant, because the usual problem is to find the eigenvalues of smallest magnitude and the zero eigenvalues cannot be computed directly. The problem still can be solved if $\dim(\mathcal{N}(K))$ is known, either adventitiously or by a partial eigenanalysis of K. The problem of finding the eigenvalues of smallest magnitude in an interval bounded away from zero is still

TABLE 4
Interpretation of $\nu(K - \sigma M)$ *or* $\nu(K - \sigma K_\delta)$.

Vibration analysis:	
M positive definite	# of eigenvalues $< \sigma$
M positive semidefinite	(# of eigenvalues $< \sigma$) $+ \gamma$
	$\gamma = \begin{cases} 0 & \text{some cases} \\ \dim(\mathcal{N}(M)) & \text{other cases} \end{cases}$
Buckling analysis:	
K positive definite	# of eigenvalues in $(0, \sigma)$ or $(\sigma, 0)$
K positive semidefinite	(# of eigenvalues in $(0, \sigma)$ or $(\sigma, 0)$) $+ \gamma$
	$\gamma = \begin{cases} 0 & \sigma \text{ of one sign} \\ \dim(\mathcal{N}(K)) & \sigma \text{ of other sign} \end{cases}$

well posed.

The result of a successful eigenextraction is a trust interval containing all of the desired eigenvalues. This goal drives our selection of shifts. We create, as soon as possible, a trust interval containing some of the desired modes; thereafter, we extend the trust interval to contain more, and eventually all, of the desired modes. The process begins with an initial shift at some point σ_1. The factorization is followed by a Lanczos run with the shifted operator $(K - \sigma_1 M)^{-1} M$ (or its counterpart in buckling analysis). We will always compute a second factorization, if only to provide the inertia to close a trust interval. If only some of the desired eigenvalues were computed during the first Lanczos run, we would like to make the factorization at σ_2 serve both as a basis for an inertia computation and as the factorization for a new Lanczos run. Ideally we would choose σ_2 close enough to σ_1 that the second Lanczos run finds all the remaining eigenvalues in the interval; at the same time, we would like σ_2 to be far enough away from σ_1 so that the second Lanczos run stops, for efficiency reasons, exactly when it has computed all the missing eigenvalues. Thus, a simple description of our shift selection is that we choose each new shift to *maximally* extend an existing trust interval.

3.2. Shifting to extend a trust interval. In selecting each new shift, we try to use as much information as we have, including any computed eigenvalues, other knowledge about the existing trust interval, and additional information from the previous Lanczos runs. In general, each Lanczos run creates a set of approximations to eigenvalues, which provide a general picture of the spectrum. Figure 5 gives an illustration of the general situation, in which the last Lanczos run was at a shift σ_i that forms the right endpoint of a trust interval. The tall, thin lines denote approximations that we accept as eigenvalues. The lines of medium height and width are approximations that are not yet acceptable as eigenvalues, though they do have accuracy estimates good enough to know that at least one significant digit is correct. We call these *Ritz values*. (All of the Lanczos approximations are Ritz values, but we abuse the mathematical term to describe only those approximations that are not good enough to be accepted, and not bad enough to be meaningless.) The short, broad lines denote approximations whose accuracy estimates are larger than their values, which we ignore.

The shift selection assumes that the inverted spectrum as viewed from σ_{i+1} will be similar to the inverted spectrum as viewed from σ_i. One view of this similarity of inverted spectra is that if the Lanczos run from σ_i computed k eigenvalues to the right of σ_i efficiently, we expect that an efficient run at any σ_{i+1} should compute

k eigenvalues to its left. We use the first k Ritz values to estimate the missing eigenvalues and place the new shift σ_{i+1} between the kth and $(k+1)$st Ritz values. The choice of the bisector is intended to avoid a choice extremely close to an eigenvalue. Furthermore, we use a relaxed tolerance to detect "clusters" of eigenvalues and bisect clusters rather than Ritz values.

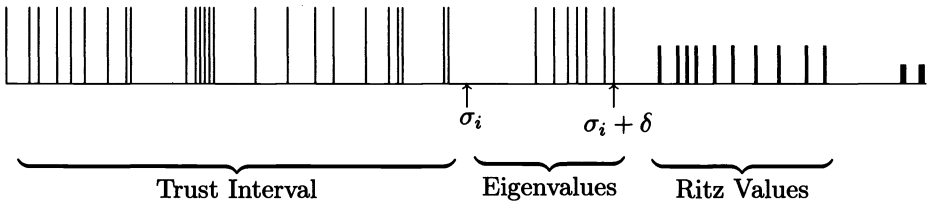


FIG. 5. *Trust intervals.*

If there are fewer than k Ritz values available to the right of σ_i, we use a second view of the inverted spectra based on the assumption that the "radius of convergence" should be about the same for each shift. We define δ to be the maximum of its previous value and the distance between the right endpoint of the trust interval and the rightmost computed eigenvalue (see Fig. 5). Initially, δ is set to the problem scale (see §3.4). Then a second choice for the next shift is $\sigma_{i+1} = \sigma_i + 2 * \delta$. We take the more aggressive choice, the maximum of the two possibilities, in the case where we still need to compute more eigenvalues than we have knowledge of Ritz values. If more Ritz values are available than there are eigenvalues left to compute, we choose the next shift based solely on the Ritz values, ignoring the shift based on δ.

Table 1 shows some results for normal, conservative, aggressive, and fixed shifting. For this table, we used a $1 * k$, $1 * \delta$ rule for conservative shifting and a $3 * k$, $3 * \delta$ rule for aggressive shifting.

We have described the general rule for choosing σ_{i+1} when σ_{i+1} is taken to the right of σ_i. Of course, we obtain two similar views of the spectra to the left of σ_i, which give another alternative for the next shift.. In general we do not know in which direction the next shift should be taken. Indeed, when finding eigenvalues nearest to an interior point we first move in one direction from σ_i and then in the other direction. At the completion of each Lanczos run in which we attempted to extend a trust interval, we compute, and save, the next shift that would extend the new trust interval further in the same direction. The first shift, unless it is at a finite endpoint of the computational interval, is treated as extending the null trust interval both to the left and to the right. The Ritz values are then discarded.

These two views of the inverted spectra, albeit simplistic, have proven to be effective. A model based on convergence rates of the eigenvalues [36] is far too pessimistic to be of any use here.

3.3. Sentinels. There are several aspects of our eigenanalysis code where the shift selection mechanism and the implementation of the Lanczos algorithm are closely tied together. For example, we do not want to recompute at later shifts eigenpairs that have been computed from earlier shifts. Any computation spent recomputing known eigenpairs is wasted. Even allowing accidental recomputation creates a difficult situation in which we must determine the correct multiplicity of a computed eigenvalue

for which several eigenvectors have been computed. We choose never to allow this situation to arise.

In theory there is a very simple fix. If the starting block for the Lanczos recurrence is chosen to be M-orthogonal to all previously computed eigenvectors, the recurrence should remain M-orthogonal to all previously computed eigenvectors. This is not sufficient in practice, as rounding errors introduce components of the excluded eigenvectors. We reorthogonalize the recurrence to specific eigenvectors only when necessary using *external selective orthogonalization* (see §4.3.3). This mechanism dramatically reduces the cost of preventing the reappearance of excluded eigenvectors.

A second mechanism for reducing this cost is in the purview of the shifting code. A common situation is depicted in Fig. 6. The new shift, σ_{i+1}, has been chosen; the nearest previous shift, σ_j, forms the end of a trust interval. (Figure 6 depicts the initial case where the trust interval including σ_j is trivial.) Between the two shifts lie a set of eigenvalues and Ritz values computed during the run at σ_j. Because the convergence rate for the eigenvalues in the Lanczos algorithm decreases as the distance from the shift increases, the usual pattern is that the accepted eigenvalues are those closest to σ_j and the Ritz values are those farther out with little or no interlacing of the two sets.

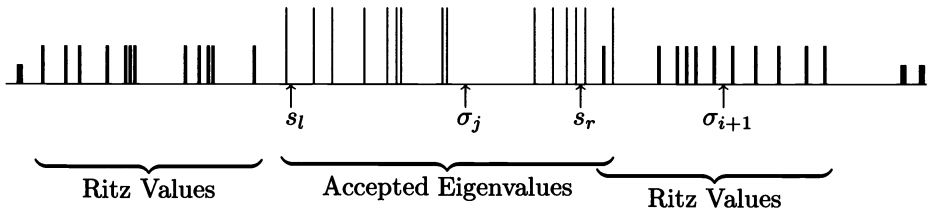


FIG. 6. *Sentinels.*

Consider in each direction the eigenvalue farthest from σ_j such that between it and σ_j no (unaccepted) Ritz values are found. There is such an eigenvalue to the right of a shift and similarly to the left, each being the last eigenvalue before a Ritz value is found. We call these two eigenvalues λ_r^* and λ_l^*. In normal circumstances we assume that there are no eigenvalues missing between σ_j and λ_r^* or λ_l^*.

We define the *right sentinel* s_r as the left endpoint of the interval of uncertainty for λ_r^*, based on the required accuracy tolerance. Thus the true value of λ_r^* lies to the right of the sentinel s_r. A *left sentinel* is defined similarly. Assume that $\sigma_{i+1} > \sigma_j$. The eigenvectors corresponding to λ_r^* and to any other eigenvalues found between s_r and σ_{i+1} are prevented from reappearing by use of external selective orthogonalization. We allow the recurrence to recompute eigenvalues which lie to the left of s_r, but these are discarded immediately. This technique allows us to trust any eigenpairs that are computed in the region in which we expect new eigenpairs to appear, without incurring the cost of extensive reorthogonalization. The reorthogonalization with respect to λ_r^*'s eigenvector removes any doubt that could exist about the exact location of this eigenvalue in the shifted and inverted spectrum for the new shift. At the same time, the eigenvector(s) most likely to reappear are suppressed.

We generalize the notion of sentinels slightly to handle clusters of eigenvalues. Should the sentinel s_r lie to the left of λ_{r-1}, we move the sentinel back to the endpoint

of the uncertainty interval for λ^*_{r-1}. We continue this process until the sentinel lies between the intervals of uncertainty for two eigenvalues, or until the shift itself is used as the sentinel.

3.4. The initial shift. The most difficult task is usually the first: getting started. The selection of the first shift must be made with no information about the spectrum other than the specification of the desired eigenvalues. We use any location information in the specification to make an initial choice for the first shift, σ_1,

$$\sigma_1 = \begin{cases} a \text{ if } |a| \le |b| \\ b \text{ if } |a| > |b| \end{cases} \Big\} \text{ if lowest modes or all modes wanted and } \min|a|,|b| < \infty,$$

$$\sigma_1 = \begin{cases} a \text{ if } |a| \ge |b| \\ b \text{ if } |a| < |b| \end{cases} \Big\} \text{ if highest modes wanted } (a \text{ and } b \text{ must both be finite}),$$

$$\sigma_1 = \xi \quad \text{if modes nearest } \xi \text{ wanted},$$

$$\sigma_1 = 0 \quad \text{otherwise.}$$

This choice of σ_1 gives a reference point in the spectrum as to which eigenvalues are important to the user. In cases where ξ is not specified by the user, we define ξ to be σ_1 as defined above. We note that 0 is a natural choice when we have no location information—in that common case we want the eigenvalues of least magnitude, i.e., closest to 0.

Unfortunately, a choice of $\sigma_1 = 0$ is fraught with difficulties. A shift at zero is not allowed in the buckling transformation and yields a singular operator in vibration analysis when K is semidefinite. If a shift at zero were taken in the latter case, it is unlikely that the singularity of the operator would be detected. It is more likely that only the zero eigenvalues would be computed and no other useful information could be extracted from the run. (The near-singularity of the operator would cause the Lanczos recurrence to break down after computing the invariant subspace of the zero eigenvalues.) This would leave us little better off than we were when we began, with no information as to where the nonzero eigenvalues are located. A better initial shift would be a shift somewhere in the vicinity of the first few nonzero eigenvalues. Such a shift would allow computing both the zero, rigid body modes and a number of the nonzero modes as well.

The difficulty is in getting some idea of the scale of the first nonzero eigenvalues. We have adopted a heuristic strategy recommended by Louis Komzsik of The MacNeal–Schwendler Corporation. This heuristic computes the geometric mean of the centers of the Gershgorin circles while excluding the centers smaller than 10^{-4}. This heuristic usually computes a reasonable *problem scale* χ. Specifically,

$$\chi = \frac{1}{l * \sum \frac{|m_{ii}|}{|k_{ii}|}},$$

where the summation is taken over all terms with $k_{ii} \neq 0$, $\frac{|m_{ii}|}{|k_{ii}|} < 10^4$; l is the number of entries included in the sum. Table 5 gives an idea of the reliability of this heuristic.

We use χ to correct the initial selection of σ_1 whenever $|\sigma_1| < \chi$. In either the vibration problem or ordinary eigenvalue problem we adjust σ_1 as

$$\sigma_1 = \begin{cases} \chi & \text{if } a \le \chi \le b, \\ -\chi & \text{otherwise, if } a \le -\chi \le b, \\ \max(|a|,|b|) & \text{otherwise.} \end{cases}$$

We adjust the initial shift in a similar fashion for the buckling problem. However, we try $\sigma_1 = -\chi$ first and then $\sigma_1 = \chi$ second, because the most common buckling analysis in structural analysis is computation of the smallest negative eigenvalue.

TABLE 5
Comparison of problem scale χ and lowest eigenvalues.

Matrix	χ	Lowest eigenvalue	Closest to eigenvalue
BCSST_08	1.8×10^{-2}	6.9×10^{0}	1
BCSST_09	1.3×10^{7}	2.9×10^{7}	1
BCSST_10	3.1×10^{-3}	7.9×10^{-2}	1
BCSST_11	3.0×10^{2}	1.1×10^{1}	12
BCSST_12	1.5×10^{3}	3.5×10^{3}	1
BCSST_13	1.2×10^{2}	1.5×10^{3}	1
BCSST_19	6.6×10^{0}	2.1×10^{0}	3
BCSST_20	5.5×10^{2}	6.6×10^{0}	7
LUND	2.1×10^{1}	2.1×10^{2}	1
PLAT1919	2.1×10^{-6}	1.1×10^{-13}	315

3.5. Choosing a direction in which to expand a trust interval. The majority of vibration analyses result in a simple, monotonic expansion of the trust interval from lowest to higher values. In these cases we know that there are no additional eigenvalues of interest to the left of the trust interval; extending the interval to the right is the only reasonable action. Cases in which we need to choose a direction arise when a shift is taken in the interior of the spectrum by accident or by design. For example, ξ is a very reasonable initial shift when we want to find eigenvalues nearest ξ. In general, finding the eigenvalues of smallest magnitude for an ordinary eigenproblem or for buckling analysis is also such a case.

We use the reference value ξ, either as set in the problem description or from the initial shift (see §3.4), to determine the direction in which to move the shift. If multiple trust intervals exist, the trust interval including or closest to ξ is *primary*; §3.7.1 describes how multiple trust intervals can exist and the logic for determining a new shift in that case. In the most typical case we have only a single trust interval, which we attempt to extend.

We distinguish two subcases, when the trust interval includes an endpoint of the computational interval and when it does not. In the first case the trust interval can only be extended in one direction without moving outside the computational interval, so the choice of direction is trivial. When the trust interval includes neither endpoint, we further distinguish between cases where ξ is or is not in the trust interval. If the trust interval does not include ξ, we shift in the direction of ξ, because that is where the eigenvalues of most importance to the user lie.

The only remaining case is of a single trust interval that contains ξ, but neither endpoint of the computational interval. In this case we compute the interval $[z_l, z_r]$ that includes the entire trust interval and all computed eigenvalues, even those outside of the trust interval. We define $r = \min(\xi - z_l, z_r - \xi)$ to be the radius of a symmetric *umbrella* about ξ where we have some degree of confidence that we have computed all the eigenvalues in the umbrella. Note that this confidence may not be confirmed by inertia values. We try to enlarge this umbrella enough to include all of the eigenvalues that the user has requested or until one end of the umbrella is an endpoint of the computational interval. We move in whichever direction increases r. Ties are broken by shifting to the left.

3.6. Analysis in a finite interval. Frequently the user of the sparse eigensolver will specify a computational interval with finite endpoints. The number of eigenvalues in the interval is usually valuable information to the user and the eigenanalysis code, even when not all of these eigenvalues are actually computed. We obtain this information at the beginning of the analysis by computing the factorization for each endpoint. If these factorizations can be used in the eigenanalysis itself, the cost of gaining this information would be nominal. (Note that both factorizations will be required in any case when all eigenvalues in the interval are requested.) We save both factorizations off-line and use them whenever it appears to be appropriate.

As discussed in the previous section, we often choose the initial shift to be one of the endpoints. If so, one of the factorizations will be used immediately. We modify the shift strategy slightly in order to take advantage of the second factorization. When the natural choice of a shift would be near an otherwise unselected finite endpoint, and when a shift at the finite endpoint would not cause a large number of extra eigenvalues to be computed, we choose the endpoint as the shift. This may result in some additional work during the Lanczos iteration, but it will save the cost of a factorization. There are cases where we can extend a trust interval to a finite endpoint without making a Lanczos run at the endpoint. These occur when the analysis at another shift results in computation of all of the eigenvalues between the shift and the endpoint.

3.7. Special cases. Robustness is one of our goals. It is naive to expect that the heuristics described above will work for all problems. Here we describe a number of special cases that can and do arise in practice and our approaches for handling them smoothly.

3.7.1. Filling gaps. The shift selection is designed to extend the trust interval obtained from previous Lanczos runs. Strange, asymmetric distributions of eigenvalues or very high multiplicities may create situations in which the shift σ_{i+1} to extend the trust interval is taken too far from σ_i to allow computation of all the eigenvalues in (σ_i, σ_{i+1}) with a single run. The inertias from σ_i and σ_{i+1} will indicate that some eigenvalues between the two shifts have not been computed.

Our goal is to maintain a trust interval, so we find the missing eigenvalues *before* we attempt to extend our knowledge beyond σ_{i+1}. We attempt to fill the *gap* between the two *active* shifts σ_i and σ_{i+1}, before proceeding. We assume that the missing eigenvalues lie between the right sentinel s_i for the shift σ_i at the left and the left sentinel s_{i+1} for the shift σ_{i+1} at the right, that is, in $[s_i, s_{i+1}]$. If the sentinel values overlap we use $[\sigma_i, \sigma_{i+1}]$ instead. In either case we have an interval $[c, d]$ in which we want to chose a shift. We choose σ_{i+2} as

$$\sigma_{i+2} = \begin{cases} \sqrt{cd} & \text{if } 0 < 2c < d, \\ -\sqrt{cd} & \text{if } c < 2d < 0, \\ \frac{c+d}{2} & \text{otherwise.} \end{cases}$$

The gap between two trust intervals is not always filled on the first attempt. The shifting strategy will continue recursively, computing missing eigenvalues, until the primary trust interval has grown large enough to contain the requested eigenvalues or when all trust intervals have been merged into one.

3.7.2. Restart at the same shift. Economizing on the number of factorizations is also a goal. In two cases a single Lanczos run will not find all the desired eigenvalues near a given shift. These occur when eigenvalues with multiplicity greater

than the blocksize exist or when a shift has been taken very close to a eigenvalue. If we suspect either case we make an additional Lanczos run at the same shift. During this run we perform external selective reorthogonalization against all newly computed eigenvectors and any other eigenvectors in the interval between this shift and any neighboring shifts. We discard any use of sentinels because the assumption behind them has probably broken down.

3.7.3. Hole in the spectrum. A particularly difficult spectrum for our selection of shifts is one with a very large disparity in the magnitudes of the desired eigenvalues. In such cases our notion of a reasonable distance may be faulty and yet we may have no Ritz value information to help us choose a new shift.

Our code treats as special a situation in which no new information is obtained at consecutive shifts. That is, we compute no meaningful Ritz values and the inertias at the two shifts σ_i and σ_{i+1} are identical. We suspect that there is a "hole" in the spectrum, that the remaining eigenvalues are farther away than our notion of a reasonable distance. We expand the notion of a reasonable distance in an attempt to cross the hole. If the computational interval $[a, b]$ has a finite endpoint that has not been used previously as a shift (see §3.6), the shift strategy will select the new shift at that endpoint. Otherwise, assuming that we are expanding a trust interval to the right, we take the new shift $\sigma_{i+2} = \sigma_{i+1} + 10\delta$ (see §3.2 for a description of δ). If this Lanczos run still provides no new information, we take $\sigma_{i+3} = \sigma_{i+2} + 100\delta$. If we still obtain no new information, we make a final attempt to cross the gap with a shift $\sigma_{i+4} = \sigma_{i+3} + 1000\delta$. If this run still provides no new information, we terminate on the assumption that the remaining eigenvalues are infinite. We return the eigenvalues already computed, together with an appropriate warning.

3.7.4. Treatment of δ in no-Ritz value cases. The setting of the "reasonable distance" value, δ, must be made carefully in cases in which the Lanczos algorithm terminates abnormally. This value is not updated if no new information is available for the next shift.

3.7.5. Overly aggressive shifts. Unusual distributions of eigenvalues or unusual convergence patterns may cause situations in which a shift is selected much farther out than required for the desired eigenvalues. We determine that the shift is too far from the current trust interval if a run at this shift will have to compute more than 30 eigenvalues before computing eigenvalues of interest to us. (The number 30 is a heuristic estimate of the number of eigenvalues we can profitably find with a single run.) In such a case we record the current shift, to keep another shift from going out too far in that direction, and select a new shift. We choose the new shift by linear interpolation between the end of the trust interval, σ_t, and the shift we reject, σ_r. The new shift is:

$$\sigma = \sigma_t + \frac{q}{[\nu(K - \sigma_r M) - \nu(K - \sigma_t M)]}(\sigma_r - \sigma_t).$$

3.8. Modifications for buckling problems. The spectral transformation used in the buckling problem for the Lanczos iteration is ill posed for shifts at or near zero. The shift strategy for the buckling problem is similar to the vibration strategy except that shifts at zero are not allowed. A shift at zero is replaced by one half the minimum of the problem scale χ, the absolute value of the shift nearest to zero, and the absolute value of the computed eigenvalue nearest to zero.

4. Implementation of the block Lanczos algorithm. The underpinning of our eigenanalysis code is the block Lanczos algorithm, as specialized for the spectral transformations (§§2.3 and 2.5). The use of the block Lanczos algorithm in the context of the spectral transformation and within applications code necessitates careful attention to a series of details: the implications of M-orthogonality of blocks; block generalizations of single vector orthogonalization schemes; effect of the spectral transformation on orthogonality loss; and interactions between the Lanczos algorithm and the shifting strategy. The success of the algorithm hinges on all of these issues.

4.1. The M-orthogonal QR factorization. Each step of the block Lanczos recurrence generates an $n \times p$ matrix R, whose column vectors are to be orthogonalized with respect to an inner product defined by a positive definite matrix, which we will call M.

Given R, we must compute its orthogonal decomposition QB such that

- $R = QB$,
- $Q^T M Q = I$,
- Q is $n \times p$,
- B is $p \times p$ and upper triangular.

When M is not the identity, the number of good choices for computing an orthogonal factorization appear to be limited. In addition, we want to avoid repeated matrix-vector multiplications with M, because we expect M, though sparse, not to be stored in main memory; each multiplication by M may require accessing secondary storage. We have developed a generalization of the modified Gram–Schmidt process that requires only matrix-block products, never matrix-vector products. We save a set of p auxiliary vectors that represent the product MQ throughout the process. This matrix is initialized to MR when the matrix that will hold Q is initialized to R; thereafter, updates made to vectors in Q are shadowed by identical updates in MQ. As a result, M is used explicitly only in the initialization.

This way of enforcing M-orthogonality certainly suggests questions of numerical stability. Following [10], we repeat the orthogonalization process up to $2p$ times, another repetition being required whenever the norm of any of the q_j vectors is less than η times its norm at the beginning of the iteration. When another repetition is required we recompute the matrix MQ by an explicit multiplication by M. The choice of $\eta = \sqrt{2}/2$ from [10] guarantees that the final set of vectors is orthonormal.

In our algorithm for computing the M-orthogonal factorization (Fig. 7), the vectors w_j are the auxiliary vectors that represent the vectors Mq_j. The matrix $\hat{B}$ is the triangular matrix computed in one iteration of the algorithm; the M-orthogonal triangular factor B is the product of all of the individual triangular matrices $\hat{B}$.

It should be noted that this algorithm may encounter a rank deficient set of vectors q_j and identically zero vectors are possible. Further details can be found in our discussion on when to terminate a Lanczos run (§4.4).

We have assumed in the discussion above that M is positive definite. In the case of M positive semidefinite, the recurrence, when properly started, generates a sequence of blocks, all of whose columns lie in the range of $(K - \sigma M)^{-1} M$. This is the subspace from which the eigenvectors corresponding to finite eigenvalues must be drawn [13]. Clearly, the orthogonalization algorithm preserves this subspace. Further, this subspace has only the trivial intersection with the nullspace of M [13], [29]. Thus, the appearance of a nontrivial column with zero M-norm represents a breakdown equivalent to rank deficiency, since such a vector cannot lie in the range of $(K - \sigma M)^{-1} M$.

Initialization:

$Q = R$
$B = I$

Factorization:

Repeat
W = MQ

For $i = 1, 2, \ldots p$ do
$\hat{b}_{ii} = \sqrt{q_i^T w_i}$
$q_i = q_i / \hat{b}_{ii}$
$w_i = w_i / \hat{b}_{ii}$
For $j = i + 1, \ldots p$ do
$\hat{b}_{ij} = q_i^T w_j$
$q_j = q_j - \hat{b}_{ij} q_i$
$w_j = w_j - \hat{b}_{ij} w_i$
End
End

$B = \hat{B}B$

Until convergence or iteration limit exceeded

FIG. 7. *M-orthogonal modified Gram–Schmidt orthogonalization.*

4.2. Analysis of the block tridiagonal matrix T_j. The original eigenvalue problem is reduced by the block Lanczos algorithm to an eigenvalue problem of the form $T_j s = s\theta$, where T_j is a block tridiagonal matrix. In §2.2 we noted the standard result by which bounds on the accuracy of the computed eigenvalues can be computed without explicit computation of the eigenvectors. These bounds are used to determine whether to terminate the Lanczos recurrence and to evaluate which eigenpairs are accurate enough to be considered to have converged. The results in §2.2 generalize to provide a bound on the accuracy of the approximate eigenvalues of the spectrally transformed problem. However, our real interest is in the accuracy of our approximations to the original, untransformed problem. We need to determine which eigenpairs of the original problem have converged, and we need accuracy estimates for all of the Ritz values for use in the shift selection process. To get these estimates we need to unravel the effects of the spectral transformation. Throughout we must account for possibly multiple eigenvalues.

Recall that the following relation (14) holds for vibration analysis:

$$(K - \sigma M)^{-1} M y - y\theta = Q_{j+1} B_{j+1} E_j^T s.$$

Therefore, because Q_{j+1} is M-orthogonal,

$$\begin{aligned} \|M(K - \sigma M)^{-1} M - My\theta\|_{M^{-1}} &= \|MQ_{j+1} B_{j+1} E_j^T s\|_{M^{-1}} \\ &= \|B_{j+1} E_j^T s\|_2 \equiv \beta_j. \end{aligned}$$

For each eigenvector s the corresponding β_j is the Euclidean norm of the product of the upper triangular matrix B_{j+1} with the last p components of s. We apply a theorem on the error in eigenvalue approximations for the generalized eigenproblem from [32, p. 318] to obtain:

$$\left|\frac{1}{\lambda-\sigma}-\theta\right| \leq \frac{\|M(K-\sigma M)^{-1}My - My\theta\|_{M^{-1}}}{\|My\|_{M^{-1}}} = \beta_j. \tag{18}$$

Thus, as in the ordinary eigenproblem, β_j is a bound on how well the eigenvalue of T_j approximates an eigenvalue of the operator to which the Lanczos algorithm is applied. We extend this to find a bound on the error $|\lambda-\nu|$.

Ericsson and Ruhe [14] show that

$$|\lambda-\nu| \leq \frac{\beta_j}{\theta^2}. \tag{19}$$

This shows how the accuracy requirements are modified by the spectral transformation. When λ is close to the shift σ we need only a moderately small β_j to guarantee a good approximate eigenvalue ν because θ is large. Conversely, eigenvalues far from the shift are transformed to small values of θ, requiring smaller values of β_j than would otherwise be expected.

The bound (19) can be improved for well-separated eigenvalues. Define the gap γ as:

$$\gamma \equiv \min_{\lambda_i \neq \lambda} \left|\frac{1}{\lambda_i-\sigma} - \frac{1}{\lambda-\sigma}\right|,$$

The gap bound theorem from [32, p. 222] then results in

$$|\lambda-\nu| \leq \frac{\beta_j^2}{\theta^2\gamma}. \tag{20}$$

Both bounds (19) and (20) are valid. In general, the first is smaller than the second for clustered eigenvalues and larger for well-separated eigenvalues. In our implementation we use whichever bound is smaller:

$$|\lambda-\nu| \leq \min\left\{\frac{\beta_j}{\theta^2}, \frac{\beta_j^2}{\theta^2\gamma}\right\}. \tag{21}$$

The definition of γ should be modified to account for clusters of eigenvalues; the gap between sets of multiple eigenvalues is used. In practice we have only an approximation to γ, which we derive from the shifted and inverted eigenvalues of T_j.

Similar error bounds can be derived for buckling analysis. Let (ν, y) be a computed eigenpair of (K, K_δ). Then $\theta = \frac{\nu}{\nu-\sigma}$ and $|\frac{\lambda}{\lambda-\sigma} - \theta| \leq \beta_j$. From the fact that $\nu = \frac{\sigma\theta}{\theta-1}$, it follows that

$$\begin{aligned}
\lambda - \nu &= \lambda - \frac{\sigma\theta}{\theta-1} \\
&= \left(\frac{1}{\theta-1}\right)(\lambda(\theta-1) - \sigma\theta) \\
&= \left(\frac{1}{\theta-1}\right)(\lambda-\sigma)\left(\theta - \frac{\lambda}{\lambda-\sigma}\right) \\
&= \left(\frac{\sigma}{\theta-1}\right)\left(\frac{\lambda-\sigma}{\sigma}\right)\left(\theta - \frac{\lambda}{\lambda-\sigma}\right).
\end{aligned}$$

The Lanczos algorithm approximates θ by a projection onto a subspace. When the inversion of the operator is taken into account, the computed eigenvalues of the transformed problem are always closer to one than the true eigenvalues of the transformed problem. Therefore,

$$\left|\frac{\lambda-\sigma}{\sigma}\right| = \frac{1}{|\frac{\lambda}{\lambda-\sigma}-1|} \leq \frac{1}{|\theta-1|}.$$

The resulting simple error bound for buckling analyses is

$$|\lambda-\nu| \leq \frac{|\sigma|}{(\theta-1)^2}\beta_j.$$

The analogous refined gap error bound is

$$|\lambda-\nu| \leq \frac{|\sigma|\beta_j^2}{(\theta-1)^2}\gamma_b,$$

where γ_b is defined by

$$\gamma_b \equiv \min_{\lambda_i \neq \lambda}\left|\frac{\lambda}{\lambda_i-\sigma}-\frac{\lambda}{\lambda-\sigma}\right|.$$

As in the vibration case, the lesser of the two bounds

$$|\lambda-\nu| \leq \min\left\{\frac{|\sigma|}{(\theta-1)^2}\beta_j, \frac{|\sigma|\beta_j^2}{(\theta-1)^2}\gamma_b\right\} \tag{22}$$

is chosen, with the definition of γ_b modified in the presence of multiple eigenvalues.

The spectral transformation preserves the eigenvectors, so there is no need to account for the transformation *vis à vis* the approximate eigenvectors. However, Ericsson and Ruhe [14] introduced a correction term that results in improved eigenvector approximations for the untransformed problem. This was later discovered to have the additional benefit [29] of ensuring that the computed eigenvectors lie in the proper subspace in cases where the metric matrix is semidefinite.

Let $\nu = \sigma + \frac{1}{\theta}$ be the computed eigenvalue. The correction step is formally one step of inverse iteration with the computed eigenvector y; $\tilde{z}$ is computed to satisfy $(K-\sigma M)\tilde{z} = My$. By (14)

$$\tilde{z} = (K-\sigma M)^{-1}My = y\theta + Q_{j+1}B_{j+1}E_j^T s.$$

The vector

$$z = \frac{1}{\theta}\tilde{z} = y + \frac{1}{\theta}Q_{j+1}B_{j+1}E_j^T s$$

can be obtained cheaply by adding a linear combination of the vectors in the next block of Lanczos vectors to y. This gives a better approximation to the eigenvector of the vibration problem and ensures that the approximate eigenvectors are uncontaminated by the effects of a semidefinite M. The corresponding correction for a semidefinite K in buckling analysis is given by

$$z = y + \frac{1}{\theta-1}Q_{j+1}B_{j+1}E_j^T s.$$

During the course of the Lanczos algorithm we need an estimate of the residual bounds. The bounds in (21) or (22) require most of the eigenvalues and the corresponding entries in the bottom block row of the matrix of eigenvectors. Parlett and Nour-Omid [33] have a very efficient algorithm for the single vector Lanczos algorithm. Block generalizations have yet to be found, so we use a more straightforward approach. The eigenvalue problem for T_j is solved by reducing the band matrix T_j to tridiagonal form and then by applying the tridiagonal QL algorithm. We use subroutines from EISPACK [16], [41], with slight modifications, to obtain only the bottom p entries of the eigenvectors of T_j. These modifications reduce considerably both computation and storage requirements for each Lanczos step. Only $p^2 j$ words are needed as opposed to $(pj)^2$ for the full eigenvector matrix. We use the corresponding unmodified routines to obtain the full eigenvectors at the conclusion of a Lanczos run, at which time temporary space used during the recurrence is available to store the entirety of the eigenvector matrix for T.

4.3. Global loss of orthogonality and reorthogonalization. Up to this point our discussion of the block Lanczos algorithm has assumed exact arithmetic, but the various error bounds hold in finite precision as well. It is well known that there is a global loss of orthogonality among the computed Lanczos vectors in inexact arithmetic. A reasonable correction is to perform *limited* reorthogonalization to keep $\mathcal{Q}_j$ sufficiently close to orthogonal. Our approach is twofold—we identify mechanisms whereby orthogonality is lost and then apply a model of the loss of orthogonality to determine when to correct the situation. In the context of the block shifted Lanczos recurrence, orthogonality is lost in three different ways. First, there is a loss of orthogonality between adjacent blocks in $\mathcal{Q}_j$, the blocks the recurrence should make orthogonal. This is corrected by use of *local reorthogonalization.* Second, the recurrence suffers a global loss of orthogonality with respect to the blocks of $\mathcal{Q}_j$ not explicitly involved in the reorthogonalization. We correct for this with a block version of *partial reorthogonalization.* Lastly, it is important that a Lanczos run at some shift not recompute eigenvectors computed as a result of a previous Lanczos run. We present a new reorthogonalization scheme, *external selective reorthogonalization,* to ensure that this does not occur. Throughout the process our goal is to apply a minimal amount of extra work, particularly as it requires accessing the entirety of $\mathcal{Q}_j$, to maintain at least $\mathcal{O}(\sqrt{\epsilon})$-orthogonality in $\mathcal{Q}_j$.

The fundamental approach is to model the Lanczos recurrence in finite precision. The following recurrence is our model of what really happens:

$$Q_{j+1}B_{j+1} = (K - \sigma M)^{-1} M Q_j - Q_j A_j - Q_{j-1} B_j^T + F_j, \tag{23}$$

where F_j represents the roundoff error introduced at step j. Then,

$$\begin{aligned} Q_k^T M Q_{j+1} B_{j+1} = {} & Q_k^T M (K - \sigma M)^{-1} M Q_j - Q_k^T M Q_j A_j \\ & - Q_k^T M Q_{j-1} B_j^T + Q_k^T M F_j. \end{aligned}$$

For convenience we define $W_{j,k} \equiv Q_k^T M Q_j$, with which the previous equation becomes

$$W_{j+1,k} B_{j+1} = Q_k^T M (K - \sigma M)^{-1} M Q_j - W_{j,k} A_j - W_{j-1,k} B_j^T + Q_k^T M F_j. \tag{24}$$

This equation is nearly sufficient for our computational purposes. We can easily find norms for the blocks A_j and B_j during the recurrence, and we will compute bounds for all the other terms except for the first term on the right side of (24).

We eliminate $Q_k^T M(K - \sigma M)^{-1} M Q_j$ from (24) by obtaining an expression for its transpose by premultiplying the occurrence of (23) with $j = k$ by $Q_j^T M$:

$$Q_j^T M(K - \sigma M)^{-1} M Q_k = W_{j,k+1} B_{k+1} + W_{k,j} A_j + W_{k-1,j} B_k^T + Q_j^T M F_k.$$

The obvious substitution then results in

$$\begin{aligned} W_{j+1,k} B_{j+1} = {} & B_{k+1}^T W_{j,k+1} + A_k W_{j,k} + B_k W_{j,k-1} \\ & - W_{j,k} A_j - W_{j-1,k} B_j^T + G_{j,k}. \end{aligned} \tag{25}$$

Here $G_{j,k} \equiv Q_k^T M F_j - F_k^T M Q_j$ represents the local roundoff error. Formula (25) explains the global loss of orthogonality. We will use this model to estimate and bound the loss of orthogonality among the Lanczos vectors and thereby determine how to correct the loss of orthogonality.

4.3.1. Monitoring the loss of orthogonality. The development of our modeling procedure has two parts, both based on the bounds available by taking norms of (25):

$$\begin{aligned} \|W_{j+1,k}\|_2 \le {} & \|B_{j+1}^{-1}\|_2 (\|B_{k+1}\|_2 \|W_{j,k+1}\|_2 \\ & + \|B_k\|_2 \|W_{j,k-1}\|_2 + \|B_j\|_2 \|W_{j-1,k}\|_2 \\ & + (\|A_j\|_2 + \|A_k\|_2) \|W_{j,k}\|_2 + \|G_{j,k}\|_2). \end{aligned}$$

We use this equation to compute a bound $\omega_{j,k}$ on $\|W_{j,k}\|_2$ at each step.

The first part of our development addresses the bounds $\omega_{j+1,k}$ for $k \le j - 1$, that is, for blocks that are not explicitly involved in the orthogonalization of the Lanczos vectors within the recurrence itself. For these blocks the loss of orthogonalization depends on the loss already incurred at previous steps. Bounds on that loss of orthogonality will be available to us from previous steps of the simulation given in Fig. 8.

Initialize:

$\epsilon_s \equiv \epsilon p \sqrt{n}$, where $\epsilon \equiv$ roundoff unit, p is the blocksize
and $n =$ number of degrees of freedom
$\omega_{2,1} = \epsilon_s$

Loop:

For $j = 2, 3, 4, \ldots$ do
$\omega_{j+1,j} = \epsilon_s$
$\omega_{j+1,j-1} = \tilde{\beta}_{j+1}(2\beta_j \epsilon_s + (\alpha_j + \alpha_{j-1})\epsilon_s + \beta_{j-1}\omega_{j,j-2})$
For $k = 1, \ldots j - 2$ do
$\omega_{j+1,k} = \tilde{\beta}_{j+1}(\beta_{k+1}\omega_{j,k+1} + \beta_k \omega_{j,k-1} + \beta_j \omega_{j-1,k} + (\alpha_j + \alpha_k)\omega_{j,k})$
End
End

FIG. 8. *Simulation of loss of orthogonality* (ω-*recurrence*).

The following quantities from the Lanczos recurrence are required for the simulation:

$$\alpha_k \equiv \|A_k\|_2,$$

$\beta_k \equiv \|B_k\|_2$,
$\tilde{\beta}_k \equiv 1/\sigma_p(B_k)$, where $\sigma_p(B_k)$ is the smallest singular value of B_k.

In addition, we follow [32], [37], [39] in making a standard assumption on a bound for the error term: $\|G_{j,k}\|_2 \leq \epsilon_s = \epsilon p\sqrt{n}$. We have left unstated the origin of the two initializing terms, $\omega_{j+1,j}$ and $\omega_{j+1,j-1}$. In examining them we will uncover a particular artifact of the *block* Lanczos algorithm. By (25),

$$\begin{aligned} W_{j+1,j-1}B_{j+1} &= B_j^T W_{j,j} + A_{j-1}W_{j,j-1} + B_{j-1}W_{j,j-2} \\ &\quad -W_{j,j-1}A_j - W_{j-1,j-1}B_j^T + G_{j,j-1} \\ &= (B_j^T W_{j,j} - W_{j-1,j-1}B_j^T) \\ &\quad +(A_{j-1}W_{j,j-1} - W_{j,j-1}A_j) \\ &\quad +B_{j-1}W_{j,j-2} + G_{j,j-1}. \end{aligned}$$

By reason of the care with which we compute the QR factorization, we assume that $Q_j^T M Q_j = I + E$, where I is the identity matrix and $\|E\|_2 \leq \epsilon_s$. For reasons discussed below, we can assume that $\|W_{j,j-1}\|_2 \leq \epsilon_s$. From this it follows that

$$\|W_{j+1,j-1}\|_2 \leq \tilde{\beta}_{j+1}(2\beta_j\epsilon_s + (\alpha_j + \alpha_{j-1})\epsilon_s + \beta_{j-1}\omega_{j,j-2}). \tag{26}$$

Notice from (26) that $\omega_{j+1,j-1} > \tilde{\beta}_{j+1}\beta_j\epsilon_s$. At the next step this term will appear as $\tilde{\beta}_{j+2}\beta_j\omega_{j+1,j-1}$; in the following step it will be one of the contributions to $\tilde{\beta}_{j+3}\beta_{j+1}\omega_{j+2,j}$. Both $\tilde{\beta}_{j+1}$ and β_{j+1} appear in this last product. The growth of the bound occurs as fast as $\kappa(B_j) = \tilde{\beta}_{j+1}\beta_{j+1}$, the condition number of B_j. The analysis of the ordinary Lanczos algorithm has unity corresponding to the term $\kappa(B_j)$, because the condition number of a nonzero 1×1 matrix is always one. The loss of orthogonality occurs more rapidly in the block Lanczos algorithm, particularly when $\kappa(B_j)$ is significantly larger than one, but also in general.

A different, but related, analysis can be used to show that the term $\kappa(B_j)$ appears in the bound for $\omega_{j+1,j}$. This was first observed in [23], where this growth was also actually observed in the Lanczos recurrence. An inexpensive correction is needed to make the recurrence useful: at each step a *local reorthogonalization* between Q_{j+1} and Q_j is performed. Because the Lanczos recurrence is itself just a special form of Gram–Schmidt orthogonalization, local reorthogonalization can be seen as a simple generalization of the reorthogonalization required in computing the M-orthogonal factorization of a single block. Local reorthogonalization ensures that ϵ_s-orthogonality holds between successive blocks of Lanczos vectors. Note that a local orthogonalization step is also performed on completion of a partial reorthogonalization. If storage is not an issue, a local reorthogonalization between Q_{j+1} and Q_{j-1} should also be performed, in which the obvious modification should be made to the algorithm for computing the ω-recurrence.

4.3.2. Partial reorthogonalization. The global loss of orthogonality modeled by the ω-recurrence can be corrected by two different schemes. These are the selective orthogonalization scheme of Parlett and Scott [35] and the partial reorthogonalization scheme of Simon [40]. Selective orthogonalization takes advantage of the fact that orthogonality is lost exactly in the direction of eigenvectors that have become well represented in $\mathcal{Q}_j$. Selective orthogonalization is implemented in two steps. In the first, the Lanczos recurrence is "interrupted" when an eigenvector converges. The

eigenvector is computed, which requires access to all previous blocks in $\mathcal{Q}_j$. The second step occurs whenever the model indicates orthogonality is lost again in the direction of the eigenvector. The second step requires that the latest two Lanczos blocks be reorthogonalized against the computed eigenvector, but does not require access to preceding blocks of $\mathcal{Q}_j$.

Partial reorthogonalization interrupts the recurrence to reorthogonalize Q_j and Q_{j+1} against all preceding blocks whenever the simulation indicates too great a loss of orthogonality. Each reorthogonalization step requires access to all of $\mathcal{Q}_j$. For this reason partial reorthogonalization has previously been recommended for situations in which the eigenvectors were not of any interest (as in solving sparse linear equations [40]). The extra cost in an application of partial reorthogonalization does have an extra payoff; orthogonality is restored against all converged and nearly converged eigenvectors simultaneously.

TABLE 6
Comparison of partial and selective reorthogonalization.

Matrix	Eigen-values	Block steps	Partial reorthog. steps	Selective orthog. steps
BCSST_26[a]	211	181	51	98
PLAT1919[b]	636	579	143	291

[a] blocksize 3, lowest 200 modes.
[b] blocksize 3, all modes in [.000025, .24].

Shifting and the block recurrence each accelerate the convergence of eigenpairs; together they cause eigenpairs to converge very rapidly. Frequently one or more eigenpairs converge at *each* block step, once the recurrence is established. In this circumstance selective orthogonalization has possibly greater requirements for accessing $\mathcal{Q}_j$ than does partial reorthogonalization. Selective orthogonalization will require an eigenvector computation at almost each step; partial reorthogonalization will occur only every three to four steps in typical problems. It would be possible to combine the two schemes—to carry out partial reorthogonalization during the computation of an eigenvector for selective orthogonalization, but it is not clear that the combination would be more effective than partial reorthogonalization alone. (See [34] for a discussion of these issues for the ordinary Lanczos recurrence.) Table 6 summarizes the reorthogonalization requirements of two extensive eigencomputations. The number of selective orthogonalization steps given in this table is the number of block steps at which one or more eigenvalues converge; the number of partial reorthogonalization steps is the number of block steps at which partial reorthogonalization was performed.

Our implementation of the Lanczos recurrence uses the block generalization of partial reorthogonalization, based on the block ω-recurrence presented above. The single vector version of this simulation has been shown previously [40] to provide a good order of magnitude estimate of growth of the loss of orthogonality, as well as a bound. We use the block version to estimate the loss of orthogonality to determine when reorthogonalization is necessary. Previous work [32], [35], [39] indicates that reorthogonalization is needed whenever

$$\max_k \omega_{j+1,k} \geq \sqrt{\epsilon}.$$

The reorthogonalization should be carried out with both of the last two block of vectors Q_j and Q_{j+1}, in order that the next block generated by the recurrence, Q_{j+2},

be strictly orthogonal to all of its predecessors. This leads to the following *partial reorthogonalization* [40] algorithm (Fig. 9) for maintaining orthogonality:

At each Lanczos step, after computing Q_{j+1} and B_{j+1}, do:

Update the ω-recurrence as above

$\omega_{\max} \equiv \max_k \omega_{j+1,k}$
If $\omega_{\max} \geq \sqrt{\epsilon}$ then
  For $k = 1, \ldots, j-1$ do
    Orthogonalize Q_j against Q_k
    Orthogonalize Q_{j+1} against Q_k
  End
  Orthogonalize Q_{j+1} against Q_j
  Reinitialize ω-recurrence:
    $\omega_{j+1,k} = \omega_{j,k} = \epsilon_s, k = 1, \ldots, j$
End if

FIG. 9. *Partial reorthogonalization.*

Note that the orthogonalization of Q_j and Q_{j+1} involves M-inner products. This requires the storage of both the Lanczos vectors and their product with M in secondary storage, or, alternatively, reapplying M to the Lanczos vectors. The appropriate form depends on cost.

4.3.3. External selective orthogonalization. A different type of loss of orthogonality occurs in the context of the shifted and inverted Lanczos algorithm. It is possible that, after computing some eigenvalues with shift σ_1, the same eigenvalues and vectors are computed again with shift σ_2. External selective orthogonalization is an efficient way of keeping the current sequence of Lanczos vectors orthogonal to previously computed eigenvectors, and thereby avoiding the recomputation of eigenvalues that are already known. External selective orthogonalization is motivated by the classical selective orthogonalization algorithm [35], but the development here is entirely new.

In theory it would be sufficient to orthogonalize the starting block against known eigenvectors, because all subsequent Lanczos vectors would be orthogonal as well. Of course, this does not hold in practice. A global loss of orthogonality occurs, similar to the one among the Lanczos vectors themselves; in addition, the computed eigenvector is not exact. The contribution of both sources of error to the recomputation of eigenvalues and vectors is analyzed below.

Let (ν, y) be an approximate eigenpair of (K, M). For clarity, denote the current shift as σ_{new}. The relationship between the eigenvector y and the Lanczos vectors obtained with the shift σ_{new} is found by premultiplying the finite precision recurrence (23) by $y^T M$ to obtain

$$y^T M Q_{j+1} B_{j+1} = y^T M (K - \sigma_{\text{new}} M)^{-1} M Q_j - y^T M Q_j A_j - y^T M Q_{j-1} B_j^T + y^T M F_j. \tag{27}$$

We assume that B_{j+1} is nonsingular. Then we can obtain a bound on the loss of orthogonality between y and Q_j by taking norms of both sides of (27):

$$\|y^T M Q_{j+1}\|_2 \leq \|B_{j+1}^{-1}\|_2 (\|y^T M (K - \sigma_{\text{new}} M)^{-1} M Q_j - y^T M Q_j A_j\|_2$$

$$+ \|y^T M Q_{j-1}\|_2 \|B_j^T\|_2 + \|y^T M F_j\|_2).$$

As with partial reorthogonalization, we can define a recurrence relation for a quantity τ_j to bound the loss of orthogonality between y and the Lanczos vectors. Assuming that $\tau_i \geq \|y^T M Q_i\|_2$ for $i = 1, \ldots, j$, we obtain

$$\|B_{j+1}^{-1}\|_2(\|y^T M(K - \sigma_{\text{new}} M)^{-1} M Q_j - y^T M Q_j A_j\|_2 + \tau_{j-1}\|B_j^T\|_2 + \|y^T M F_j\|_2)$$

as a bound for the right-hand side of the $j+1$st step. Of the three terms on the right-hand side of this equation, the second is easily computed and we have a standard assumption for a bound on the third: $\|F_j\|_2 \leq \epsilon p\sqrt{n}$. We need then only to bound the first term $\|y^T M(K - \sigma_{\text{new}} M)^{-1} M Q_j - y^T M Q_j A_j\|_2$. The spectral transformations preserve eigenvectors, so y is also an approximate eigenvector of the spectrally transformed problem. Define the transformed residual vector z_{new} by

$$(K - \sigma_{\text{new}} M)^{-1} M y - \frac{1}{\nu - \sigma_{\text{new}}} y = z_{\text{new}}.$$

Then

$$y^T M(K - \sigma_{\text{new}} M)^{-1} M Q_j = \frac{1}{\nu - \sigma_{\text{new}}} y^T M Q_j + z_{\text{new}}^T M Q_j,$$

from which it follows that

$$\|y^T M(K - \sigma_{\text{new}} M)^{-1} M Q_j - y^T M Q_j A_j\|_2 \leq \left\| \left(\frac{1}{\nu - \sigma_{\text{new}}} I - A_j \right) \right\|_2 \tau_j + \|z_{\text{new}}^T M Q_j\|_2.$$

But

$$\begin{aligned} \|z_{\text{new}}^T M Q_j\|_2 &= \|(z_{\text{new}}^T M^{1/2})(M^{1/2} Q_j)\|_2 \\ &\leq \|z_{\text{new}}^T M^{1/2}\|_2 \|M^{1/2} Q_j\|_2 = \|z_{\text{new}}^T\|_M \|Q_j\|_M = \|z_{\text{new}}^T\|_M. \end{aligned}$$

Thus, the following simple recurrence for τ gives a bound for the loss of orthogonality observed in (27):

(28)

$$\tau_{j+1} = \|B_{j+1}^{-1}\|_2 \left(\tau_j \left\| \left(\frac{1}{\nu - \sigma_{\text{new}}} I - A_j \right) \right\|_2 + \tau_{j-1}\|B_j^T\|_2 + \|z_{\text{new}}\|_M + \epsilon p\sqrt{n} \right).$$

The same analysis applies to the buckling spectral transformation, where the eigenvector orthogonality error (27) becomes:

$$\begin{aligned} y^T K Q_{j+1} B_{j+1} = {} & y^T K(K - \sigma_{\text{new}} K_\delta)^{-1} K Q_j - y^T K Q_j A_j \\ & - y^T K Q_{j-1} B_j^T + y^T K F_j. \end{aligned}$$

The transformed residual vector z_{new} is

$$(K - \sigma_{\text{new}} K_\delta)^{-1} K y - \frac{\nu}{\nu - \sigma_{\text{new}}} y = z_{\text{new}}.$$

By the same analysis as above, the recurrence for τ in the buckling context is

(29)

$$\tau_{j+1} = \|B_{j+1}^{-1}\|_2 \left(\tau_j \left\| \left(\frac{\nu}{\nu - \sigma_{\text{new}}} I - A_j \right) \right\|_2 + \tau_{j-1}\|B_j^T\|_2 + \|z_{\text{new}}\|_K + \epsilon p\sqrt{n} \right).$$

The recurrences in (28) and (29) provide a mechanism for estimating the loss of orthogonality to externally computed eigenvectors, regardless of the source. Each requires computing the transformed residual vector, z_{new}, and its norm, but the recurrence applies to situations where eigenvectors are known adventitiously. For example, in the vibration analysis of structures where K is singular, the so-called rigid body modes, the zero eigenvalues and vectors, often can be computed at much less cost than a factorization. Typically, the cost of computing the residual norms for all of the vectors involved in external selective orthogonalization is less than the cost of one additional step of the Lanczos recurrence.

In the context of a Lanczos code within a larger shifting strategy, it would be attractive to use the information from the Lanczos recurrence to bound the errors in the computed eigenvectors and thereby avoid having to compute $\|z_{\text{new}}\|_M$. In [20] we provide an analysis for the case where the approximate eigenpair (ν, y) was computed by the Lanczos code at a previous shift σ_{old}. However, we use the more general form exclusively in our code.

As with partial reorthogonalization, we define a recurrence relation for a quantity τ_j that estimates the loss of orthogonality of the Lanczos vectors with respect to y. In the recurrence, τ_j is defined be:

$$\tau_{j+1} = \tilde{\beta}_{j+1}(\alpha_{\nu\sigma j}\tau_j + \beta_j\tau_{j-1} + \|z_{\text{new}}\|_M), \tag{30}$$

which we initialize with $\tau_0 \equiv 0$ and $\tau_1 = \epsilon p\sqrt{n}$. The terms β_j and $\tilde{\beta}_{j+1}$ are defined as in the ω-recurrence. The term $\alpha_{\nu\sigma j} \equiv \|(\nu - \sigma)^{-1}I - A_j\|_2$.

An external selective orthogonalization is performed whenever $\tau_{j+1} \geq \sqrt{\epsilon}$. A relatively large residual for the computed eigenvector will cause frequent reorthogonalization, but, as noted below, usually only a very small number of vectors are actually involved. External selective orthogonalization is implemented as in Fig. 10.

Before the Lanczos iteration:
    Determine the set of SO-vectors (eigenvectors for selective orthogonalization)
    Orthogonalize Q_1 against the SO-vectors.
    Orthogonalize Q_2 against the SO-vectors.

At each Lanczos step $j = 3, 4, \ldots$ do:
    Update the τ-recurrence according to (30) for each SO-vector;
    If (τ_j has been greater than $\sqrt{\epsilon}$ or $\tau_{j+1} \geq \sqrt{\epsilon}$) then
        Orthogonalize Q_{j+1} against y
        Set $\tau_{j+1} = \epsilon_s$
    End if

FIG. 10. *External selective orthogonalization.*

It is unnecessary to perform external selective orthogonalization against *all* previously computed eigenvectors. From (28) and (29) it is evident that one of the main driving forces in the loss of orthogonality is $(\nu - \sigma)^{-1}$. It would appear that loss of orthogonality should mostly occur in the direction of eigenvectors corresponding to eigenvalues close to the new shift. Furthermore, as discussed in §3.3, only a few eigenvectors, again usually those close to the new shift, need be considered in order to avoid confusing new eigenvectors with the old. In our implementation, we use sentinels to reduce the cost of maintaining orthogonality. The set of eigenvectors used

for external selective orthogonalization is usually the eigenvectors corresponding to any known eigenvalues closer to the shift than the sentinels. Eigenvalues beyond the sentinels are discarded in the analysis of the block tridiagonal system.

The effect of using sentinels on the work required for external selective orthogonalization is more dramatic than is suggested by the analysis above. Although proximity to the shift is the driving force in the growth of τ, neither recurrence (28) nor (29) begins at ϵ. The term $\|z_{\text{new}}\|_M$ is usually near $\sqrt{\epsilon}$. The eigenvalues themselves are only good to the convergence tolerance (usually $\epsilon^{2/3}$ in our code). Furthermore, the spectral transformations preserve eigenvectors, but do not preserve the property of being the best minimizers for approximate eigenvalues (see [14] for a discussion of the need to modify the approximate eigenvectors). As a result, external selective orthogonalization happens more often than we might expect, often at every step for the eigenpairs nearest the sentinels, which frequently are simultaneously least accurate and nearest the new shift.

Experimental results are shown for two examples in Table 7. The results shown as "with sentinels" refers to the selection described in §3.3; the results shown as "without sentinels" uses as SO-vectors all eigenvectors in the intervals between the current shift and any neighboring trust intervals. The figure given as "cpu cost" includes both cpu time and i/o processor time. The difference between the costs for the two variations gives only a rough idea of the added cost for complete selective orthogonalization because the difference in cost affects the termination decision for each run and thereby changes the choice of shifts.

TABLE 7
External selective orthogonalization.

Matrix	With sentinels			Without sentinels		
	Average number of S.O. Vectors	Total number of S.O. Steps	cpu cost	Average number of S.O. Vectors	Total number of S.O. Steps	cpu cost
BCSST_26[a]	2.1	313	174.2	15.7	2265	222.7
PLAT1919[b]	6.4	2776	668.2	28.1	8692	801.5

[a] blocksize 3, lowest 200 modes.
[b] blocksize 3, all modes in [.000025, .24].

The orthogonalizations involve again both y and My. In order to avoid the repeated computation of My, all selective orthogonalization vectors are premultiplied by M and the result is stored on the same random access file as the eigenvectors y. This computation is performed before the actual Lanczos run begins.

4.3.4. Summary of reorthogonalization schemes. We now present in summary form the reorthogonalized block Lanczos algorithm we use in our production code. Our scheme consists of applying, in turn, external selective, partial, and local reorthogonalization to the result of a single block Lanczos step. The first two schemes are applied only when the respective model signals a need; each should be applied before orthogonality is lost badly enough that repeated orthogonalizations are needed. The local reorthogonalization is applied at each step. It may be applied repeatedly, but this normally occurs only when the recurrence has broken down, which will cause termination. The integration of these is indicated in Fig. 11.

4.4. Cost analysis and termination of a Lanczos run. The block Lanczos algorithm exists as part of a larger code, in which each Lanczos run solves only a

Initialization:

Set $Q_0 = 0$
Set $B_1 = 0$
Choose R_1 and orthonormalize the columns of R_1 to obtain Q_1 with $Q_1^T(MQ_1) = I_p$.

Lanczos Loop:

For $j = 1, 2, 3 \ldots$ do

Set $U_j = (K - \sigma M)^{-1}(MQ_j) - Q_{j-1}B_j^T$
Set $A_j = U_j^T(MQ_j)$
Set $R_{j+1} = U_j - Q_jA_j$
Compute Q_{j+1} and (MQ_{j+1}) such that
a) $Q_{j+1}B_{j+1} = R_{j+1}$
b) $Q_{j+1}^T(MQ_{j+1}) = I_p$

Evaluate γ-recurrence for each SO vector and perform selective orthogonalization if necessary
Evaluate ω-recurrence and perform partial reorthogonalization if necessary

Repeat up to $2p$ times:
Reorthogonalize Q_{j+1} to Q_j
Recompute M-orthogonal factor of Q_{j+1}
Until orthogonal factorization occurs in one step

End loop

FIG. 11. *Spectral transformation block Lanczos algorithm preserving semi-orthogonality.*

subproblem. In this environment there are three ways in which a given Lanczos run can terminate:

1. All eigenvalues required for this subproblem have converged.
2. The B_{j+1}-block is ill conditioned or singular. In this case a continuation of the Lanczos run is either numerically difficult or impossible. Singular or ill conditioned B_{j+1}-blocks can be encountered for the following reasons:
   - The shift is very close to an eigenvalue.
   - The effective space of Lanczos vectors is exhausted—we cannot compute more orthogonal vectors than the problem has finite eigenvalues.
   - Dependencies within the starting block cause a singular B_{j+1} at some later stage.
3. Eigenvalues farther from the shift appear to be converging slowly. The estimated cost for computing them in the current Lanczos run is great enough that a new shift should be chosen.

The first of these is easy to detect in most cases. There is a minor complication when we want the eigenvalues closest to some specified value because we do not know in advance how many eigenvalues are required on each side of ξ. At a given shift our

conservative code looks for as many eigenvalues as are required to complete the total, even if these may represent more than what is needed on one side of ξ. As a result, we may not terminate as early as we might with hindsight.

Breakdown in the recurrence is perhaps more likely than might otherwise be expected. The first of the causes we try to avoid during the shift selection process; the second occurs primarily during user evaluation of the code, when it is not uncommon to be faced with problems like finding all of the eigenvalues of 7×7 matrices using a blocksize of 5. The third we have never seen. Breakdown is detected by one of two mechanisms—the norm of the residual block is very small compared to the norm of the diagonal block or the off-diagonal block is ill conditioned and presumed rank-deficient. We use a relative norm of $1/\sqrt{\epsilon}$ for the first case. For the second we compute, at each step, the extreme singular values of the off-diagonal block B_i; we terminate if the condition number of $B_i \geq \frac{1}{\epsilon}$. We really want only the condition number of B_i, but the cost of a singular value decomposition of a $p \times p$ matrix is trivial compared to the cost of an $n \times n$ sparse block solve.

The most common reason for termination is that computing more eigenvalues in the current run is inefficient. Normally, eigenvalues far from the shift converge slowly and require a large number of steps. Usually the fastest convergence occurs early, with the number of eigenvalues converging per step tapering off as the length of the run increases. Initially the cost *per eigenvalue* decreases rapidly, as the cost of the factorization is amortized over several eigenvalues. Later, as the convergence rate slows and the other costs increase, the average cost also increases. Our goal is to stop at the minimum average cost.

The cost of a Lanczos run depends on a number of parameters, each a function of the number of steps taken. The factorization typically represents the largest single cost, but it occurs once. There is a large constant cost per step, comprising the matrix-block solve and multiplication and other operations in the recurrence. The cost of the eigenanalysis of T_j increases quadratically in the number of block steps. Inasmuch as the eigenvalue nearest the shift is usually the first to converge, and dominates the reappearance of banished subspaces, the frequency with which partial reorthogonalization is needed is generally independent of the number of eigenvalues that have converged and so represents another quadratic term. Terminating the run by computing the converged eigenvectors from the Lanczos vectors is a cubic term.

We determine when to terminate a given Lanczos run by modeling the cost of continuing the recurrence beyond the current step. The residual bounds estimating the accuracy of yet unconverged eigenvalues are monitored step by step; the observed changes are used to estimate future convergence. We attempt to locate a point in an individual run where the average cost per eigenvalue is minimized. This is itself a heuristic attempt to minimize the average cost for all eigenvalues. The effectiveness of the heuristic is demonstrated for a particular example in Table 8.

TABLE 8
Comparison of variations on termination model.

Matrix	Standard Strategy			Termination Early			Termination Late		
	Shifts	Block steps	cpu cost	Shifts	Block steps	cpu cost	Shifts	Block steps	cpu cost
BCSST_26[a]	9	181	177.2	10	209	200.5	7	182	206.8
PLAT1919[b]	21	595	706.5	33	735	736.1	19	696	956.3

[a] blocksize 3, lowest 200 modes.
[b] blocksize 3, all modes in [.000025, .24].

We assume that a measure of the real user cost, including i/o, is available. We use this in a cubic model of cost, from which we obtain a least squares fit to the real cost over the preceding ten steps. From this model we predict the real cost over the next few steps. The cost of the final step, computing the eigenvectors, is estimated from measurements of components of the computation as they appear elsewhere in the recurrence. To start the process, we require that a certain minimum number of steps be taken. The number required is a function of blocksize and the type of problem, as indicated in Table 9. (The values in Table 9 are heuristic values derived from extensive empirical testing.)

The rate of convergence for the as yet unconverged eigenvalues is estimated by taking a weighted geometric average of the change in accuracy of the first unconverged Ritz value over the previous five steps. From this, we extrapolate to estimate the accuracy of the unconverged eigenvalues over a small number of additional steps. The number of extrapolated steps is also a function of blocksize and the type of problem; the actual values used are given in Table 9. We continue the Lanczos run if the estimated average cost per eigenvalue decreases for any of the steps over which we extrapolate convergence. In addition, if we predict that all of the eigenvalues remaining to be computed will converge in the steps corresponding to twice the number of steps given in Table 9, we continue the recurrence to avoid computing another factorization.

TABLE 9
Steps to initialize cost model and over which convergence is extrapolated.

	Vibration		Buckling		
			$\leq$ 10 modes	$>$ 10 modes	
Blocksize	Initial steps	Extrapolation steps	Initial steps	Initial steps	Extrapolation steps
1	35	6	15	30	6
2	20	4	15	25	6
3	20	2	10	25	4
≥ 4	15	2	10	10	4

Our experience with this scheme is that the cost curve is relatively flat near its minimum, making the choice of where to stop appear to be flexible. This is misleading; the global minimum is quite sensitive to the local choice. To demonstrate the value of a well-tuned dynamic scheme for evaluating cost, we include some simple experiments here. We modified our standard scheme to make it terminate early and to force it to run ten steps beyond where it would normally stop. The results are given in Table 8 and show some sensitivity to small changes in the stopping procedure.

4.5. Choice of blocksize and starting block. The two largest benefits of the block algorithm are in i/o cost reduction and in treating multiple eigenvalues. However, the costs of obtaining the M-orthogonal factorization and of the eigenanalysis of T_j increase quadratically with the blocksize p. In general, it is best to choose a blocksize as large as the largest expected multiplicity if eigenvalues of moderate multiplicities are expected. This is particularly important if many clusters of eigenvalues are expected (Table 12). A blocksize of 6 or 7 works well in problems with rigid body modes. We rarely find that $p > 10$ is cost-effective.

The effect of input/output cost is considerable. Within the MacNeal–Schwendler NASTRAN product, which runs on a variety of commercial systems, extensive testing resulted in a default blocksize of 7 on all systems. Input and output is particularly expensive within NASTRAN. In an environment in which input/output cost is less costly, a blocksize of 3 was found to be more effective. We provide our results on a

small number of experiments in §5; it is likely that the optimal blocksize would change on other systems.

One would like to start the Lanczos algorithm with a good guess at a solution. We begin the first Lanczos run with a randomly generated starting block. Thereafter, the approximate eigenvectors (Ritz vectors) from unconverged Ritz values are available as estimates of the next eigenvectors to be found. At the time that the eigenvectors of T are available, we do not know where the next shift will be taken. Therefore, we take a starting block built from all of these Ritz vectors. If t vectors are available, each column in the starting block is taken to be the sum of t/p Ritz vectors. We fill the block out randomly when $t < p$. We adopted this approach after extensive experiments comparing various choices of starting blocks, including mixtures of Ritz vectors and random components. We did not find a significant change in the overall cost of the eigensolution with any of the approaches.

5. Experimental results. The algorithm described in the paper was developed as a general purpose eigensolver for the MacNeal–Schwendler Corporation's structural engineering package NASTRAN [18]. One of the goals in the software design was to make the eigensolver independent of the form of the sparse matrix operations representing the matrices involved: the matrix-block products, triangular block solves, and sparse factorizations. The eigensolver has been used in MSC NASTRAN with two different approaches to the sparse linear equations involved, a profile and a sparse multifrontal factorization. In both cases the factorization and solve modules are the standard operations of MSC NASTRAN, used directly by the eigensolver. The code has also been incorporated in four other structural engineering packages and in mathematics libraries supplied by Boeing Computer Services (BCSLIB-EXT)[1] [1] and Convex Computer Corporation (Veclib). In all of these implementations the sparse linear equations are solved with vectorized multifrontal codes based on the work in [2]–[4]. The multifrontal code computes a stable symmetric indefinite factorization, as described in [26].

In this section we report on experiments using our standard eigensolver from BCSLIB-EXT. The experiments were all performed on a Sun 4/690 workstation with 64 megabytes of main memory. The codes are all written in Fortran 77, and were run with the "-O" optimization option of the Sun Fortran compiler, which is quite effective with the inner loops of the numerical operations. We note that our code is always a block code, even when run with blocksize 1. This results in greater costs for the analysis of the tridiagonal system, where the results of Parlett and Nour-Omid would be available [33]. However, the cost of the tridiagonal analysis is less than 1% in general.

The test problems are drawn from the symmetric eigenproblems from the Harwell–Boeing test collection [11]. Our code has been used to solve eigenproblems with more than a million degrees of freedom, but the largest problem in the current test collection is of order 15,439 and most of the problems are much smaller. As a result, the order independent costs of the Lanczos algorithm, primarily the analysis of the block tridiagonal systems, are more important than they would be in large production problems. For most of the examples, we report the costs of the required eigenanalysis as a function of blocksize. For the largest problem we also report the breakdown of the cost in terms of the functional operations of the code.

[1] BCSLIB-EXT is available at no cost on all Cray Research, Inc. computers.

5.1. Some empirical examples. Throughout this paper we have used some of the problems from the Harwell–Boeing test collection [11] to demonstrate particular aspects of our algorithms. We close by using a small subset to illustrate some of the global behavior of the code, particularly as it concerns aspects over which the user exercises control. We chose four test problems, listed in Table 10, which were collected from actual industrial or research applications.

TABLE 10
Test problems.

Matrix	order	Nonzeros in K	Nonzeros in M	Description
BCSST_08	1074	7017	1074	television station
BCSST_25	15439	133840	15439	76-story skyscraper
BCSST_26	1992	16129	1922	nuclear reactor containment floor
PLAT1919	1919	17159	—[a]	Atlantic and Indian Oceans

[a] ordinary eigenvalue problem.

Two of the problems have been used as the primary examples in this paper. They are BCSST_26, a model of a nuclear reactor containment floor used for seismic analysis, and PLAT1919, a finite difference model of tidal currents. These models were included in the test collection because of the large number of eigenpairs that were required of each. In both cases the number of modes is large because the analysis depended on knowing *all* of the modes in specified intervals.

Details of the eigenanalysis of the nuclear reactor containment floor problem, as a function of blocksize, are given in Table 11. These results exhibit a pattern common to all of the problems: The number of factorizations and Lanczos runs decrease rapidly as the blocksize increases; the cost of the eigenanalysis initially decreases as well, but then increases. This reflects the fact that as the blocksize increases, the length of the Lanczos runs increase in terms of the dimension of $\mathcal{Q}_j$. Longer runs involve greater costs, particularly for maintaining semi-orthogonality and for the back transformation of the eigenvectors. For these relatively small matrices, the costs of longer runs begin rather early to dominate the costs of factoring and applying the matrix operators. For reference, an analysis with a single Lanczos run with a blocksize of 3 had a cost of 543.4 for this problem, nearly three times the cost of the analysis with shifting.

TABLE 11
Computation of 200 *eigenvalues from* BCSST_26 (*shift statistics*).

Block-size	cpu cost	Factor-izations	Runs	Block steps	Block solves
1	131.1	12	12	440	475
2	143.3	11	10	254	283
3	188.5	9	8	181	204
4	272.8	8	8	182	205
5	346.3	7	7	162	182
6	301.2	4	4	93	104

The desired eigenvalues in the oceanography problem are very much in the interior of the spectrum. There are 818 eigenvalues above and 465 eigenvalues below the values we want. This problem was analyzed without the use of the spectral transformation in [5], [23]. Without shifting, it was barely possible to compute the eigenvalues in the interval [.0001, .24]; the eigenvalues in [.000025, .0001] were also of interest, but

were impossible to compute. Secondly, all the eigenvalues, except a singleton at zero, are positive and occur in pairs. These multiple eigenvalues can play havoc with an ordinary, point Lanczos algorithm. With either a blocksize of 1 or 2, it is difficult for a code to be sure that it has exhibited the full multiplicities of the eigenvalues—the shifting strategy must be prepared to assist. Even with shifting, the single vector code of Ericsson and Ruhe [12], [15] was unable to cope with the multiplicities of the eigenvalues [25].

Table 12 shows the difficulty that arises with rank determination when the blocksize is the same as the multiplicity of the eigenvalues. When we use a blocksize of 2, we cannot distinguish between doubletons that are truly doubletons and those that are only two of a larger number of copies of a multiple eigenvalue. As a result, our code makes a large number of reruns to ensure that we have the full multiplicities of eigenvalues. This is shown by the discrepancy between the number of factorizations and the number of runs. Although the reruns incur no new cost for factorizations, they do require more extensive use of external selective orthogonalization than would an ordinary run. Surprisingly, the point version of the code is able to cope well with this problem. As expected, blocksize larger than the multiplicity of 2 have no difficulties.

TABLE 12
Computation of 636 *eigenvalues from* PLAT1919 (*shift statistics*).

Block-size	cpu cost	Factor-izations	Runs	Block	
				steps	solves
1	659.6	33	33	1461	1526
2	1101.9	19	35	1068	1137
3	696.0	21	22	595	638
4	825.2	16	16	427	458
5	953.8	15	14	362	389
6	1043.6	12	12	291	314

BCSST_08 is a model of a building housing a television studio. Its claim to fame is the presence of isolated double and near triple eigenvalues. The lowest 24 eigenvalues are given in Table 13. The close eigenvalues cause relatively slow convergence, which causes our code to make more runs than we might expect. This problem can be solved easily enough with a single run, but at increased cost. We note that the multiple eigenvalues provide some challenges for blocksizes of 1 or 2. Details are given in Table 14.

TABLE 13
Lowest 26 *eigenvalues of* BCSST_08.

i	λ_i
1	6.900
2	18.14206
3	18.1423664462086
4	18.1423664462086
5	84.78615
6	84.7864335537914
7	84.7864335537914
8	85.54

i	λ_i
9	91.05
10	93.45
11	130.9
12	131.5
13	132.9
14	136.2
15	137.2
16	138.4

i	λ_i
17	138.7
18	139.6
19	140.6
20	141.1
21	141.566
22	141.638
23	142.19
24	142.642

BCSST_25 is an incomplete seismic model of the Columbia Center, a 76-story skyscraper in Seattle, Washington. The spectrum of this model is pathologically difficult—the lowest 132 eigenvalues are listed in Table 15. For reference, the largest eigenvalue of this structure is 1.51×10^8.

TABLE 14
Computation of lowest 20 *eigenvalues from* BCSST_08 (*shift statistics*).

Block-size	cpu cost	Factor-izations	Runs	Block steps	Block solves
1	37.6	5	5	179	193
2	26.0	4	3	63	71
3	22.2	2	2	39	44
4	34.3	4	3	46	54
5	33.4	3	2	31	36
6	37.9	2	2	29	34

TABLE 15
Lowest 132 *eigenvalues of* BCSST_25.

i	λ_i
1	9.6140×10^{-4}
2	9.7948×10^{-4}
3	9.7961×10^{-4}
4	9.8380×10^{-4}

i	λ_i
5	9.85801×10^{-4}
⋮	⋮
68	9.85803×10^{-4}

i	λ_i
69	9.86240×10^{-4}
⋮	⋮
132	9.86243×10^{-4}

The smallest eigenvalues are nearly negligible when compared to the largest eigenvalue and they are very close to one another. Our code determines clusters of eigenvalues based on its accuracy tolerance, which defaults to 2.31×10^{-11} in IEEE arithmetic. We apply this tolerance to the transformed eigenvalues, which are *not* close enough to be treated as a cluster or even as two clusters and four isolated values. (Note that if we applied the tolerance to the untransformed eigenvalues, all of these values would be a cluster, which is not appropriate.) As a result, this problem counters our usual shifting strategy—in this case we must take a shift very close to the eigenvalues in order to overcome the very poor separation and slow convergence. This distribution, eigenvalues almost, but not quite, in a cluster represents a worst case. Table 16 documents the performance of our code on this problem. We see that for this problem the costs of larger blocksizes are more than offset by the additional power they provide in attacking the very close and large clusters of eigenvalues. In Table 17 we present the breakdown of cost by function within the algorithm for this, the largest of our test problems. This breakdown is typical of larger problems in that neither the cost of analyzing T nor of choosing shifts is significant. It is atypical in that the startup cost is high, a result of there being a large number of vectors involved in external selective orthogonalization.

TABLE 16
Computation of 132 *eigenvalues from* BCSST_25 (*shift statistics*).

Block-size	cpu cost	Factor-izations	Runs	Block steps	Block solves
1	6372.8	7	7	586	606
2	5451.8	9	9	293	320
3	3683.3	5	5	158	172
4	3935.4	5	5	126	140
5	4063.3	5	5	108	122
6	2743.3	2	2	56	61

5.2. Summary. The results in the previous section illustrate some of the characteristics of the shifted block Lanczos algorithm. Only BCSST_25 is large enough to begin to demonstrate the behavior of the algorithm on large problems. For larger

TABLE 17
Computation of lowest 132 *eigenvalues from* BCSST_25 (*cost breakdown*).

Percent of Cost							
Block-size	Recur-rence	Factor-ization	Re-orthog.	Block tridiag.	Eigen-vector	Start up	Shift select.
1	25	15	44	0	4	12	0
2	28	22	30	0	5	15	0
3	33	18	33	1	10	4	0
4	33	16	35	1	10	4	0
5	34	16	35	1	10	5	1
6	32	10	40	1	16	1	1

problems we expect to see the cost of the factorization and linear equation solutions to increase faster than linearly. Assuming that the eigenvalue distributions do not change, the cost of reorthogonalization, of generating the starting block, and of the eigenvector computation will increase linearly with the change in problem size. The block tridiagonal eigenanalysis and the shift selection should remain constant and their contributions to cost will become even smaller. We note that the cost of the necessary reorthogonalizations is an important fraction of the cost—this is a strong argument for preserving only *semi-orthogonality* rather than complete orthogonality. We remind the reader that our cost measures include a factor for i/o traffic, an essential ingredient in preserving semi-orthogonality.

The reader will see the difficulty in making an a priori choice of blocksize. The advantages and disadvantages of the block algorithm are clearly demonstrated, but we see no optimal choice for blocksize. A choice of three is always good on these problems on our Sun workstation, but is likely to be less than optimal for a vibration problem with six rigid body modes. Systems that impose higher costs for i/o will make higher blocksizes more effective, particularly when the problems are large enough that the factored matrices must reside on secondary storage.

These issues should be kept in the perspective of the power of the spectral transformation. None of the problems described here is solvable in any practical sense using the naive reduction to standard form. For example, the oceanography problem, PLAT1919, was analyzed in [5], [23] without any transformation—the desired eigenvalues were not close to appearing after N steps. (In unreported experiments, $3N$ steps had resulted in little improvement.) Although it is possible to solve some of the simpler problems by inverting the problem, as in (2), this is clearly not sufficient for all of the problems. The oceanography problem, PLAT1919, is singular, so some nontrivial shift is required. Even with a shift at the lower endpoint, .000025, a single Lanczos run to compute the lowest 200 eigenvalues above this point had a cost of 5382 for blocksize 3. In contrast, our standard shifted code with the same blocksize had a cost of 696 for computing *all* 636 desired eigenvalues. The Columbia Center model has the same characteristics. The naive reduction would result in a problem with separations of 10^{-13} for the "well-separated" eigenvalues; the simple reciprocal transformation would be clearly inadequate to begin to solve this problem. It is only with the combined power of the block Lanczos algorithm and the spectral transformation that we can solve these problems in a reasonable amount of time.

A. Matrix inertias. We need to interpret the number of negative eigenvalues of $K - \sigma M$ and $K - \sigma K_\delta$ in terms of the eigenvalues of the original vibration or buckling problems. The result we want to prove follows in Table 18. We use this result to conclude that

$$\nu(K - \sigma_2 M) - \nu(K - \sigma_1 M) = \text{number of eigenvalues in } (\sigma_1, \sigma_2),$$

where we assume that $\sigma_2 > \sigma_1$. In the case of buckling analyses we further assume that both σ_1 and σ_2 have the same sign.

TABLE 18
Interpretation of $\nu(K - \sigma M)$ *or* $\nu(K - \sigma K_\delta)$.

Vibration analysis:	
M positive definite	# of eigenvalues $< \sigma$
M positive semidefinite	(# of eigenvalues $< \sigma$) $+ \gamma$
	$\gamma = \begin{cases} 0 & \text{some cases} \\ \dim(\mathcal{N}(M)) & \text{other cases} \end{cases}$
Buckling analysis:	
K positive definite	# of eigenvalues in $(0, \sigma)$ or $(\sigma, 0)$
K positive semidefinite	(# of eigenvalues in $(0, \sigma)$ or $(\sigma, 0)$) $+ \gamma$
	$\gamma = \begin{cases} 0 & \sigma \text{ of one sign} \\ \dim(\mathcal{N}(K)) & \sigma \text{ of other sign} \end{cases}$

There are four cases, which will be considered in pairs. In all cases we assume that the problem is a definite generalized symmetric eigenproblem, i.e., that there exists some linear combination $\alpha K + \beta M$ that is positive definite.

A.1. $Kx = \lambda Mx$ **with** M **positive definite.** We can apply the obvious reduction to standard form. The eigenvalues of $Kx = \lambda Mx$ are the same as the eigenvalues of $C = L_M^{-1} K L_M^{-T}$, where L_M is the Cholesky factor of M. It follows that the number of eigenvalues of C less than σ is the same as the number of eigenvalues of $Kx = \lambda Mx$ less than σ. But $C - \sigma I$ is congruent to $L_M(C - \sigma I)L_M^T$ and this is simply $K - \sigma M$. Thus, the decomposition of $K - \sigma M$ gives the number of eigenvalues less than σ. Obviously, the interpretation of the inertia has the same meaning here as in the ordinary eigenvalue problem.

A.2. $Kx = \lambda Mx$ **with** M **positive semidefinite.** Signs must be assigned to the infinite eigenvalues when M is singular. Assume that M is positive semidefinite, with p zero eigenvalues. Then there exists a nonsingular matrix W so that WMW^T is the two-by-two block-partitioned matrix

$$WMW^T = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix},$$

where I is an $(n-p) \times (n-p)$ identity matrix. Partition $WKW^T = C$ conformally as

$$WKW^T = \begin{pmatrix} C_{11} & C_{21}^T \\ C_{21} & C_{22} \end{pmatrix}.$$

Some linear combination $\alpha K + \beta M$ is positive definite, from which it follows that $\alpha(WKW^T) + \beta(WMW^T)$ is positive definite. But a positive definite matrix has positive definite principal minors, which implies that αC_{11} is positive definite. Let y satisfy $(WKW^T)y = \lambda(WMW^T)y$ and partition y conformally as $[y_1, y_2]^T$. Then

$$C_{11}y_1 + C_{21}^T y_2 = 0 \tag{31}$$

and

$$C_{21}y_1 + C_{22}y_2 = \lambda y_2. \tag{32}$$

Equation (31) then implies that

$$y_1 = -C_{11}^{-1} C_{21}^T y_2.$$

Substituting in (32), we obtain

$$(C_{22} - C_{21} C_{11}^{-1} C_{21}^T) y_2 = \lambda y_2.$$

Thus, the finite eigenvalues of $Kx = \lambda Mx$ are the eigenvalues of the Schur complement of C_{11} in C.

By Sylvester's theorem the inertia of $(K - \sigma M)$ is the same as the inertia of $WKW^T - \sigma WMW^T$. But the partitioned form of the LDL^T decomposition of $WKW^T - \sigma WMW^T$ has as its (1,1) block the decomposition of C_{11}, and as its (2,2) block the decomposition of $(C_{22} - C_{21}C_{11}^{-1}C_{21}^T) - \sigma I$. The inertia for the entire matrix is offset by the inertia of the (1,1) block. The offset is constant—it describes the sign given to the infinite eigenvalues. That all of the infinite eigenvalues have the same sign is due to the fact that a positive definite linear combination of K and M exists, that is, that the problem is a definite generalized symmetric eigenproblem [13]. The difference between $\nu(K - \sigma_1 M)$ and $\nu(K - \sigma_2 M)$ will still be the number of eigenvalues in $[\sigma_1, \sigma_2)$, since the constant term cancels.

Furthermore, in vibration analysis, we know that both K and M are positive semidefinite. It follows that both α and β will be positive when M is only semidefinite. The positive semidefiniteness of K then implies that C_{11} is a positive definite matrix, so $\nu(C_{11}) = 0$. Thus, the inertia of the factored matrix retains exactly the same meaning for the positive semidefinite vibration case as for the positive definite case.

A.3. $Kx = \lambda K_\delta x$ **with** K **positive definite.** In buckling analysis, only K has any definiteness properties. We can invert the problem when K is positive definite. Thus

$$Kx = \lambda K_\delta x$$

implies

$$K_\delta x = \frac{1}{\lambda} Kx = \mu Kx,$$

and all the eigenvalues μ in the second equation are finite. This transformed problem is in the standard (K_δ, K) form in which the right-hand side matrix, K, is positive definite. We will determine the number of eigenvalues of (K_δ, K) that lie in the image of the interval of interest in the original problem. Thus, to determine the number of eigenvalues of $Kx = \lambda K_\delta x$ less than σ, we find the number of eigenvalues of the inverted problem (K_δ, K) in the interval(s) in the variable $\mu = \frac{1}{\lambda}$ that corresponds to the interval $(-\infty, \sigma)$ in the variable λ.

There are three subcases that must be considered. The first is the case $\sigma = 0$. The interval $(-\infty, 0)$ for λ is mapped to the interval $(-\infty, 0)$ in $\frac{1}{\lambda}$. Thus, the number of negative eigenvalues of $Kx = \lambda K_\delta x$ is the same as the number of eigenvalues of (K_δ, K) less than 0. This is simply the number of negative eigenvalues of $K_\delta, \nu(K_\delta)$.

The second case is the case $\sigma < 0$. The transformation from λ to $\frac{1}{\lambda}$ transforms σ to $\frac{1}{\sigma}$. The number of eigenvalues in $(-\infty, \sigma)$ is the same as the number of eigenvalues of (K_δ, K) in the interval $(\frac{1}{\sigma}, 0)$. This is

$$\nu(K_\delta) - \nu\left(K_\delta - \frac{1}{\sigma} K\right),$$

which, because σ is negative, is the same as

$$\nu(K_\delta) - \nu(K - \sigma K_\delta).$$

Note that the number of eigenvalues between σ and 0 is simply $\nu(K - \sigma K_\delta)$.

The third case is $\sigma > 0$. In this case, the interval $(-\infty, \sigma)$ in λ must be treated as the union of the interval $(-\infty, 0)$ and the interval $[0, \sigma)$. There are $\nu(K_\delta)$ eigenvalues in the first subinterval. The second subinterval is transformed into $(\frac{1}{\sigma}, +\infty)$. The union has

$$\nu(K_\delta) + \pi\left(K_\delta - \frac{1}{\sigma}K\right)$$

or

$$\nu(K_\delta) + \nu(K - \sigma K_\delta)$$

eigenvalues. Even in this case $\nu(K - \sigma K_\delta)$ is the number of eigenvalues between 0 and σ.

The buckling problem will have infinite eigenvalues if K_δ is singular. However, the signs of these eigenvalues are irrelevant to the interpretation of the inertias because the interpretation always considers only finite subintervals.

A.4. $Kx = \lambda K_\delta x$ **with** K **positive semidefinite.** The most general case we consider is a buckling analysis in which K is only positive semidefinite. We combine the analysis for the semidefinite vibration case with the positive definite buckling case to assign signs to the zero eigenvalues.

We assume K is semidefinite, with P zero eigenvalues. As before, there exists a nonsingular matrix $\widehat{W}$ such that

$$\widehat{W}K\widehat{W}^T = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}.$$

Partition $\widehat{W}K_\delta\widehat{W}^T = E$ conformally as

$$\begin{pmatrix} E_{11} & E_{21}^T \\ E_{21} & E_{22} \end{pmatrix}.$$

The eigenvalues of $KX = K_\delta X\Lambda$ are those of $\widehat{W}K\widehat{W}^TY = \widehat{W}K_\delta\widehat{W}^TY\Lambda$. Let y be an eigenvector, partitioned conformally as $[y_1, y_2]^T$. Then

$$E_{11}y_1 + E_{21}^Ty_2 = 0 \tag{33}$$

and

$$\lambda(E_{21}y_1 + E_{22})y_2 = y_2.$$

As before the (1,1) block of the transformed linear combination, βE_{11}, is a positive definite matrix. Equation (33) then implies that

$$y_1 = -E_{11}^{-1}E_{21}^Ty_2,$$

or

$$\lambda(E_{22} - E_{21}E_{11}^{-1}E_{21}^T)y_2 = y_2.$$

Thus, the finite, nonzero eigenvalues of $Kx = \lambda K_\delta x$ are the reciprocals of the nonzero eigenvalues of the Schur complement of E_{11} in E.

The partitioned form of the LDL^T decomposition of $\widehat{W}K\widehat{W}^T - \sigma\widehat{W}K_\delta\widehat{W}^T$ has as its (1,1) block the decomposition of $-\sigma E_{11}$, and as its (2,2) block the decomposition of $I - \sigma(E_{22} - E_{21}E_{11}^{-1}E_{21}^T)$. The Schur complement block is in the form of §A.3, taking the identity matrix as M. Again, the inertia of the full matrix is the inertia of $I - \sigma(E_{22} - E_{21}E_{11}^{-1}E_{21}^T)$ offset by the inertia of the (1,1) block. Notice that the offset depends on the sign of the shift—it describes the signs of the eigenvalues of $-\sigma E_{11}$. Because E_{11} is definite, either all the eigenvalues of $-\sigma E_{11}$ are positive or all are negative. Thus, the offset will be zero for shifts of one sign and nonzero for shifts of the other sign. Still, the difference between $\nu(K - \sigma_1 K_\delta)$ and $\nu(K - \sigma_2 K_\delta)$ will still be the number of eigenvalues in $[\sigma_1, \sigma_2)$, as long as both shifts have the same sign. The dimension of the nullspace of K, $\nu(E_{11})$, is often known adventitiously; if not, it can be estimated by factoring $K - \rho I$, where ρ is chosen smaller than the least nonzero eigenvalue of K, but large enough so that the factorization is stable.

Acknowledgments. The authors would like to thank David Scott for his participation in the initial design of this code and Thomas Ericsson and Bahram Nour-Omid for many enlightening discussions during the preparation of this code. We particularly thank Beresford Parlett for his interest and intellectual support, which led to important changes, even after we thought our work was done. We also are grateful to Louis Komzsik and The MacNeal–Schwendler Corporation for their support of this work. Finally, Linda Kaufman provided a fine example of careful and thorough editing.

REFERENCES

[1] *The Boeing Extended Mathematical Subprogram Library*, Boeing Computer Services, P.O. Box 24346, Seattle, WA 98124-0346, 1989.

[2] C. C. ASHCRAFT, *A vector implementation of the multifrontal method for large sparse symmetric positive linear systems*, Tech. Report ETA-TR-51, Boeing Computer Services, P.O. Box 24346, Seattle, WA 98124-0346, 1987.

[3] C. 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.

[4] C. C. ASHCRAFT, R. G. GRIMES, J. G. LEWIS, B. W. PEYTON, AND H. D. SIMON, *Progress in sparse matrix methods for large linear systems on vector supercomputers*, Internat. J. Supercomput. Appl., 1 (1987), pp. 10–30.

[5] A. CLINE, G. GOLUB, AND G. PLATZMAN, *Calculations of normal modes of oceans using a Lanczos method*, in Sparse Matrix Computations, J. Bunch and D. Rose, eds., Academic Press, New York, 1976, pp. 409–426.

[6] J. CULLUM AND W. E. DONATH, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large sparse real symmetric matrices*, Proc. 1974 IEEE Conference on Decision and Control, pp. 505–509, IEEE Computer Society, 1974.

[7] J. CULLUM AND R. WILLOUGHBY, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 1 Theory*, Birkhäuser, Boston, 1985.

[8] ———, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 2 Users Guide*, Birkhäuser, Boston, 1985.

[9] ———, *Computing eigenvalues of large matrices, some Lanczos algorithms and a shift and invert strategy*, in Advances in Numerical Partial Differential Equations and Optimization: Proc. 5th Mexico–United States Workshop, S. Gomez, J. P. Hennart, and R. A. Tapia, eds., Proc. in Applied Mathematics, Vol. 47, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1991.

[10] J. W. DANIEL, W. B. GRAGG, L. KAUFMAN, AND G. W. STEWART, *Reorthogonalizations and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[11] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.

[12] T. ERICSSON, *User Guide for STLM*, Tech. Report UMINF-96.82, University of Umea, Umea, Sweden, 1982.

[13] ———, *A generalized eigenvalue problem and the Lanczos algorithm*, in Large Scale Eigenvalue Problems, J. Cullum and R. Willoughby, eds., North–Holland, Amsterdam, 1986.

[14] T. ERICSSON AND A. RUHE, *The spectral transformation Lanczos method*, Math. Comp., 34 (1980), pp. 1251–1268.

[15] ———, *STLM-A Software Package for the Spectral Transformation Lanczos Algorithm*, Tech. Report UMINF-101.82, University of Umea, Umea, Sweden, 1982.

[16] B. S. GARBOW, J. M. BOYLE, J. J. DONGARRA, AND C. B. MOLER, *Matrix Eigensystem Routines—EISPACK Guide Extension*, Vol. 51 of Lecture Notes in Computer Sciences, Springer-Verlag, Berlin, 1977.

[17] G. GOLUB AND R. UNDERWOOD, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. Rice, ed., Academic Press, New York, 1977.

[18] R. GRIMES, J. LEWIS, L. KOMZSIK, D. SCOTT, AND H. SIMON, *Shifted block Lanczos algorithm in MSC/NASTRAN*, in Proc. MSC/NASTRAN User's Conference, Los Angeles, 1985.

[19] R. GRIMES, J. LEWIS, AND H. SIMON, *Eigenvalue Problems and Algorithms in Structural Engineering*, in Large Scale Eigenvalue Problems, J. Cullum and R. Willoughby, eds., North–Holland, Amsterdam, 1986.

[20] ———, *A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems*, Tech. Report AMS-TR-166, Boeing Computer Services, P.O. Box 24346, Seattle, WA 98124-0346, 1991.

[21] M. T. JONES AND M. L. PATRICK, *LANZ: Software Solving the Large Sparse Symmetric Generalized Eigenproblem*, Tech. Report NAS1-18605, ICASE, NASA Langley Research Center, Hampton, VA, August 1990.

[22] C. LANCZOS, *An iteration method for the solution of eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Stand., 45 (1950), p. 255.

[23] J. LEWIS, *Algorithms for Sparse Matrix Eigenvalue Problems*, Ph.D. thesis, Dept. of Computer Science, Stanford University, Stanford, CA, 1977.

[24] J. LEWIS AND R. GRIMES, *Practical Lanczos algorithms for solving structural engineering eigenvalue problems*, in Sparse Matrices and Their Uses, I. Duff, ed., Academic Press, London, 1981.

[25] J. LEWIS AND H. SIMON, *Numerical Experience with the Spectral Transformation Lanczos Method*, Tech. Report ETA-TR-16, Boeing Computer Services, P.O. Box 24346, Seattle, WA 98124-0346, 1983.

[26] J. W.-H. LIU, *A partial pivoting strategy for sparse symmetric matrix decomposition*, ACM Trans. Math. Software, 13 (1987), pp. 173–182.

[27] M. NEWMAN AND P. FLANAGAN, *Eigenvalue Extraction in NASTRAN by the Tridiagonal Reduction (FEER) Method—Real Eigenvalue Analysis*, NASA Contractor Report CR-2731, NASA, Washington, D.C., 1976.

[28] B. NOUR-OMID, *The Lanczos Algorithm for Solution of Large Generalized Eigenproblems*, in The Finite Element Method, T. Hughes, ed., Prentice-Hall, Englewood Cliffs, NJ, 1987.

[29] B. NOUR-OMID, B. N. PARLETT, T. ERICSSON, AND P. S. JENSEN, *How to implement the spectral transformation*, Math. Comp., 48 (1987), pp. 663–673.

[30] C. C. PAIGE, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, University of London, London, UK, 1971.

[31] ———, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

[32] B. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[33] B. N. PARLETT AND B. NOUR-OMID, *The use of refined error bounds when updating eigenvalues of a growing symmetric tridiagonal matrix*, Linear Algebra Appl., 68 (1985), pp. 179–219.

[34] B. PARLETT, B. NOUR-OMID, AND Z. A. LIU, *How to Maintain Semi-Orthogonality Among Lanczos Vectors*, Tech. Report PAM-420, Center for Pure and Applied Mathematics, University of California, Berkeley, CA, 1988.

[35] B. PARLETT AND D. SCOTT, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.

[36] Y. SAAD, *On the rates of convergence of the Lanczos and block-Lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[37] D. SCOTT, *Block Lanczos Software for Symmetric Eigenvalue Problems*, Tech. Report ORNL/CSD 48, Oak Ridge National Laboratory, Oak Ridge, TN, 1979.

[38] D. SCOTT, *The advantages of inverted operators in Rayleigh–Ritz approximations*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 68–75.

[39] H. SIMON, *Analysis of the symmetric Lanczos algorithm with reorthogonalization methods*, Linear Algebra Appl., 61 (1984), pp. 101–131.

[40] ———, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–136.

[41] B. T. SMITH, J. M. BOYLE, J. J. DONGARRA, B. S. GARBOW, Y. IKEBE, V. C. KLEMA, AND C. B. MOLER, *Matrix Eigensystem Routines—EISPACK Guide*, Lecture Notes in Computer Sciences, Vol. 6, Springer-Verlag, Berlin, 1976.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 273–283, January 1994

FACTORING SYMMETRIC INDEFINITE MATRICES ON HIGH-PERFORMANCE ARCHITECTURES*

MARK T. JONES† AND MERRELL L. PATRICK‡

Abstract. The Bunch–Kaufman algorithm is the method of choice for factoring symmetric indefinite matrices in many applications. However, the Bunch–Kaufman algorithm uses matrix-vector operations and, therefore, may not take full advantage of high-performance architectures with a memory hierarchy. It is possible to modify the Bunch–Kaufman algorithm so that it uses rank-k updates. However, this straightforward modification allows unrestricted row/column interchanges during the algorithm, thus making it unsuitable for banded and sparse matrix factorization. A new algorithm, based on Bunch–Kaufman factorization, is described that uses rank-k updates to take advantage of high-performance architectures while limiting the number of row/column interchanges. Results from implementations on the CRAY Y-MP and the Alliant FX/8 are presented.

Key words. Bunch–Kaufman factorization algorithm, symmetric indefinite matrices, block algorithms

AMS subject classification. 65F05

1. Introduction.

The Bunch–Kaufman algorithm is considered to be one of the best stable methods for factoring full, symmetric, indefinite matrices [3], [4]. A modified version has been successfully used to factor sparse, indefinite matrices [7]. Recently, Bunch–Kaufman factorization has been shown to be the method of choice for a subset of banded, symmetric indefinite matrices [11]. The Bunch–Kaufman algorithm maintains the symmetry of the matrix during factorization and yields the inertia of the matrix essentially for free, an important consideration for eigenvalue calculations [9].

Much of the recent work in numerical linear algebra has focused on constructing algorithms appropriate for execution on high-performance architectures. Many of these algorithms utilize matrix-matrix operations to exploit the memory hierarchies used in many high-performance architectures [8]. The Bunch–Kaufman algorithm, as formulated in [4], uses matrix-vector operations. For dense matrices, the ratio

$$\frac{\textit{floating point operations}}{\textit{memory references}} \tag{1}$$

is often higher for algorithms that rely on matrix-matrix operations rather than for those that rely on matrix-vector operations. As a result, algorithms that use matrix-matrix operations can better exploit memory hierarchies. For sparse matrix factorization, it has been shown that by using matrix-matrix operations, the ratio

$$\frac{\textit{floating point operations}}{\textit{memory references and other overhead}} \tag{2}$$

* Received by the editors April 9, 1990; accepted for publication (in revised form) May 5, 1992. This research was supported by National Aeronautics and Space Administration (NASA) contract NAS1-18107 and NAS1-18605 and Air Force Office of Scientific Research grant 88-0117 while the authors were in residence at Institute for Computer Applications in Science and Engineering (ICASE), Hampton, Virginia. Additional support was provided by NASA grant NAG-1-466. The first author also received support from the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, contract W-31-109-Eng-38.

† Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439 (mjones@mcs.anl.gov).

‡ Department of Computer Science, Duke University, Durham, North Carolina 27706.

is much smaller than when matrix-vector operations are used and that excellent performance can be achieved on a vector processor [2].

In the LAPACK project a simple modification to the Bunch–Kaufman algorithm allows the use of matrix-matrix operations [1]. However, this algorithm may require up to $n-1$ row/column interchanges during the algorithm. This could result in a large amount of fill-in during banded or sparse factorization.

Version D of the Bunch–Kaufman algorithm,[1] however, can be implemented such that the number of row/column interchanges is bounded by the number of negative eigenvalues [4].[2] We shall give a modification to version D that uses rank-k updates while maintaining the bound on the number of row/column interchanges. This algorithm could be used to factor banded or sparse matrices for which the number of negative eigenvalues is small relative to n.

Block algorithms are briefly discussed in §2. In §3, one of several variations of the Bunch–Kaufman algorithm is reviewed. We describe a new algorithm that uses rank-k updates while minimizing the number of row/column interchanges in §4. Results from implementations of the algorithm are given in §5. Finally, a summary is given in §6.

2. Block algorithms. Linear algebra kernel computations can be placed in three categories: (1) vector operations, e.g., vector inner product; (2) matrix-vector operations, e.g., multiplication of a matrix by a vector; and (3) matrix-matrix operations, e.g., rank-p update of a matrix. These three categories correspond to the three linear algebra subroutine collections: Level 1 BLAS, Level 2 BLAS, and Level 3 BLAS [5], [6], [15]. The computation rates that can be achieved on high-performance architectures are higher for the higher-level BLAS. Two benefits of matrix-matrix operations are (1) better use of the memory hierarchy via the reuse of data in closer, fast memory, and (2) increased flexibility in the way the computation can be structured, allowing for more efficient use of computational units. For example, on the CRAY Y-MP, a register-to-register machine with independent segmented computational units capable of being chained together, the use of matrix-matrix operations allows an increased ability to reuse data that have been loaded into the vector registers and an increased ability to structure the computations to enable maximum chaining between computational units and thus achieve better parallelism among the computational units.

As an example, consider the LDL^T decomposition of a symmetric positive definite full matrix. In the outer product version of LDL^T decomposition given in Fig. 1, the computation of a single pivot column in the loop in steps 2–5 is a vector operation, and the updating of the remaining submatrix in the nested loops in steps 6–10 is a matrix-vector operation. Because each iteration of the outer loop depends on results of previous iterations, if we wish to change the algorithm to use matrix-matrix operations, we must restructure it as a block algorithm [8]. Instead of computing a single pivot column, we compute a block of pivot columns and use this block to update the remaining submatrix. Such an algorithm is given in Fig. 2, where the block size is p. The block pivots are computed by using matrix-matrix operations involving $p \times p$ and $p \times (n-ip)$ matrices. The updating of the remaining submatrix is accomplished with matrix-matrix operations involving $p \times (n-ip)$ and $(n-ip) \times (n-ip)$ matrices.

[1] Four variants, A–D, of the Bunch–Kaufman algorithm were presented in [4].

[2] In many applications, the number of negative eigenvalues is much less than the order of the matrix.

```
1)      DO I = 1, N
    C       Solve for the pivot column
2)          DO J = I + 1, N
3)              V(J) = A(J, I)
4)              A(J, I) = A(J, I)/A(I, I)
5)          ENDDO
    C       Use the pivot column to update the remaining submatrix
6)          DO J = I + 1, N
7)              DO K = J, N
8)                  A(J, K) = A(J, K) − V(K) * A(J, I)
9)              ENDDO
10)         ENDDO
11)     ENDDO
```

FIG. 1. *The LDL^T algorithm.*

```
    C   Code assumes that N is divisible by p
1)      DO I = 1, N, p
    C       A_{i,i} is the ith p × p diagonal block of A
2)          Factor: A_{i,i} = L_{i,i} D_{i,i} L_{i,i}^T and store into A_{i,i}
    C       V_i is a (n − ip) × p matrix
    C       A_{i,2} is the (n − ip) × p matrix beneath A_{i,i}
3)          Solve: L_{i,i} V_i^T = A_{i,2}^T, D_{i,i} Â_{i,2}^T = V_i^T
    C       A_{2,2} is the (n − ip) × (n − ip) matrix in the bottom right corner of A
4)          Update: A_{2,2} = A_{2,2} − Â_{i,2} V_i^T
5)      ENDDO
```

FIG. 2. *The block LDL^T algorithm.*

The partitioning of A at Step i of the block algorithm is given in Fig. 3. Such a restructuring is what we desire for the Bunch–Kaufman algorithm.

3. The Bunch–Kaufman algorithm. The Bunch–Kaufman algorithm factors A, an $n \times n$ real symmetric indefinite matrix, into MDM^T while doing symmetric permutations on A to maintain stability, resulting in

$$PAP^T = MDM^T, \tag{3}$$

where P is a permutation matrix, M is a lower triangular matrix and D is a symmetric block diagonal matrix where the blocks are 1×1 or 2×2. Although several variations of the algorithm exist, the focus here is on Algorithm D from [4] because it is the one for which the number of row/column interchanges is restricted.

The Bunch–Kaufman algorithm maintains stability by using a 2×2 pivot combined with a symmetric permutation on A if a 1×1 pivot is not stable. Because this paper will focus on 1×1 pivots, only stability for these pivots will be discussed in detail. A 1×1 pivot operation for element $a_{i,j}$ at step k takes the form

$$a_{i,j} = a_{i,j} - \frac{a_{i,k} a_{j,k}}{a_{k,k}}. \tag{4}$$

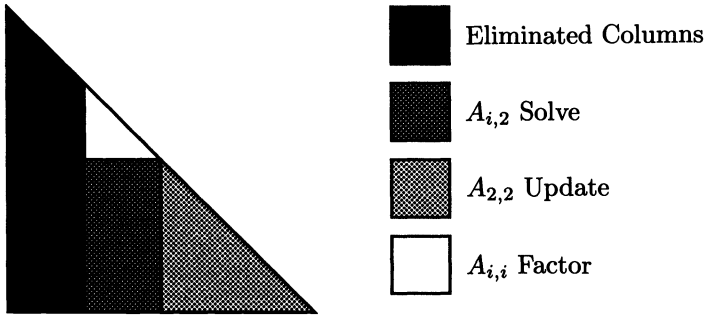


FIG. 3. *Partitioning of A at step i.*

Let μ_k be the maximum of the absolute values of the uneliminated elements at step k. Step 2 of the algorithm, shown in Fig. 4, finds the maximum element, λ_k, in the pivot column. By substituting μ_k and λ_k into (4), the bound on μ_{k+1} becomes

$$\mu_{k+1} \leq \mu_k + \frac{\lambda_k^2}{|\, a_{k,k} \,|} \leq \mu_k \left(1 + \frac{\lambda_k}{|\, a_{k,k} \,|}\right). \tag{5}$$

Step 4 ensures that a 1×1 pivot operation occurs if $\alpha < \frac{|a_{i,i}|}{\lambda_k}$, where the parameter α has been chosen to be 0.525 to maximize stability for Algorithm D [4]. By substituting α into (5),

$$\mu_{k+1} \leq \mu_k \left(1 + \frac{1}{\alpha}\right). \tag{6}$$

Therefore, the bound on the growth of an element resulting from a 1×1 pivot is 2.905.

If the test in step 4 fails, then a subsequent row search and another stability test determines whether a 2×2 pivot and a permutation are necessary. The stability checks and possible permutation at each step of the Bunch–Kaufman algorithm prevent the use of the same straightforward blocking that was used for LDL^T decomposition. Because the stability checks and permutation must be completed before a pivot column is computed, pivot columns cannot be grouped as they were in Fig. 2 without invalidating the bounds on element growth. In the 1×1 pivot case, solving for the pivot columns is a vector operation, and updating the remaining submatrix is a rank-1 update. The situation is slightly better for the 2×2 case for which solving for the two pivot columns is a matrix-matrix operation where the dimensions of the matrices are 2×2 and $2 \times (n - i - 1)$. Updating the remaining submatrix is a rank-2 update.

Two approaches have been taken to implement the Bunch–Kaufman algorithm on high-performance architectures. Kaufman has implemented Algorithm B, an inner-product formulation that uses matrix-vector products, and found that it performs much better than the outer-product formulations on machines like the Alliant and Convex but not on the CRAY X-MP [14]. The inner-product formulation works on rectangular matrices rather than triangular matrices; this is a significant advantage on the Alliant and Convex. Algorithm B, however, has unrestricted row/column interchanges and is, therefore, as formulated, not suitable for band or sparse matrices.

```
1)    DO k = 1, n
2)        λ_k = max_{j=k+1,n} | a_{j,k} |
3)        set r to the row number of λ_k
4)        IF λ_k α <| a_{k,k} | THEN
5)            perform a 1 × 1 pivot operation
6)        ELSE
7)            σ = max_{j=k+1,n} | a_{r,j} |
8)            IF α λ_k^2 < σ | a_{k,k} | THEN
9)                perform a 1 × 1 pivot operation
10)           ELSE
11)               exchange rows and columns r and k + 1
12)               perform a 2 × 2 pivot operation
13)               k = k + 1
14)           ENDIF
15)       ENDIF
16)   ENDDO
```

FIG. 4. *The Bunch–Kaufman factorization Algorithm D.*

In the LAPACK project, the Bunch–Kaufman algorithm has been restructured so that matrix-matrix operations can be used.[3] The LAPACK block algorithm, an outline of which is given in Fig. 5, requires a row/column interchange at every step and is therefore not suitable for band or sparse matrices. The LAPACK implementation allows a specific block size, p, to be specified. The LAPACK implementation requires a scratch space of size $n \times p$, however, if one is willing to slightly degrade performance, it is possible to implement the algorithm with a scratch space of $p \times p + 2n$ [14].

To compare the structural damage that occurs during factorization of a banded matrix when the LAPACK block algorithm and version D of the Bunch–Kaufman algorithm are used, we factored a 256×256 matrix with a semi-bandwidth of 50. The structure of the resulting factors is shown in Fig. 6.

An alternative to using the Bunch–Kaufman algorithm for symmetric banded matrices is to use LU factorization and ignore the symmetry of the matrix. Ignoring the symmetry can cost up to mn, where m is the semi-bandwidth, storage locations, and increase the operation count by a factor of 4. In addition, the inertia of the matrix cannot be determined by using LU factorization, making this an unacceptable choice for some applications. These disadvantages make LU factorization a poor choice for the factorization of symmetric indefinite matrices with a small number of negative eigenvalues.

4. New algorithm. In this section, a modification to the Bunch–Kaufman algorithm is developed that allows pivot columns to be put into a block without row/column interchanges and without updating each column as it is put into the block. The algorithm described in this section[4] is a variant of Algorithm D from [4], and therefore the number of row/column interchanges is bounded by the number of negative eigenvalues of the matrix being factored. Unlike the LAPACK block algorithm, the block sizes vary throughout the execution of the algorithm. The general

[3] The algorithm and code referenced in this paper are from preliminary release 2 of LAPACK.

[4] The authors give three other block algorithms in [10] and have determined that the algorithm described here is the most practical.

```
1)   DO k = 1, n
2)       bs = 0
3)       WHILE bs ≤ p and k + bs ≤ n DO
4)           Update column k + bs using the previous bs columns
5)           Find the largest element in column k + bs, i.e., row r
6)           IF stability tests dictate THEN
7)               Copy row/column r into a work vector
8)               Update this work vector using the previous bs columns
9)               IF stability tests dictate THEN
10)                  ignore this work vector
11)              ELSE IF further tests dictate THEN
12)                  copy this work vector into A as a 1 × 1 pivot
13)              ELSE
14)                  combine column k + bs and the work vector into a 2 × 2 pivot
15)              ENDIF
16)          ENDIF
17)          Increment bs by 1 or 2 (depending on the pivot step)
18)      ENDWHILE
19)      update the remaining submatrix using the block pivot
20)      increment k by bs
21)  ENDDO
```

FIG. 5. *The Bunch–Kaufman block variant in* LAPACK.

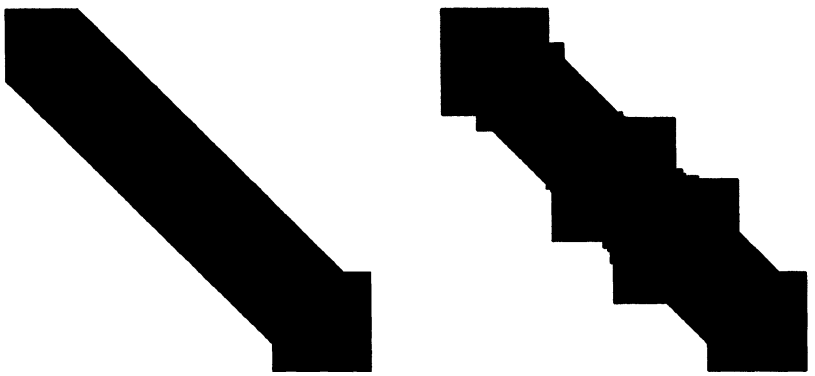


FIG. 6. *Structure of factored matrix for version D and* LAPACK *algorithms.*

strategy is to try to group several 1×1 pivots into a single step in a stable fashion. Because 2×2 pivots involve a permutation of A, they are not candidates for inclusion in a block.

The new algorithm uses an a priori error bound approach to maintain stability. The algorithm computes a bound on the maximum element growth if the next p columns are grouped into a pivot block. The bound is computed *without* actually computing the pivot block or performing any row/column interchanges. If the computed bound is small enough, then the p columns can be used as a pivot block. The

bound can be computed incrementally; if the bound is computed for p columns, then computing it for $p+1$ columns is cheap.

From (6), for p successive 1×1 pivots to maintain stability, the maximum element growth must be bounded by

$$\left(1+\frac{1}{\alpha}\right)^p. \tag{7}$$

We have from (5) that the growth for a 1×1 pivot at step k is

$$\mu_{k+1} \le \mu_k \left(1+\frac{\lambda_k}{|\,a_{k,k}\,|}\right). \tag{8}$$

Therefore, if one had a bound for μ_{k+1} and λ_{k+1}, and knew what $a_{k+1,k+1}$ was, then a bound for μ_{k+2} could be computed.

By factoring the 2×2 diagonal block, denoted $A_{i,i}$ in Fig. 3, $a_{k,k}$ and $a_{k+1,k}$ can be computed. The bound on λ_{k+1}, denoted $\hat{\lambda}_{k+1}$, after a 1×1 pivot at step k, is

$$\hat{\lambda}_{k+1} \le \lambda_{k+1} + \frac{\lambda_k a_{k+1,k}}{a_{k,k}}. \tag{9}$$

The bound on element growth for two columns becomes

$$\mu_{k+2} \le \mu_{k+1}\left(1+\frac{\hat{\lambda}_{k+1}}{a_{k+1,k+1}}\right). \tag{10}$$

To compute an element growth bound for p columns, one must factor the $p\times p$ submatrix, $A_{i,i}$, and search each of the p columns for its largest element. The new algorithm is given in Fig. 7.

The algorithm uses matrix-matrix operations for the computation of the pivot columns and the updating of the remaining submatrix. Like the LAPACK block algorithm implementation, our implementation currently requires $n\times p$ scratch space. However, the same technique used in [14] could be used here to reduce the scratch space. Also, it is possible to search for the λ values of several columns simultaneously. However, if the growth bound becomes large before all the λ's are examined, then some of the searches could be wasted. It is also possible to continue the combination of pivot columns (steps 5–14) beyond the step in which the growth bound becomes too large, in the hope that it will become acceptable again at a future step.

5. Results.

5.1. Uniprocessor implementation. A version of the algorithm described in §4 suitable for matrices with a variable bandwidth was implemented on a CRAY Y-MP. This implementation allows block sizes of up to 5. The matrix-matrix operations are implemented by using loop unrolling, because the structure of the matrices the authors are interested in do not allow for the efficient use of higher-level BLAS. When the maximum block size is fixed at 1, this implementation is identical to the Bunch–Kaufman algorithm.

The CRAY Y-MP is a register-to-register parallel/vector computer with up to eight processors. Each processor has independent, segmented functional units. To demonstrate the benefits of the new algorithm on the CRAY Y-MP, we factored three

```
1)  DO k = 1, n
2)      bs = 0
3)      μ = 1.0
4)      max_growth = 1.0
5)      WHILE (bs ≤ p_max) and (μ ≤ max_growth) DO
6)          Update the factor of the diagonal block A_{k,k}
7)          λ_{k+bs} = max_{j=k+bs+1,n} | a_{j,k+bs} |
8)          DO i = 0, bs − 1
9)              λ_{k+bs} = λ_{k+bs} + λ_i | a_{k+bs,k+i} |
10)         ENDDO
11)         μ = μ(1.0 + λ_{k+bs} / |a_{k+bs,k+bs}|)
12)         max_growth = max_growth(1 + 1/α)
13)         IF (μ ≤ max_growth) THEN bs = bs + 1
14)     ENDWHILE
15)     IF (bs = 0) THEN
16)         σ = max_{j=k+1,n} | a_{j,k} |
17)         depending on σ and λ_k perform a 1 × 1 or 2 × 2 pivot operation
18)         k = k + 1 or k = k + 2
19)     ELSE
20)         Solve for the pivot block using the factored A_{k,k}
21)         Updating the remaining submatrix using the pivot block
22)         k = k + bs
23)     ENDIF
24) ENDDO
```

FIG. 7. *New block variant of Bunch–Kaufman factorization.*

indefinite matrices that arise from an application in structural engineering.[5] Each matrix has ten negative eigenvalues. Results from these computations showing a significant reduction in factorization time as the block size increases, are plotted in Fig. 8. The average block size used during each factorization was very close to the maximum block size specified.

5.2. Multiprocessor implementation. The algorithm in §4 also provides benefits for parallel implementation. The major benefit is a reduction in the number of synchronizations that are necessary because of the use of block operations.

The variable band matrix factorization implementation described in the preceding subsection was explicitly parallelized using the Force [13], a Fortran-based parallel programming language that has been shown to be useful for implementing parallel linear algebra algorithms [12]. The primary source of parallelism is the updating of the reduced matrix. The implementation was run on a four-processor CRAY Y-MP for block sizes 1–5. An examination of the results in Table 1 show that good speedup is maintained as the maximum block size increases for a matrix of order 12,054 and an average semi-bandwidth of 328. At the same time, the new algorithm achieves increased computational efficiency; reduction in the computation time is offset by the reduction in the amount of synchronization needed.

[5] The matrix being factored was actually the difference of two matrices, K and M. The factorization times reported include the cost of computing $K - \sigma M$.

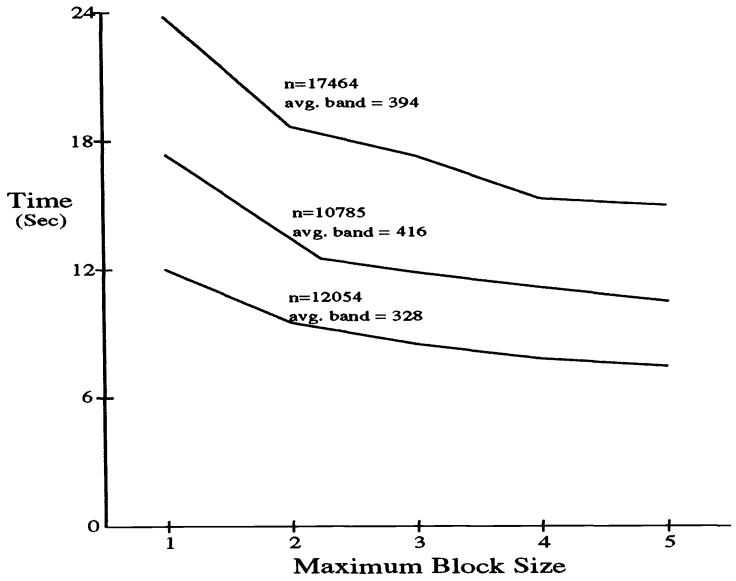


FIG. 8. *Uniprocessor factorization times on the* CRAY Y-MP.

TABLE 1
Effects of different maximum block sizes on parallel performance.

Block Size	1 Processor Time (sec)	2 Processors Time (sec)	3 Processors Time (sec)	4 Processors Time (sec)
1	10.65	5.63	3.94	3.10
2	8.36	4.44	3.12	2.47
3	7.50	3.94	2.76	2.17
4	6.70	3.54	2.46	1.92
5	6.54	3.45	2.39	1.87

5.3. Comparison with other methods. The algorithm presented in this paper is designed for factoring matrices with some structure that must be preserved, but it can, of course, be used to factor full, dense matrices. A comparison with other variants of the Bunch–Kaufman algorithm for factoring full, dense matrices is, therefore, of interest. Such a comparison will give some idea of how much more expensive this algorithm is than an algorithm that can specify the same block size every step, such as the LAPACK version, or how much better the block algorithms can do than algorithms that utilize matrix-vector operations only.

Four codes were used in the comparison: (1) an outer product variant using Level 2 BLAS from LAPACK, (2) the Level 3 BLAS variant from LAPACK described in §3, (3) the inner product variant using Level 2 BLAS described in §3, and (4) the Level 3 BLAS algorithm described in §4. The codes were run in double precision using eight processors on an Alliant FX/8 and compiled[6] using the Alliant Fortran compiler with optimization flags "-O -DAS -alt." Matrices of size 512 and 2,048 were used with 10 and 20 negative eigenvalues, respectively. The block size for the LAPACK Level 3 BLAS code was varied from 32 to 128, and the maximum block

[6] Optimized versions of the BLAS provided by Alliant were used.

TABLE 2
Comparison of performance of different algorithms.

Algorithm	n=512 Block Size	Time (sec)	n=2048 Block Size	Time (sec)
LAPACK-2	n/a	5.57	n/a	367.62
LAPACK-3	4	13.83	4	376.46
LAPACK-3	32	2.46	32	95.82
LAPACK-3	64	1.99	64	89.49
LAPACK-3	128	1.87	128	91.17
New Method	4(3.6)	2.41	4(3.8)	139.56
New Method	32(18)	2.45	32(21)	126.97
New Method	64(31)	2.37	64(41)	99.89
New Method	128(31)	2.32	128(84)	95.78
Inner Product	n/a	2.34	n/a	122.95

TABLE 3
Comparison of accuracy of different algorithms.

Algorithm	n=512 Block Size	Norm of the Residual	n=2048 Block Size	Norm of the Residual
LAPACK-2	n/a	8.2E−11	n/a	4.9E−9
LAPACK-3	4	1.4E−10	4	9.5E−9
LAPACK-3	32	8.2E−11	32	3.4E−9
LAPACK-3	64	1.8E−10	64	2.9E−9
LAPACK-3	128	9.6E−11	128	3.3E−9
New Method	4(3.6)	4.6E−10	4(3.8)	3.9E−9
New Method	32(18)	1.3E−9	32(21)	5.6E−9
New Method	64(31)	7.8E−10	64(41)	2.1E−8
New Method	128(31)	9.0E−10	128(84)	7.2E−9
Inner Product	n/a	3.9E−11	n/a	8.8E−10

size for the new algorithm was varied over the same range. The performance results are given in Table 2. In the "Block Size" column the numbers in parentheses for the new method are the average block size used. From the results, we can see that for realistic block sizes: (1) the LAPACK block algorithm has a slight edge over the new block algorithm, (2) the inner-product algorithm is slightly slower than the block algorithms, and (3) the Level 2 BLAS LAPACK algorithm is much worse than the other algorithms. Another interesting aspect of these results is that for very small block sizes, the LAPACK block algorithm performs very poorly. However, one should note that the LAPACK block algorithm will default to the unblocked algorithm if the unblocked algorithm will be faster. In Table 3, we give the norms of the residuals for each of the experiments in Table 2. We see that, as we might expect, the new block algorithm has a slightly larger residual than the other methods because of its somewhat relaxed stability test. At a sacrifice of some speed[7] the accuracy can be made comparable to that of the other algorithms by requiring the growth factor at each individual pivot to be less than $(1 + \frac{1}{\alpha})$ rather than requiring the growth factor of p pivots to be bounded by $(1 + \frac{1}{\alpha})^p$.

6. Summary. A block algorithm, based on the Bunch–Kaufman algorithm, suitable for factoring symmetric indefinite matrices on high-performance architectures was given. The algorithm, unlike other high-performance variants of Bunch–Kaufman factorization, limits the number of row/column interchanges to the number of negative

[7] In experiments on the Alliant, we found increases in execution time of approximately 10%.

eigenvalues in the matrix. Therefore, this algorithm does not destroy, to a large extent, the structure of a matrix and can be used for factoring banded or sparse matrices. The block algorithm was shown to be faster than its nonblocked counterpart for factoring banded matrices on a multivector computer. It was also shown to be only slightly more expensive than the block algorithm used in LAPACK for factoring indefinite dense matrices.

Acknowledgments. We thank Linda Kaufman for her many helpful comments and for the use of her factorization code. We thank Chris Bischof for comments that led to improvements of the paper. Finally, we thank the staff of the North Carolina Supercomputing Center for providing computer time on the CRAY Y-MP.

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DUCROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[2] C. ASHCRAFT, *A vector implementation of the multifrontal method for large sparse, symmetric positive definite linear systems*, Tech. Report ETA-TR-51, Boeing Computer Services, Seattle, WA, 1987.

[3] V. BARWELL AND A. GEORGE, *A comparison of algorithms for solving symmetric indefinite systems of linear equations*, ACM Trans. Math. Software, 2 (1976), pp. 242–251.

[4] J. R. BUNCH AND L. KAUFMAN, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comput., 31 (1977), pp. 163–179.

[5] J. DONGARRA, J. DUCROZ, I. DUFF, AND S. HAMMARLING, *A set of level 3 basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[6] J. DONGARRA, J. DUCROZ, S. HAMMARLING, AND R. HANSON, *An extended set of fortran basic linear algebra subprograms*, ACM Trans. Math. Software, 14 (1988), pp. 1–32.

[7] I. S. DUFF, J. K. REID, N. MUNKSGAARD, AND H. B. NIELSEN, *Direct solution of sets of linear equations whose matrix is sparse, symmetric and indefinite*, J. Inst. Math. Appl., 23 (1979), pp. 235–250.

[8] K. A. GALLIVAN, R. J. PLEMMONS, AND A. H. SAMEH, *Parallel algorithms for dense linear algebra computations*, SIAM Rev., 32 (1990), pp. 54–135.

[9] M. T. JONES AND M. L. PATRICK, *The Use of Lanczos's Method to Solve the Large Generalized Symmetric Definite Eigenvalue Problem*, Tech. Report 89-67, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA, 1989.

[10] ———, *Factoring Symmetric Indefinite Matrices on High-Performance Architectures*, Tech. Report 90-8, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA, 1990.

[11] ———, *Bunch–Kaufman factorization for real symmetric indefinite banded matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 553–559.

[12] M. T. JONES, M. L. PATRICK, AND R. G. VOIGT, *Language comparison for scientific computing on mimd architectures*, Tech. Report No. 89-6, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA, 1989.

[13] H. JORDAN, *The Force*, Computer Systems Design Group, University of Colorado, Boulder, CO, 1987.

[14] L. KAUFMAN, *Computing the MDM^T Decomposition*, Numerical Analysis Manuscript 11274-911011-20, Bell Labs, Murray Hill, NJ, 1991.

[15] C. LAWSON, R. HANSON, D. KINCAID, AND F. KROGH, *Basic linear algebra subprograms for fortran usage*, ACM Trans. Math. Software, 5 (1979), pp. 308–325.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 284–298, January 1994

022

COMPUTATION OF STABLE INVARIANT SUBSPACES OF HAMILTONIAN MATRICES*

R. V. PATEL†, Z. LIN†, AND P. MISRA‡

Abstract. This paper addresses some numerical issues that arise in computing a basis for the stable invariant subspace of a Hamiltonian matrix. Such a basis is required in solving the algebraic Riccati equation using the well-known method due to Laub. Two algorithms based on certain properties of Hamiltonian matrices are proposed as viable alternatives to the conventional approach.

Key words. Hamiltonian matrices, eigenvalues, invariant subspaces, algebraic Riccati equation

AMS subject classifications. 65F15, 65H10, 93D15

1. Introduction.

A matrix $Z \in \mathcal{R}^{2n\times 2n}$ is called a Hamiltonian matrix if

$$JZJ^T = -Z^T, \tag{1.1}$$

where

$$J = \begin{bmatrix} O_n & I_n \\ -I_n & O_n \end{bmatrix}. \tag{1.2}$$

In (1.2), I_n denotes the $n\times n$ identity matrix and O_n denotes the $n\times n$ null matrix. In this paper we consider some numerical issues related to the problem of finding a basis for the invariant subspace of a Hamiltonian matrix corresponding to a certain subset of its eigenvalues. Such a problem arises in one of the more commonly used methods [1] for solving the well-known continuous-time algebraic Riccati equation (ARE):

$$\mathbf{CR}(X) \equiv A^TX + XA - XGX + F = O, \tag{1.3}$$

where $A \in \mathcal{R}^{n\times n}$; $G = BB^T \in \mathcal{R}^{n\times n}$ and $F = C^TC \in \mathcal{R}^{n\times n}$ are positive semidefinite matrices. If (A, B) is a stabilizable pair and (A, C) is a detectable pair [2], then (1.3) has a unique positive semidefinite solution $X = X^T \in \mathcal{R}^{n\times n}$, which is a stabilizing solution, i.e., $A - GX$ has all its eigenvalues in the open left half-plane. Numerical solutions of equations of the form (1.3) are required in several control system applications, e.g., the linear quadratic optimal control problem, Kalman filtering, and H^∞ robust control. It is not surprising, therefore, that many methods have been proposed for solving the ARE, e.g., see [3], [4] for an extensive bibliography. In this paper, we

*Received by the editors September 18, 1989; accepted for publication (in revised form) May 7, 1992. This work was supported by the Natural Sciences and Engineering Research Council of Canada grant OGP0001345 and by Wright Research and Development Center, Wright Patterson Air Force Base grant F33615-88-C-3605.

†Department of Electrical and Computer Engineering, Concordia University, Montréal, Québec, Canada H3G 1M8 (rajni@davinci.concordia.ca). Current address for Z. Lin, International Submarine Engineering Ltd., 1734 Broadway Street, Port Coquitlam, British Columbia, Canada V3C 2M8 (ise@cs.sfu.ca).

‡Department of Electrical Engineering, Wright State University, Dayton, Ohio 45435 (pmisra@valhalla.cs.wright.edu).

shall concentrate on one of the more reliable and elegant methods for solving (1.3). The Hamiltonian matrix corresponding to (1.3) is given by

$$Z = \begin{bmatrix} A & G \\ F & -A^T \end{bmatrix}.$$

The method proposed by Laub [1] is based on the following results concerning the matrix Z:

(a) If $\lambda \in C$ is an eigenvalue of Z, then so is $-\lambda$.

(b) A symmetric matrix X is the desired stabilizing solution of (1.3) if and only if $X = -U_{21}U_{11}^{-1}$, where the columns of $[U_{11}^T \quad U_{21}^T]^T$ span the n-dimensional invariant subspace of Z corresponding to its stable eigenvalues.

The algorithm proposed by Laub for computing X involves the following steps:

Algorithm I

Step 1. Reduce Z to a real Schur form[1] (RSF), $\hat{R} \in \mathcal{R}^{2n\times 2n}$. Accumulate the orthogonal transformations in a matrix $\hat{U} \in \mathcal{R}^{2n\times 2n}$, i.e.,

$$\hat{R} = \hat{U}^T Z \hat{U}. \tag{1.4}$$

Comment: This step can be performed by first reducing Z to upper Hessenberg form and then using the QR Algorithm [5]. The reduction to upper Hessenberg form requires approximately $\frac{5}{3}(2n)^3$ flops (floating point operations) and the reduction of the resulting upper Hessenberg matrix to an RSF requires approximately $4\kappa(2n)^3$ flops, where κ represents the average number of QR steps required per eigenvalue (≈ 1.5).

Step 2. Rearrange the eigenvalues of $\hat{R}$ so that the n stable eigenvalues are in the top left corner.

Comment: This can be achieved by means of orthogonal transformations on $\hat{R}$ using the subroutines EXCHNG and HQR3 [6] (also note the corrections in [7]) and requires more than $4\kappa(2n)^3$ flops. Let this eigenvalue rearrangement operation be represented by

$$\tilde{R} = \tilde{U}^T \hat{R} \tilde{U}, \tag{1.5}$$

where the orthogonal transformations resulting from using EXCHNG and HQR3 are accumulated in $\tilde{U}$, and the submatrix $\tilde{R}_{11} \in \mathcal{R}^{n\times n}$ is in RSF with all its eigenvalues in the open left half-plane.

Step 3. Set $U = \hat{U}\tilde{U}$ and let

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, \text{ where } U_{11}, U_{21} \in \mathcal{R}^{n\times n}.$$

Step 4. Solve $XU_{11} = -U_{21}$ for X.

Comment: This step can be performed using the appropriate subroutines from LINPACK and requires approximately $\frac{4}{3}n^3$ flops.

The above approach (also called the Schur vector approach) can be regarded as a generalization, and a numerically much more robust implementation, of the eigenvector approach of MacFarlane [8]. In the rest of this paper, we concentrate on Steps 1

[1] By a real Schur form, we mean an upper quasi-triangular real matrix with a scalar along the diagonal for each real eigenvalue and a 2×2 block for each pair of complex-conjugate eigenvalues.

and 2, and more specifically on the problem of numerically computing the columns of $[U_{11}^T \quad U_{21}^T]^T$ that span the n-dimensional invariant subspace of Z corresponding to its stable eigenvalues. Our intention is to suggest some ways of improving the efficiency and accuracy of the computations that are required in these steps.

2. Computing the eigenvalues of a Hamiltonian matrix. The method proposed by Laub uses an algorithm for reduction of a general matrix to upper Hessenberg form and the QR Algorithm to find the eigenvalues of Hessenberg matrices. Therefore, it does not take into account the special structure of Z, so that the transformations employed in this algorithm destroy the Hamiltonian structure of Z. On the other hand, if similarity transformations on Z are carried out using symplectic matrices, then its Hamiltonian structure will be preserved. In other words, the matrix $Z_0 = V^{-1}ZV$ will be a Hamiltonian matrix if V is a symplectic matrix. A matrix $V \in \mathcal{R}^{2n\times 2n}$ is symplectic if $V^TJV = J$, where J is defined by (1.2). From the point of view of numerical reduction of Z to a condensed form, such as a block upper triangular form, it is desirable to perform the required transformations using orthogonal symplectic matrices. The following result shows the existence of one such condensed form.

THEOREM 2.1. *If Z has no eigenvalues on the imaginary axis, then there exists an orthogonal symplectic matrix*

$$V = \begin{bmatrix} V_{11} & V_{21} \\ -V_{21} & V_{11} \end{bmatrix} \tag{2.1}$$

with V_{11}, $V_{12} \in \mathcal{R}^{n\times n}$, such that

$$V^TZV = \begin{bmatrix} R & \hat{G} \\ O_n & -R^T \end{bmatrix} \equiv \hat{Z}, \tag{2.2}$$

where $\hat{G} = \hat{G}^T \in \mathcal{R}^{n\times n}$, and $R \in \mathcal{R}^{n\times n}$ is in RSF with eigenvalues in the open left half-plane.

Proof. See [9].

There are two types of orthogonal symplectic matrices that are particularly useful in performing reductions on Hamiltonian matrices. The first type consists of Householder symplectic matrices defined by

$$P(k, \mathbf{u}) = \begin{bmatrix} \hat{P} & O_n \\ O_n & \hat{P} \end{bmatrix}, \quad \hat{P} \in \mathcal{R}^{n\times n}, \tag{2.3}$$

where

$$\hat{P} = I_n - \frac{2\mathbf{u}\mathbf{u}^T}{\mathbf{u}^T\mathbf{u}} \tag{2.4a}$$

and

$$\mathbf{u}^T = [0, \ldots, 0, u_k, \ldots, u_n] \neq \mathbf{0}^T. \tag{2.4b}$$

The second type consists of Givens symplectic matrices defined by

$$J(k, c, s) = \begin{bmatrix} C & S \\ -S & C \end{bmatrix}, \quad C, S \in \mathcal{R}^{n\times n}, \tag{2.5}$$

where

$$C = \text{diag}\{\underbrace{1,\ldots,1}_{k-1}, c, 1, \ldots, 1\},$$
$$S = \text{diag}\{\underbrace{0,\ldots,0}_{k-1}, s, 0, \ldots, 0\}, \tag{2.6}$$

and $c^2 + s^2 = 1$. Algorithms are given in [12] for computing $P(k, \mathbf{u})$ and $J(k, c, s)$ to zero specific entries in a vector.

Theorem 2.1 and its proof in [9] show that it is possible to reduce Z using the structure-preserving orthogonal symplectic transformations to the block upper triangular form (2.2), but no algorithm for doing so is provided. In fact, so far success in developing an *efficient* QR-type algorithm for this reduction has been reported only for a special case [10], [17], namely, when $\text{rank}(G) = 1$ or $\text{rank}(F) = 1$. In this case, Byers has provided an extension of the implicitly shifted QR algorithm that uses orthogonal symplectic transformations. However, because the method is only applicable for a special case, we shall not consider it further in this paper. It suffices to mention that in the algorithm proposed by Byers, a reordering of the eigenvalues is required to ensure that R is a stable matrix. This reordering is performed using EXCHNG and HQR3 to bring an unstable eigenvalue of R to its (n, n)th position followed by a Givens symplectic transformation, $J(n, c, s)$, to interchange the (n, n)th entry of R with that of $-R^T$. Also, it is worth mentioning that reduction of Z to a block triangular condensed form has been achieved in the general case using nonorthogonal symplectic similarity transformations [11]. However, the use of such transformations may cause numerical instability.

An elegant method that uses orthogonal symplectic matrices to "approximate" the eigenvalues of a Hamiltonian matrix has been proposed by Van Loan [12]. The algorithm given in [12] computes the eigenvalues of Z^2, i.e., of

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \equiv \begin{bmatrix} A & G \\ F & -A^T \end{bmatrix}^2, \tag{2.7}$$

where

$$M_{11} = A^2 + GF = M_{22}^T, \tag{2.8a}$$

$$M_{12} = AG - GA^T = -M_{12}^T, \tag{2.8b}$$

and

$$M_{21} = FA - A^T F = -M_{21}^T. \tag{2.8c}$$

Note that M_{12} and M_{21} are skew-symmetric matrices. Also, it can be easily shown that the structure of M is preserved under symplectic similarity transformations [12]. Furthermore, if Z has eigenvalues $\{\lambda_1, -\lambda_1, \ldots, \lambda_n, -\lambda_n\}$, then the eigenvalues of M are $\{\lambda_1^2, \lambda_1^2, \ldots, \lambda_n^2, \lambda_n^2\}$. The eigenvalues of Z can, therefore, be easily obtained from those of M. However, it should be noted that the algorithm in [12] does not give us an RSF of Z, nor does it enable us to compute the stable invariant subspace of Z. The eigenvalues of M can be computed using the following steps:

Algorithm II

Step 1. Form $M = \begin{bmatrix} A & G \\ F & -A^T \end{bmatrix}^2$

Step 2. Compute an orthogonal symplectic matrix Q such that

$$Q^T M Q = \begin{bmatrix} H & N \\ O_n & -H^T \end{bmatrix}$$

where H is in upper Hessenberg form (see Algorithm SR in [12]).

Step 3. Compute the eigenvalues of H $(\mu_i, i = 1, \ldots, n)$ using the QR Algorithm.

Step 4. Compute $\lambda_i = \sqrt{\mu_i}, i = 1, \ldots, n$. Set $\lambda_{n+i} = -\lambda_i, i = 1, \ldots, n$.

The symplectic orthogonal matrix Q in Step 2 is a product of symplectic Householder and Givens transformation matrices and is structure-preserving. The complete algorithm for the reduction is given in [12] with implementation details and numerical properties. It suffices to mention here that the algorithm requires approximately $53n^3/3$ flops which is about 25–30% of the computation required by the QR Algorithm applied to Z. Furthermore, if floating point arithmetic with base b and precision t is used, then it can be shown that the computed eigenvalues of Z obtained using Algorithm II are the exact eigenvalues of a matrix $Z + E_s$, where $E_s \in \mathcal{R}^{2n \times 2n}$ satisfies

$$\|E_s\|_2 \approx \sqrt{b^{-t}} \|Z\|_2. \tag{2.9}$$

As a comparison, if the eigenvalues of Z are computed using the QR Algorithm (as is done in [1]), then the computed eigenvalues are the exact eigenvalues of a matrix $Z + E_Q$, where $E_Q \in \mathcal{R}^{2n \times 2n}$ satisfies

$$\|E_Q\|_2 \approx b^{-t} \|Z\|_2. \tag{2.10}$$

This implies that the error in computing the eigenvalues of Z using Algorithm II may be up to $\sqrt{b^t}$ times as large as that using the QR Algorithm. Also, in general, Algorithm II gives better accuracy for eigenvalues with larger magnitudes than those with smaller magnitudes.

The fact that Algorithm II computes the eigenvalues of Z less accurately than the QR Algorithm is not a matter of concern in our application. It is sufficient at this point to know approximately the set of n stable or unstable eigenvalues of Z. More accurate values of these will be obtained and at the same time reordered to get a stabilizing solution of the ARE.

3. Condensed forms for Z with specified eigenvalue ordering. We now consider the problem of reducing Z to condensed forms in which the eigenvalues of Z are separated into sets of stable and unstable eigenvalues, i.e., Z is reduced to a block triangular form

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ O_n & Z_{22} \end{bmatrix}, \tag{3.1}$$

where Z_{11} and $Z_{22} \in \mathcal{R}^{n \times n}$ and have only stable and unstable eigenvalues, respectively. In this section, we show how two such condensed forms can be computed. Our approach uses a modification of the QR Algorithm. The algorithms described in this section can be regarded as alternatives to using the EXCHNG and HQR3 subroutines [6]. In this context, it is worth mentioning that since no interchanging of eigenvalues is done in our approach, we avoid the numerical difficulties that may be encountered in attempting to exchange nearly equal eigenvalues using EXCHNG and HQR3 [13].

We start by showing how a specified eigenvalue of Z can be made to appear in the bottom right (Algorithm QR-down) or top left (Algorithm QR-up) position. For the sake of brevity, we shall present only Algorithm QR-down in detail. Algorithm QR-up can be stated in an analogous manner. We assume that Z is in unreduced upper Hessenberg form and the eigenvalue to be positioned is given.

Algorithm QR-down(Z, n, λ, D)

Step 1. If the eigenvalue to be shifted (λ) is complex, go to Step 4.

Step 2. Form a real shift corresponding to λ: $Z := Z - \lambda I_{2n}$

Step 3. For $k = 1, 2, \ldots, 2n-1$, determine a Householder matrix $\hat{D}_k \in \mathcal{R}^{2\times 2}$ such that

$$\hat{D}_k \begin{bmatrix} z_{kk} \\ z_{k+1,k} \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix} \tag{3.2}$$

$Z := D_k Z D_k^T$, where $D_k = \operatorname{diag}\{I_{k-1}, \hat{D}_k, I_{2n-k-1}\}$

$D := D_k D$

end

$Z := Z + \lambda I_{2n}$

exit

Step 4. Form an implicit double shift corresponding to the complex-conjugate pair (λ, λ^*):

$p_1 := z_{11}^2 - z_{11}(\lambda + \lambda^*) + \lambda\lambda^* + z_{12}z_{21}$

$q_1 := z_{21}[z_{11} + z_{22} - (\lambda + \lambda^*)]$

$r_1 := z_{21}z_{32}$

Determine a Householder matrix $\hat{D}_0 \in \mathcal{R}^{3\times 3}$ such that

$$\hat{D}_0 \begin{bmatrix} p_1 \\ q_1 \\ r_1 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \\ 0 \end{bmatrix}$$

$Z := D_0 Z D_0^T$, where $D_0 = \operatorname{diag}\{\hat{D}_0, I_{2n-3}\}$

$D := D_0 D$

Step 5. For $k = 1, 2, \ldots, 2n-3$, determine a Householder matrix $\hat{D}_k \in \mathcal{R}^{3\times 3}$ such that

$$\hat{D}_k \begin{bmatrix} z_{k+1,k} \\ z_{k+2,k} \\ z_{k+3,k} \end{bmatrix} = \begin{bmatrix} \times \\ 0 \\ 0 \end{bmatrix} \tag{3.3}$$

$Z := D_k Z D_k^T$, where $D_k = \operatorname{diag}\{I_k, \hat{D}_k, I_{2n-k-3}\}$

$D := D_k D$

end

Determine a Householder matrix $\hat{D}_{2n-2} \in \mathcal{R}^{2\times 2}$ such that

$$\hat{D}_{2n-2} \begin{bmatrix} z_{2n-1,2n-2} \\ z_{2n,2n-2} \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$$

$Z := D_{2n-2} Z D_{2n-2}^T$, where $D_{2n-2} = \operatorname{diag}\{I_{2n-2}, \hat{D}_{2n-2}\}$

$D := D_{2n-2} D$

exit

Remark 3.1. Algorithm QR-up (Z, n, λ, U) can be defined in a similar manner to the way in which QR-down was defined. In this case, the eigenvalue(s) corresponding to the shift are made to appear in the top left corner of Z.

Remark 3.2. Algorithms QR-down and QR-up perform one step (single for a real eigenvalue and double for a complex-conjugate pair of eigenvalues) of the QR Algorithm [5]. Explicit single shifts are used for real eigenvalues and implicit double shifts for complex-conjugate pairs of eigenvalues. The implicit shifts introduce nonzero terms in certain locations below the subdiagonal of the upper Hessenberg matrix. Algorithm QR-down uses row operations to "chase" these nonzero terms to the bottom right corner, whereas Algorithm QR-up uses column operations to "chase" the terms to the top left.

Remark 3.3. If a shift is an accurately computed eigenvalue of Z, then Algorithm QR-down will transform the matrix Z to the form

$$\begin{bmatrix} \hat{Z}_{11} & \hat{\mathbf{z}}_{12}^T \\ \mathbf{0} & \lambda \end{bmatrix}$$

for a real single shift λ, or to the form

$$\begin{bmatrix} \tilde{Z}_{11} & \tilde{Z}_{12} \\ O & \Phi \end{bmatrix},$$

where $\Phi \in \mathcal{R}^{2\times 2}$ for a double shift. Similarly, Algorithm QR-up will position λ and Φ in the top left corner of a quasi-triangular matrix. Now, if a shift is not equal to an accurately computed eigenvalue of Z, then the subdiagonal element(s) next to λ (Φ) (or below λ (Φ) in the case of QR-up) may not become zero after one iteration, in which case two or more iterations of the algorithm may be required. In this case the shifts for the second and subsequent iterations would correspond to the scalar (for a real eigenvalue) or the 2×2 matrix (for a complex-conjugate pair of eigenvalues) in the bottom right corner (Algorithm QR-down) or top left corner (Algorithm QR-up). The effect of performing these additional iterations would be to reduce the appropriate subdiagonal term(s) to zero and yield more accurate value(s) for the eigenvalue(s).

Remark 3.4. The orthogonal similarity transformations on Z are accumulated in $D \in \mathcal{R}^{2n\times 2n}$ for Algorithm QR-down and in $U \in \mathcal{R}^{2n\times 2n}$ for Algorithm QR-up.

Remark 3.5. It has been assumed in algorithms QR-down and QR-up that the upper Hessenberg matrix Z is unreduced. If this is not the case, then the sequence of transformations D_k (in QR-down) and U_k (in QR-up) cannot be completed. However, this is not a limitation because, if Z is not unreduced, then it can be made unreduced by applying an arbitrary QR shift to introduce coupling between the corresponding blocks [6].

3.1. A real Schur form with eigenvalue rearrangement. In this section we show how the algorithms QR-down and QR-up can be used to obtain the condensed form (3.1), in which $Z_{11} \in \mathcal{R}^{n\times n}$ and $Z_{22} \in \mathcal{R}^{n\times n}$ are in RSF and have only stable and unstable eigenvalues, respectively.

Algorithm III (RSF)

Input: A Hamiltonian matrix $Z \in \mathcal{R}^{2n\times 2n}$

Output: An orthogonal matrix $\hat{U} \in \mathcal{R}^{2n\times 2n}$ such that

$$\hat{U}^T Z \hat{U} = \begin{bmatrix} R_1 & \hat{Z}_{12} \\ O & R_2 \end{bmatrix} \tag{3.4}$$

where $R_1 \in \mathcal{R}^{n\times n}$ and $R_2 \in \mathcal{R}^{n\times n}$ are in RSF with stable and unstable eigenvalues, respectively.

Step 1. Reduce Z to an upper Hessenberg form Z_1:

$Z_1 := \hat{U}_1^T Z \hat{U}_1$
$n_1 := n$
$\rho := 0$

Step 2. For $k = 1, 2, \ldots, n$, compute an eigenvalue λ_k of Z_1 (using one or more iterations of Algorithm QR-down); accumulate the transformations in $\hat{U}_2$.

If $\mathrm{Re}(\lambda_k) > 0$,
  Call QR-up $(Z_k, n_k, -\lambda_k, U_k)$
else
  Call QR-down $(Z_k, n_k, \lambda_k, U_k^T)$
  Call QR-up $(Z_k, n_k, -\lambda_k, U_k)$
end
$\hat{U}_2 := \hat{U}_2 \ \mathrm{diag}\{I_\rho, U_k, I_\rho\}$
If λ_k is real,
  $Z_{k+1} := Z_k(2 : 2n_k - 1, 2 : 2n_k - 1)$
  $n_{k+1} := n_k - 1$
  $\rho := \rho + 1$
else
  $Z_{k+1} := Z_k(3 : 2n_k - 2, 3 : 2n_k - 2)$
  $n_{k+1} := n_k - 2$
  $\rho := \rho + 2$
end
end

Step 3. $\hat{U} := \hat{U}_1 \hat{U}_2$

Remark 3.6. Algorithm III computes only n eigenvalues of Z. After an eigenvalue λ_k has been determined using Algorithm QR-down, if λ_k is an unstable eigenvalue, then a shift $-\lambda_k$ is applied using Algorithm QR-up to position the stable eigenvalue $-\lambda_k$ in the top left corner. On the other hand, if λ_k is a stable eigenvalue, then a shift $-\lambda_k$ using Algorithm QR-down and another shift λ_k using Algorithm QR-up are applied to position the unstable eigenvalue $(-\lambda_k)$ in the bottom right corner and the stable eigenvalue (λ_k) in the top left corner, respectively. As an illustration, let us consider the case $k = 2$, $n = 4$, with computed real eigenvalues $\lambda_1 > 0$ and $\lambda_2 > 0$. Then the structure of the resulting matrix would be

$$\begin{bmatrix} -\lambda_1 & \times & \times & \times & \times & \times & \times & \times \\ 0 & -\lambda_2 & \times & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 \end{bmatrix}.$$

Thus, as k increases by 1, the eigenvalue problem deflates by 2. Therefore, since we compute only n eigenvalues of Z, the amount of computation required is approximately $\frac{1}{2}(4\kappa)(2n)^3$ flops, where κ represents the average number of QR steps required per eigenvalue and is usually overestimated by a factor of 1.5. Also, once λ_k has been determined, a shift of $-\lambda_k$ normally yields the eigenvalue $-\lambda_k$ in the appropriate location in one sweep of the QR Algorithm ($\kappa = 1$). Therefore, the remaining n eigenvalues require approximately $16n^3$ operations. The complete reduction of the upper Hessenberg matrix Z_1 to an RSF can thus be achieved using approximately $40n^3$

flops. For comparison, we note that the reduction of a $2n \times 2n$ upper Hessenberg matrix to RSF using the QR Algorithm requires approximately $4\kappa(2n)^3$ operations. Using $\kappa = 1.5$, this amounts to approximately $48n^3$ operations. If the ordering of the eigenvalues in the RSF requires about 25% additional operations, then we have approximately $60n^3$ operations for computing all $2n$ eigenvalues of an upper Hessenberg matrix *and* rearranging them in groups of stable and unstable eigenvalues using EXCHNG and HQR3.

Remark 3.7. Further speedup in Algorithm III may be achieved by *first* using Algorithm II to compute the eigenvalues of Z. The approach would then be reduced to performing shifts corresponding to these eigenvalues to position them in the appropriate blocks on the diagonal. However, as we shall see in the next section, we need only apply shifts corresponding to the unstable eigenvalues, thereby further reducing the computational effort required.

3.2. A Hessenberg–Schur form. We now describe the reduction of Z to a block upper triangular form (3.1) in which $Z_{11} \in \mathcal{R}^{n\times n}$ is in upper Hessenberg form and $Z_{22} \in \mathcal{R}^{n\times n}$ is in real Schur form. Furthermore, Z_{11} will have all its eigenvalues in the open left half of the complex plane. The corresponding columns of the accumulated transformation matrix will then immediately give us a basis for the stable invariant subspace of Z. The stabilizing solution of the ARE can then be obtained as mentioned in §1.

Algorithm IV (Hessenberg–Schur Form)

Input: A Hamiltonian matrix $Z \in \mathcal{R}^{2n\times 2n}$

Output: An orthogonal matrix $U \in \mathcal{R}^{2n\times 2n}$ such that

$$U^T Z U = \begin{bmatrix} H & Z_{12} \\ O_n & R \end{bmatrix} \tag{3.5}$$

where $H \in \mathcal{R}^{n\times n}$ is an upper Hessenberg matrix with only stable eigenvalues and $R \in \mathcal{R}^{n\times n}$ is in an RSF with only unstable eigenvalues.

Step 1. Compute the "approximate" *unstable* eigenvalues, $\tilde{\lambda}_i$, $i = 1, \ldots, n$ of Z using Algorithm II.

Step 2. Reduce Z to upper Hessenberg form : $Z_1 := U_1^T Z U_1$, and $n_1 := n$, $\rho := 0$, $U_2 := I_{2n}$.

Step 3. For $k = 1, 2, \ldots, n$,

Call QR-down $(Z_k, n_k, \tilde{\lambda}_k, \tilde{U}_k^T)$

$U_2 := U_2 \operatorname{diag}\{\tilde{U}_k,\ I_\rho\}$

If λ_k is real

$Z_{k+1} := Z_k(1 : 2n_k - 1, 1 : 2n_k - 1)$

$n_{k+1} := n_k - 1$

$\rho := \rho + 1$

else

$Z_{k+1} := Z_k(1 : 2n_k - 2, 1 : 2n_k - 2)$

$n_{k+1} := n_k - 2$

$\rho := \rho + 2$

end

end

Step 4. $U := U_1 U_2$

Remark 3.8. By partitioning U as

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix},$$

where U_{11}, $U_{21} \in \mathcal{R}^{n\times n}$, it is easy to see that $[U_{11}^T \quad U_{21}^T]^T$ is a basis for the invariant subspace corresponding to the stable eigenvalues of Z. Therefore, a stabilizing solution of the ARE is $X = -U_{21}U_{11}^{-1}$.

Remark 3.9. It should be noted that only n shifts are applied to Z_1 using the QR Algorithm and these shifts correspond to the unstable eigenvalues of Z. If the transformations that are accumulated in U_1 and U_2 are performed on Z (via Algorithm QR-down), then the result will be a real Schur matrix R with the unstable eigenvalues of Z in the bottom right corner, leaving an upper Hessenberg matrix H with the stable eigenvalues of Z in the top left corner. However, if we are only interested in the solution X of the ARE, we do not need to store the matrix resulting from the operations on Z, since X can be computed from the first n columns of U.

Remark 3.10. If the eigenvalues $\tilde{\lambda}_i$, $i = 1, \ldots, n$, computed using Algorithm II are accurate, then the process of reducing Z to the Hessenberg–Schur form becomes finite because the required shifts are known a priori. In this case Algorithm IV will be slightly faster than Algorithm III. On the other hand, if the eigenvalues $\tilde{\lambda}_i$, $i = 1, \ldots, n$ are not computed accurately, then it may be necessary to perform two or more iterations of Algorithm QR-down with updated values of $\tilde{\lambda}_i$ to provide more accurate shifts. This will result in finer tuning of the shifts yielding more accurate values for the unstable eigenvalues, which in turn will improve the accuracy of H and U.

Remark 3.11. As mentioned in §2, computation of the eigenvalues of Z using Van Loan's method [12] requires approximately $53n^3/3$ operations. The ordering of n unstable eigenvalues to get the Hessenberg–Schur structure involves approximately $4\kappa(2n)^3/2$ operations. We can use $\kappa = 1$, since the shifts employed are the eigenvalues computed using Van Loan's method and, in general, one iteration per eigenvalue should suffice to "fine tune" the eigenvalues to the accuracy of the QR Algorithm. This results in an operations count of approximately $33.7n^3$ flops for the reduction of the upper Hessenberg matrix Z_1 to the Hessenberg–Schur form (3.5).

Remark 3.12. It is worth mentioning here that in a recent report [14], an algorithm has been proposed that uses the "approximate" eigenvalues computed using Van Loan's method to implement block shifts (of n eigenvalues at a time) on a condensed form of Z obtained by Paige and Van Loan [9] using orthogonal symplectic transformations. The algorithm in [14], therefore, uses only orthogonal symplectic transformations to obtain the RSF in (2.2); but no analysis or numerical experiments are given to show the efficiency or accuracy of the algorithm. As mentioned in [9], performing orthogonal symplectic updates of the condensed form destroys its nice zero structure. Consequently, carrying out implicit double shifts in this case becomes computationally expensive, so that no advantage is gained even though only orthogonal symplectic transformations are used. The use of implicit n simultaneous (block) shifts overcomes this problem, especially if the shifts are the "approximate" eigenvalues of Z. However, the buildup of rounding errors in implementing a large number of shifts simultaneously can cause difficulties with convergence of this symplectic block QR-type algorithm, particularly if one or more of the eigenvalues are badly conditioned.

4. Numerical examples. In this section, we illustrate the numerical performance and properties of the algorithms proposed in the preceding sections. All computations were performed on a SUN 4/370 computer using the f77 compiler.

Example 1. In this example, we use the model for the high-speed vehicle control problem described in [1]. State matrices of orders 9, 19, 29, 39, and 49 were considered and, with appropriate choice of matrices G and F, Hamiltonian matrices of orders 18, 38, 58, 78, and 98, respectively, were obtained. The eigenvalue problems for these

matrices are quite well conditioned. For each Hamiltonian matrix, the comparisons were performed for the following computations:

(a) Computing the eigenvalues and separating them into groups of stable and unstable eigenvalues.

(b) Computing the positive semidefinite solution of the corresponding ARE from the stable invariant subspace.

The above computations were done for three cases:

1. The QR Algorithm and the EXCHNG subroutine [6] were used to compute the eigenvalues and order them into groups of stable and unstable eigenvalues. The computations were performed using both single and double precision arithmetic.
2. Algorithm III was used in single precision to obtain the RSF in (3.4).
3. Algorithm IV was used in single precision to reduce the Hamiltonian matrix to the Hessenberg–Schur form in (3.5).

Note that case 3 requires the use of Van Loan's method [12] (Algorithm II) to "approximate" the unstable eigenvalues of the Hamiltonian matrix. In fact, since the Hamiltonian matrices in this example are well conditioned, the eigenvalues computed using Van Loan's method are almost as accurate as those obtained using the QR Algorithm (case 1). The accuracies of the unstable eigenvalues obtained using single precision arithmetic (as compared with the double precision results obtained in case 1) are shown in Table 4.1 for the three cases and the five Hamiltonian matrices.

TABLE 4.1
Maximum relative error in $\lambda(Z)$.

Size of Z	HQR3-EXCHNG	Algorithm III	Algorithm IV
18×18	9.68×10^{-7}	4.37×10^{-7}	2.76×10^{-7}
38×38	1.68×10^{-6}	6.97×10^{-7}	5.96×10^{-7}
58×58	2.29×10^{-6}	8.29×10^{-7}	1.43×10^{-6}
78×78	2.48×10^{-6}	4.53×10^{-6}	3.59×10^{-6}
98×98	3.14×10^{-6}	6.70×10^{-6}	2.35×10^{-6}

In this table and elsewhere, the "relative error" in the eigenvalues is the value $|\lambda_i^d - \lambda_i^s|/|\lambda_i^d|$, where λ_i^s and λ_i^d denote the ith eigenvalue computed using single and double precision arithmetic, respectively. The maximum is taken over all the eigenvalues of Z. It is clear from Table 4.1 that the accuracy obtained using Algorithms III and IV is comparable to or better than that obtained using the conventional approach (with the HQR3 and EXCHNG subroutines).

For the three cases and the five Hamiltonian matrices, we obtained bases for the stable invariant subspaces from the accumulated transformations and computed the positive semidefinite solutions of the corresponding AREs. All the computations were performed in single precision. For comparison, we also obtained the solution of each ARE in double precision for case 1. We used this to compute the relative errors in the single precision solutions of the AREs in the three cases, i.e., $\|X_d - X_s\|_2/\|X_d\|_2$, where X_d is the double precision solution of the ARE using Algorithm I with EXCHNG and HQR3 (case 1) and X_s is the single precision solution of the ARE computed using Algorithm I (case 1), Algorithm III (case 2), and Algorithm IV (case 3). The relative errors in the solutions are shown in Table 4.2 for the three cases and the five Hamiltonian matrices.

Example 2. In this example, we generated random matrices of orders 9, 19, 29, 39, and 49 for A. Also, the matrices G and F were constructed as $G = BB^T$ and

TABLE 4.2
Relative error in X.

Size of Z	HQR3-EXCHNG	Algorithm III	Algorithm IV
18×18	7.46×10^{-7}	7.29×10^{-7}	6.73×10^{-7}
38×38	1.61×10^{-6}	1.39×10^{-6}	1.37×10^{-6}
58×58	2.22×10^{-6}	2.25×10^{-6}	1.63×10^{-6}
78×78	2.86×10^{-6}	2.31×10^{-6}	2.29×10^{-6}
98×98	3.72×10^{-6}	2.71×10^{-6}	2.89×10^{-6}

$F = C^TC$, where B and C were random matrices. Therefore, once again we have Hamiltonian matrices of orders 18, 38, 58, 78, and 98, respectively. The computations (a) and (b) for the three cases (1–3) mentioned in Example 1 were performed on these matrices. The corresponding results are shown in Tables 4.3 and 4.4.

TABLE 4.3
Maximum relative error in $\lambda(Z)$.

Size of Z	HQR3-EXCHNG	Algorithm III	Algorithm IV
18×18	2.97×10^{-6}	1.05×10^{-6}	6.78×10^{-7}
38×38	3.49×10^{-6}	2.08×10^{-6}	1.74×10^{-6}
58×58	1.31×10^{-5}	1.11×10^{-5}	1.26×10^{-5}
78×78	4.44×10^{-6}	3.43×10^{-6}	2.05×10^{-6}
98×98	1.89×10^{-5}	1.90×10^{-5}	7.75×10^{-6}

In this example, we note again that the accuracy obtained using Algorithms III and IV is comparable to or better than that obtained using the conventional approach (with the HQR3 and EXCHNG subroutines).

TABLE 4.4
Relative error in X.

Size of Z	HQR3-EXCHNG	Algorithm III	Algorithm IV
18×18	4.82×10^{-6}	5.54×10^{-6}	3.26×10^{-6}
38×38	3.23×10^{-6}	3.15×10^{-6}	2.47×10^{-6}
58×58	7.91×10^{-6}	6.84×10^{-6}	7.87×10^{-6}
78×78	5.99×10^{-6}	5.55×10^{-6}	5.24×10^{-6}
98×98	6.79×10^{-6}	6.20×10^{-6}	5.03×10^{-6}

Example 3. This is an example of a Hamiltonian matrix of order 24 with some very ill-conditioned eigenvalues. The example is the same as that used by Van Loan [12], and the matrix A and its eigenvalues are given in [15]. The latter were used in obtaining the relative errors in the computed eigenvalues. The computations in Example 1 (a) were carried out for cases 1–3 in double precision. The results for the four most ill-conditioned eigenvalues are shown in Table 4.5, where the quantity $s(\lambda_i)$ is the cosine of the angle between the left and right eigenvectors associated with the eigenvalue λ_i. The reciprocal of $s(\lambda_i)$ denotes the conditioning of λ_i [15].

In this example, because of the ill conditioning, only double precision arithmetic was used. For the ill-conditioned eigenvalues, Van Loan's method (Algorithm II) was significantly less accurate than the QR Algorithm (case 1). Therefore, two or more QR steps (QR-down) are needed in Algorithm IV to improve the accuracy and to

TABLE 4.5
Relative error.

$\lambda \approx_i$	$s(\lambda_i) \approx$	HQR3-EXCHNG	Algorithm III	Algorithm IV
0.1436	10^{-7}	5.54×10^{-8}	4.61×10^{-8}	6.44×10^{-8}
0.0812	10^{-8}	4.73×10^{-7}	4.13×10^{-7}	5.08×10^{-7}
0.0495	10^{-8}	1.21×10^{-6}	1.08×10^{-6}	1.26×10^{-6}
0.0310	10^{-8}	9.49×10^{-7}	8.49×10^{-7}	9.72×10^{-7}

isolate the unstable eigenvalues in the matrix R in (3.5). Table 4.6 shows the relative errors in the ill-conditioned eigenvalues computed using Van Loan's method and those obtained using Algorithm IV.

TABLE 4.6
Relative error.

$\lambda_i \approx$	Algorithm II	Algorithm IV
0.1436	4.74×10^{-6}	6.44×10^{-8}
0.0812	5.66×10^{-5}	5.08×10^{-7}
0.0495	2.16×10^{-4}	1.26×10^{-6}
0.0310	2.55×10^{-4}	9.72×10^{-7}

Example 4. Here we tested the algorithms for two scenarios: when some of the eigenvalues of the Hamiltonian matrices as well as the corresponding AREs are poorly conditioned, and when there are multiple eigenvalues. The computations (a) and (b) for the three cases (1–3) mentioned in Example 1 were performed for five Hamiltonian matrices. The relative errors in the eigenvalues and in the solutions of the AREs are shown in Tables 4.7 and 4.8, respectively. The three poorly conditioned examples correspond to Hamiltonian matrices, which were generated using the Frank matrix, the data for the boiler model [16], and an example given by Byers [17]. It should be noted that the values of $\min\{s(\lambda_i)\}$ in Tables 4.7 and 4.8 are for the closed-loop eigenvalues, i.e., for the eigenvalues of the Hamiltonian matrices with nonzero F and G matrices, whereas in Table 4.5, the values of $s(\lambda_i)$ are for the open-loop case, i.e., for the Frank matrix. The term κ_R in Table 4.8 denotes the "condition number" of the ARE [17] corresponding to a given Hamiltonian matrix, and is given by

$$\kappa_R = \frac{(1 + \|X\|)^2 \|Z\|}{\|X\| SEP[(A - GX), -(A - GX)^T]},$$

where $SEP[N, -N^T] = \min\{\|PN + N^T P\| \mid \|P\| = 1\}$. A large value of κ_R implies an ill-conditioned ARE.

From Tables 4.7 and 4.8, we note that the eigenvalues and the AREs corresponding to the Hamiltonian matrices with multiple eigenvalues are well conditioned. As expected, for these two cases the three algorithms give very good and comparable accuracy. In the first case, the Hamiltonian matrix has eigenvalues at ± 20, ± 30, ± 40, and ± 50 with multiplicity 3; in the second case, the eigenvalues are at ± 4 with multiplicity 6, and at ± 10, ± 20, and ± 30 each with multiplicity 2.

The three other Hamiltonian matrices considered in Tables 4.7 and 4.8 are relatively poorly conditioned. The boiler problem has the worst conditioning of the three cases, both with respect to the eigenvalues of the Hamiltonian matrix as well as the

TABLE 4.7
Maximum relative error in $\lambda(Z)$.

Examples	$\min\{s(\lambda_i)\} \approx$	HQR3-EXCHNG	Algorithm III	Algorithm IV
Boiler problem (18 × 18)	10^{-8}	1.80×10^{-3}	1.04×10^{-3}	6.40×10^{-3}
Frank matrix (24 × 24)	10^{-4}	8.62×10^{-4}	7.05×10^{-4}	7.16×10^{-4}
Byers's example (10 × 10)	10^{-3}	1.64×10^{-5}	1.52×10^{-5}	3.99×10^{-6}
Multiple case 1 (24 × 24)	10^{-1}	1.24×10^{-6}	9.53×10^{-7}	6.86×10^{-7}
Multiple case 2 (24 × 24)	10^{-1}	1.20×10^{-6}	1.04×10^{-6}	1.40×10^{-6}

TABLE 4.8
Relative error in X.

Examples	$\kappa_R \approx$	HQR3-EXCHNG	Algorithm III	Algorithm IV
Boiler problem (18 × 18)	10^{15}	1.20×10^{-3}	1.20×10^{-3}	5.90×10^{-3}
Frank matrix (24 × 24)	10^{7}	1.10×10^{-3}	9.71×10^{-4}	9.01×10^{-4}
Byers's example (10 × 10)	10^{9}	4.01×10^{-6}	4.14×10^{-6}	4.03×10^{-6}
Multiple case 1 (24 × 24)	20	4.99×10^{-8}	3.99×10^{-8}	3.94×10^{-8}
Multiple case 2 (24 × 24)	40	5.10×10^{-8}	4.46×10^{-8}	5.89×10^{-8}

corresponding ARE. This results in slightly lower accuracy for the computed closed-loop eigenvalues and the solution of the ARE than in the other two cases, for which the measures of conditioning have similar orders of magnitude.

From Tables 4.7 and 4.8, we note that both Algorithms III and IV perform as well as or slightly better than the conventional approach (HQR3-EXCHNG) in all cases except one. The exception corresponds to the results obtained using Algorithm IV for the boiler model, which is the most ill conditioned of the examples considered in Tables 4.7 and 4.8. The slightly lower accuracy in this case can be explained as follows: For a Hamiltonian matrix with some ill-conditioned eigenvalues, there is a significant loss of accuracy in computing the corresponding eigenvalues of the square of the Hamiltonian matrix using Van Loan's method [12]. Consequently, the shifts used in Algorithm IV for very ill-conditioned eigenvalues will have poor accuracy.

5. Concluding remarks. In this paper, some numerical issues in computing a basis for the stable invariant subspace of a Hamiltonian matrix have been discussed. In particular, two alternatives to the use of the EXCHNG subroutine for reordering eigenvalues of a Hamiltonian matrix have been proposed. These were derived using certain properties of Hamiltonian matrices and were shown to require significantly less computation than the conventional approach (using the HQR3 and EXCHNG subroutines). Numerical experiments that have been carried out suggest that the proposed algorithms give accuracy that is often comparable to or better than that obtained using the conventional approach.

Acknowledgments. The authors wish to thank the referees and the review editor for their helpful comments and suggestions.

REFERENCES

[1] A. J. LAUB, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, AC-24 (1979), pp. 913–921.

[2] R. V. PATEL AND N. MUNRO, *Multivariable System Theory and Design*, Pergamon Press, Oxford, 1982.

[3] A. J. LAUB, *Invariant subspace methods for the numerical solution of Riccati equations*, in The Riccati Equation, S. Bittanti, A. J. Laub, and J. Willems, eds., Springer-Verlag, New York, 1991.

[4] A. BUNSE-GERSTNER, R. BYERS, AND V. MEHRMANN, *Numerical methods for algebraic Riccati equations*, in Proc. Workshop on the Riccati Equation in Control, Systems and Signals, Italy, pp. 107–115, Pitagora Editrice, Bologna, Italy, June 1989.

[5] G. H. GOLUB AND C. VAN LOAN, *Matrix Computations*, 2nd Ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[6] G. W. STEWART, *HQR3 and EXCHNG: Fortran subroutines for calculating and ordering the eigenvalues of a real Hessenberg matrix*, ACM Trans. Math. Software, 2 (1976), pp. 275–280.

[7] D. S. FLAMM AND R. A. WALKER, *Remark on Algorithm* 506, ACM Trans. Math. Software, 8 (1982), pp. 219–220.

[8] A. MACFARLANE, *An eigenvector solution of the optimal linear regular problem*, J. Electr. Control, 14 (1965), pp. 643–654.

[9] C. C. PAIGE AND C. VAN LOAN, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 4 (1981), pp. 11–32.

[10] R. BYERS, *A Hamiltonian QR-algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.

[11] A. BUNSE-GERSTNER AND V. MEHRMANN, *A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 1104–1113.

[12] C. VAN LOAN, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 16 (1984), pp. 233–251.

[13] J. J. DONGARRA, S. HAMMARLING, AND J. H. WILKINSON, *Numerical Considerations in Computing Invariant Subspaces*, Tech. Report TM-11704, Oak Ridge National Laboratory, Oak Ridge, TN, November 1990.

[14] G. AMMAR AND V. MEHRMANN, *On Hamiltonian and Symplectic Hessenberg Forms*, Tech. Report 90-002, Universität Bielefeld, Germany, 1990.

[15] J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1963.

[16] G. AXELBY, A. J. LAUB, AND E. J. DAVISON, *Further discussion on the calculation of transmission zeros*, Automatica, 14 (1978), pp. 403–405.

[17] R. BYERS, *Hamiltonian and Symplectic Algorithms for the Algebraic Riccati Equation*, Ph.D. thesis, Cornell University, Ithaca, NY, 1983.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 299–312, January 1994

023

SPARSITY PATTERNS WITH HIGH RANK EXTREMAL POSITIVE SEMIDEFINITE MATRICES*

J. WILLIAM HELTON†‡, DANIEL LAM†, AND HUGO J. WOERDEMAN†§

Abstract. This article concerns the positive semidefinite matrices $M_+(G)$ with zero entries in prescribed locations; that is, matrices with given sparsity graph G. The issue here is the rank of the extremals of the cone $M_+(G)$. It was shown in [J. Agler, J. W. Helton, S. McCullough, and L. Rodman, *Linear Algebra Appl.*, 107 (1988), pp. 101–149] that the key in constructing high rank extreme points resides in certain atomic graphs G called blocks and superblocks. The k-superblocks are defined to be sparsity graphs G that contain an extreme point of rank k while containing (in an extremely strong sense) no graph with the same property. The goal of this article is to write down all graphs that are superblocks. The article succeeds completely for $k \leq 4$ and it lists necessary conditions in general as well as sufficient conditions. The subject is closely related to orthogonal representations of graphs as studied earlier in [L. Lovász, M. Saks, and A. Schrijver, *Linear Algebra Appl.*, 114/115 (1989), pp. 439–454] and in the previously mentioned paper by Alger et al. Indeed, the paper is an extension of the findings of Alger et al.

Key words. extremal matrix, order of a graph, superblocks, orthogonal representation

AMS subject classifications. primary 05C50; secondary 05B20, 15A57

Introduction. Let G be an undirected graph without multiple edges or loops. Let $V(G)$ $(= \{1, \ldots, n\})$ denote the set of vertices of G and $E(G) \subset V(G) \times V(G)$ the set of edges. Note that the absence of loops means that $(i,i) \notin E(G)$, $i = 1, \ldots, n$. Define $M_+(G)$ to be the closed cone of all positive semidefinite $n \times n$ real symmetric matrices whose (i,j) entry is zero whenever $(i,j) \in E(G)$. (Note the difference in definition compared to preceding papers on the subject ([AHMR], [HPR]); in those papers the zero entries would correspond to edges in the complementary graph.)

A matrix A in $M_+(G)$ is called an extremal when each additive decomposition of A in $M_+(G)$ is a trivial one, i.e., A is an *extremal* when $A = B + C$ with $B, C \in M_+(G)$ yields that $B, C \in \operatorname{span}\{A\}$. We say that G has *order* k if k is the maximum of the ranks of extremals in $M_+(G)$.

We are interested in determining the order of a given graph. The graphs of order 1 have a very elegant characterization (see [AHMR], [PPS]) based on the main result in [GJSW]. The general case turns out to be very hard, and, therefore, some reductions must be made. Recall from [AHMR] the following definitions. A graph G is called a *k-block* if G has order k but no induced subgraph has order k. (The graph $\hat{G}$ is an *induced subgraph* of G if $V(\hat{G}) \subset V(G)$ and $E(\hat{G}) = E(G) \cap (V(\hat{G}) \times V(\hat{G}))$.) In terms of matrices this means that for a k-block G any rank k extremal in $M_+(G)$ does not have zero rows or columns. A full description of k-blocks, $k = 1, 2, \ldots$, would give a solution to our problem, since the order of a graph equals the maximal k for which the graph has a k-block as an induced subgraph ([AHMR, Thm. 1.2]). Classifying k-blocks is based on the study of much better behaved objects called k-superblocks.

*Received by the editors October 22, 1990; accepted for publication (in revised form) May 4, 1992.

†Department of Mathematics, University of California at San Diego, La Jolla, California 92093 (helton@osiris.ucsd.edu, hugo@cs.wm.edu).

‡Supported in part by the Air Force Office of Scientific Research and the National Science Foundation.

§Supported by the Netherlands Organization for Scientific Research (NWO).

A graph G is called a *k-superblock* when it is a k-block that does not properly contain another k-block. (In this paper "$\hat{G}$ is *contained* in G" always means $V(\hat{G}) \subset V(G)$ and $E(\hat{G}) \subset E(G)$.) In terms of matrices, this means that as soon as you allow some of the zero entries prescribed by G to be nonzero there are no extremals of rank k anymore. It is true (see [AHMR]) that any k-block (or any order k graph, for that matter) contains a k-superblock. However, to obtain all k-blocks assuming one knows how to characterize k-superblocks still requires work.

The following theorem gives necessary conditions for a sparsity pattern to be a k-superblock.

THEOREM 0.1. *Let G be a k-superblock. Then the following are true:*

(i) $\#E(G) = \frac{1}{2}(k+2)(k-1)$;

(ii) *G contains no $K_{p,q}$, $p+q \geq k+1$;*

(iii) *For all $i_1, \ldots, i_m \in V(G)$ with $1 \leq m < k$ we have that*

$$(0.1)\quad \#\{(i,j) \in E(G) \mid i \text{ or } j \in \{i_1, \ldots, i_m\}\} < \frac{1}{2}(k+2)(k-1) - \frac{1}{2}(k-m+2)(k-m-1) = \frac{1}{2}m(2k+1-m)\,.$$

Conversely, when $k = 1, 2, 3,$ or 4 these conditions imply that G is a k-superblock.

Here $K_{p,q}$ denotes the bipartite graph described by

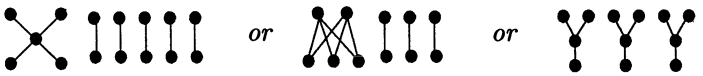
$$V(K_{p,q}) = \{1, \ldots, p+q\},\ E(K_{p,q}) = \{(i,j) \mid 1 \leq i \leq p,\ p+1 \leq j \leq p+q\}\,.$$

The necessary conditions (i) and (ii) were established earlier in [AHMR]. Condition (iii) is implied by (i) and (ii) when $k = 1, 2, 3$ but not when $k \geq 4$. For $k = 1, 2, 3$ the k-superblocks were described earlier in [AHMR], and indeed they are precisely the graphs which satisfy the necessary conditions in Theorem 0.1. For $k = 4$ this is also true (as stated in Theorem 0.1). This follows from the description of 4-superblocks given in the next theorem, which is the second main result in this paper.

THEOREM 0.2. *Let G be a graph with nine edges. The following are equivalent:*

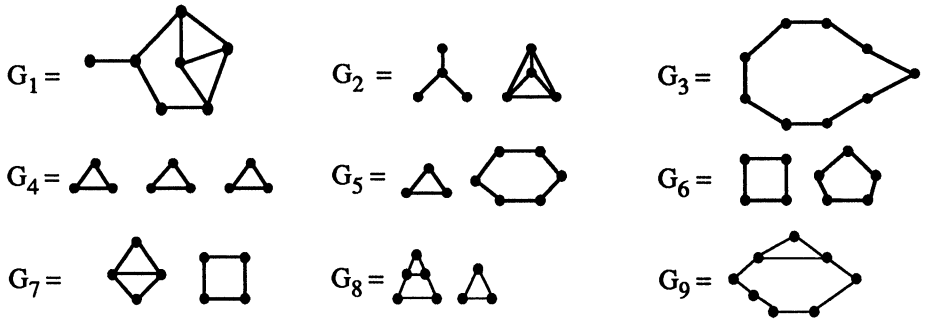
(i) *G is a 4-superblock;*

(ii) *G cannot be obtained from*

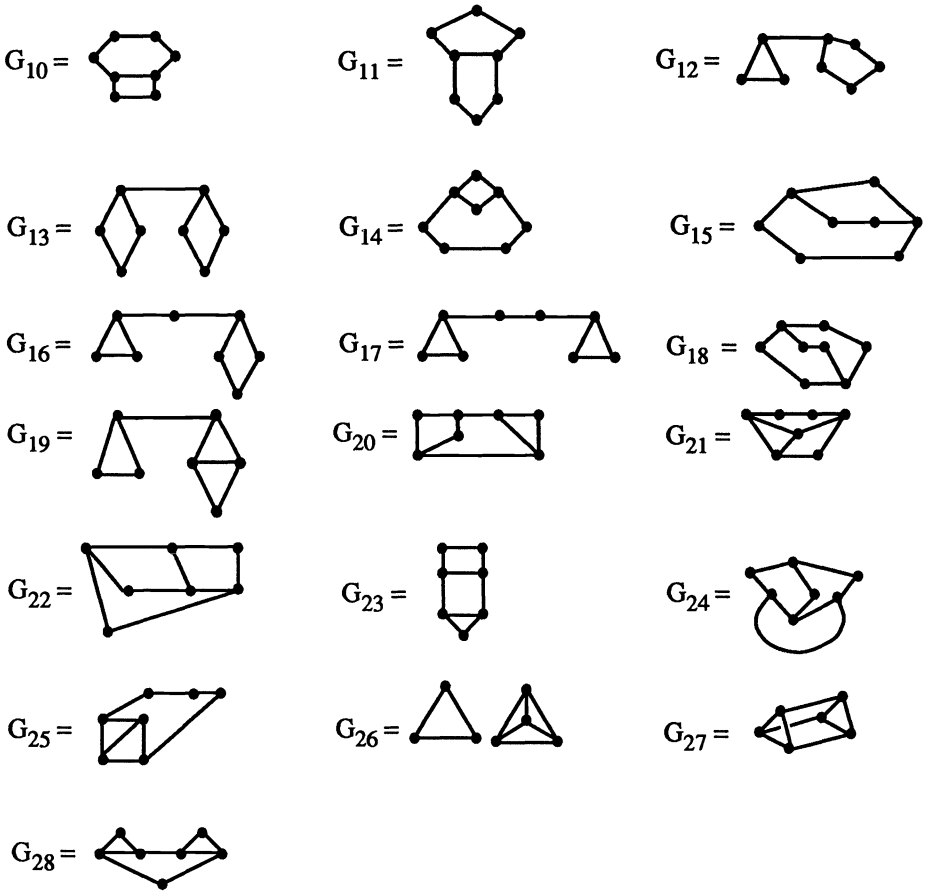


by identifying vertices;[1]

(iii) *G is a graph which, after identifying vertices, is one of the following* 28 *graphs:*



[1]See the definition of a collapse of a graph in §2.



Note that (ii) in Theorem 0.2 is a restatement of the necessary conditions in Theorem 0.1 for $k = 4$.

In §2 we develop some results which give sufficient conditions for a graph to be a k-superblock. Unfortunately these conditions do not equal the necessary conditions in Theorem 0.1. In the case when $k = 4$, for instance, 17 of the 28 graphs in Theorem 0.2(iii) meet the necessary conditions of Theorem 0.1, but not our general sufficient conditions. To prove that these 17 graphs are 4-superblocks, we used a computer program employing Mathematica (using integer arithmetic). It is natural to ask whether this gap in the theory can be dissolved in the following way.

SPECULATION 0.3. *Let G be a graph satisfying* (i), (ii), *and* (iii) *in Theorem* 0.1. *Then G is a k-superblock.*

We shall point out in the end of §1 what remains to be done in order to prove this speculation. We used our Mathematica program to check some likely candidates for counterexamples (with $k = 5$ and 6), but so far we have been unsuccessful (partly because the program is very slow when k is large).

1. Making extremals in $M_+(G)$. From [AHMR] one can deduce the following recipe for making *all* extremals in $M_+(G)$ of rank k.

Let G be a graph, and let $k \leq \#V(G)$.

(Step I) Find an assignment $f: V(G) = \{1, \dots, n\} \longrightarrow \mathbb{R}^k$ such that
(a) $\langle f(i), f(j) \rangle = 0$, $(i,j) \in E(G)$;
(b) span $\{f(j) \mid j \in V(G)\} = \mathbb{R}^k$.

(Step II) Check if all $M = M^T \in \mathbb{R}^{k \times k}$ satisfying

$$\langle Mf(i), f(j) \rangle = 0, \qquad (i,j) \in E(G)\,, \tag{1.1}$$

are multipliers of the $k \times k$ identity matrix I_k.

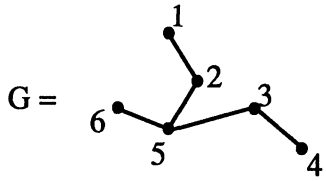
(Step III) If so, then

$$\begin{pmatrix} f(1)^T \\ \vdots \\ f(n)^T \end{pmatrix} \begin{pmatrix} f(1) \cdots f(n) \end{pmatrix}$$

is an extremal of rank k in $M_+(G)$.

An assignment $f: V(G) \longrightarrow \mathbb{R}^k$ such that (a) in Step I holds is called an *orthogonal representation* of G. Such representations were introduced and studied independently in [LSS] and [AHMR] (for quite different reasons).

Example. Let



Then

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & -1 \\ 0 & 0 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 5 \\ 0 & 1 & 1 & -1 & 0 & 7 \end{pmatrix}$$

are extremals in $M_+(G)$ of ranks 1 and 2, respectively. There are no extremals of rank ≥ 3 because of dimension counting. Indeed, if $k = 3$, either $f(5) = 0$ or span $\{f(2), f(3), f(6)\}$ has dimension of at most 2. This results in at most four constraints, appearing in (1.1), on M. This can never force M to be a scalar multiple of I_3. Furthermore, when $k > 3$, the five constraints (1.1) on M are too few to force M to be a scalar multiple of I_k.

The type of arguments used in the example led in [AHMR] to the following result.

Let G be a graph. If order $G = k$, then $\#E(G) \geq \frac{1}{2}(k+2)(k-1)$. Furthermore, if order $G = k$ and $\#E(G) = \frac{1}{2}(k+2)(k-1)$, then G contains no $K_{p,q}$, $p+q \geq k+1$.

Let us now prove that a k-superblock must satisfy the conditions (i), (ii), and (iii) in Theorem 0.1.

Proof of the necessary part of Theorem 0.1. Let G be a k-superblock. In particular, order $G = k$, and thus the recipe works for a representation f, say. Let $i_1, \dots, i_k \in V(G)$ be such that $f(i_j)$, $j = 1, \dots, i_k$, span $\mathbb{R}^k$. Furthermore, choose

distinct $i_{k+1},\dots,i_l$ such that each edge in $E(G)$ has an endpoint in $\{i_1,\dots,i_l\}$ and write

$$f(i_p)=\sum_{j=1}^{k}\lambda_j^{(p)}f(i_j)\,,\qquad p=k+1,\dots,l.$$

Let M be a $k\times k$ matrix, and put $w_j=Mf(i_j)$, $j=1,\dots,k$. Let $u_1^{(j)},\dots,u_{p_j}^{(j)}$ denote the vertices adjacent to i_j that are not in the set $i_1,\dots,i_{j-1}$. Put

$$U_j=[f(u_1^{(j)})\cdots f(u_{p_j}^{(j)})]\,,\qquad j=1,\dots,l.$$

Put

$$W=\begin{bmatrix} U_1 & & & \lambda_1^{(k+1)}U_{k+1} & \cdots & \lambda_1^{(l)}U_l\\ & U_2 & & & & \\ & & \ddots & \vdots & & \vdots\\ & & U_k & \lambda_k^{(k+1)}U_{k+1} & \cdots & \lambda_k^{(l)}U_l\end{bmatrix}$$

and

$$\Sigma=\begin{bmatrix} -f(i_2) & -f(i_3) & 0 & \cdots & -f(i_k) & 0 & \cdots & 0\\ f(i_1) & 0 & -f(i_3) & \cdots & 0 & -f(i_k) & \cdots & 0\\ 0 & f(i_1) & f(i_2) & \cdots & 0 & 0 & & 0\\ \vdots & \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -f(i_k)\\ 0 & 0 & 0 & \cdots & f(i_1) & f(i_2) & \cdots & f(i_{k-1})\end{bmatrix}.$$

That is to say, if $\frac{1}{2}j(j-1)<p\le\frac{1}{2}(j+1)j$ then column p in Σ has on the $p-\frac{1}{2}j(j-1)$ row the entry $-f(i_{j+1})$ and on the $(j+1)$ row the entry $f(i_{p-j(j-1)/2})$ $(j=1,\dots,k-1)$. Note that W is of size $k\times\#E(G)$ and Σ of size $k\times\frac{1}{2}k(k-1)$. The equations (1.1) are equivalent to

$$\operatorname{col}(w_i)_{i=1}^{k}\in\text{cokernel } W$$

and the symmetry of M is ensured by

$$\operatorname{col}(w_i)_{i=1}^{k}\in\text{cokernel }\Sigma\,.$$

Checking Step II in the recipe now comes down to checking that

$$\text{cokernel}\,[W,\Sigma]=\operatorname{span}\{\operatorname{col}(f(i_j))_{j=1}^{k}\}\,. \tag{1.2}$$

Note that the inclusion $\supset$ in (1.2) is always fulfilled since f is an orthogonal representation. We assume that the recipe works, and therefore (1.2) holds. Suppose that $\#E(G)>\frac{1}{2}(k+2)(k-1)$, then the number of columns in $[W,\Sigma]$ is $\ge k^2$. Since the columns in Σ are linearly independent, and because of (1.2), we can remove a column in W without changing the cokernel of $[W,\Sigma]$. But removing a column in W corresponds to removing an edge in G, yielding that G properly contains a k-block. Thus $\#E(G)=\frac{1}{2}(k+2)(k-1)$.

Now it follows from the quoted result before the proof that (ii) holds. It remains to prove (iii). First note that if the recipe works for f, it also works for an orthogonal representation $\tilde f$ with the property that $\|f(i)-\tilde f(i)\|$ is small enough. Indeed, such a perturbation will not destroy the invertibility of a $(k^2-1)\times(k^2-1)$ invertible submatrix

of $[W, \Sigma]$. Since the graph does not contain any $K_{p,q}$'s, $p+q \geq k+1$, we know from [LSS] that in any neighborhood of f we can find an orthogonal representation $\tilde{f}$ that has the property that any set of k representing vectors are linearly independent (in the terms of [LSS]: $\tilde{f}$ is in *general position*). Thus without loss of generality we may assume that f has the latter property. Choose now $i_1, \ldots, i_m \in V(G)$, $m < k$, arbitrary. Then $f(i_1), \ldots, f(i_m)$ are linearly independent. Choosing the $i_1, \ldots, i_m$ as the first m vertices in $\{i_1, \ldots, i_k, i_{k+1}, i_l\}$ we can set up the matrix W and Σ as before. After permutation of columns of the matrix $[W, \Sigma]$ we obtain the matrix

$$\begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ 0 & \Lambda_{22} \end{bmatrix}, \tag{1.3}$$

where

$$\Lambda_{11} = \begin{bmatrix} U_1 & & -f(i_2) & \cdots & -f(i_m) & 0 & \cdots & 0 \\ & U_2 & f(i_1) & & 0 & f(i_m) & & 0 \\ & \ddots & 0 & & 0 & 0 & & 0 \\ & & \vdots & & \vdots & \vdots & \ddots & \vdots \\ & & 0 & & 0 & 0 & & -f(i_{m-1}) \\ & U_m & 0 & & f(i_1) & f(i_2) & & f(i_{m-1}) \end{bmatrix}.$$

Note that

$$\operatorname{col}(f(i_j))_{j=1}^{m} \in \operatorname{coker} \Lambda_{11}.$$

But then, since the cokernel of (1.3) should have dimension exactly equal to 1, the matrix Λ_{22} should have at least as many columns as rows. Since Λ_{22} is of size

$$(k^2 - mk) \times \left[(k^2 - 1) - \left(\#\{(i,j) \in E(G) \mid i \text{ or } j \in \{i_1, \ldots, i_m\}\} + \frac{1}{2} m(m-1) \right) \right],$$

this inequality precisely yields (0.1). □

In order to prove Speculation 0.3, it remains to prove that for a graph satisfying (i), (ii), and (iii) there is an orthogonal representation f such that the matrix $[W, \Sigma]$, constructed in the proof of Theorem 0.1, has a one-dimensional cokernel.

2. A sufficiency result. For a vertex $v \in V(G)$ the *degree* is defined to be the number of adjacent vertices.

THEOREM 2.1. *Let P be a graph with an induced subgraph G that satisfies*

(i) $\#E(G) \geq \frac{1}{2}(k+2)(k-1)$;

(ii) *G contains no $K_{p,q}$'s, $p+q \geq k+1$;*

(iii) *for any $\{i_1, \ldots, i_m\} \subset V(G)$ with $1 \leq m \leq k-1$,*

$$\#\{(i,j) \in E(G) \mid i \text{ or } j \in \{i_1, \ldots, i_m\}\} < \#E(G) - \frac{1}{2}(k-m+2)(k-m-1); \tag{2.1}$$

(iv) *G has $k-1$ vertices $\{u_1, \ldots, u_{k-1}\}$ of degree $k-1$ such that one of the vertices u_j $(1 \leq j \leq k-1)$ is not adjacent to any u_i, $j \neq i \in \{1, \ldots, k-1\}$.*

Then order $P \geq k$.

The conclusion remains true when (iv) *is replaced by*

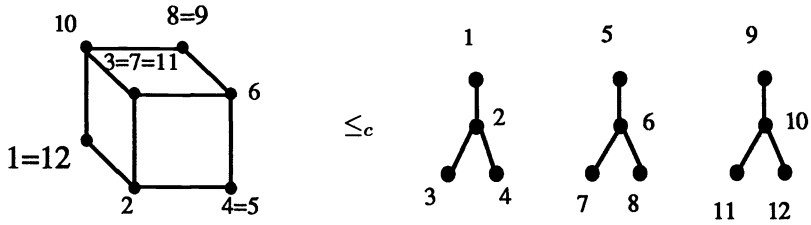
(v) *G has at least k vertices of degree $k-1$, and, after deleting m of these vertices, the remaining graph always contains a $K_{p,q}$, $p+q \geq k-m+1$, $m = 1, \ldots, k-2$.*

It should be noted that it follows from the last part of the proof of Theorem 0.1 that (iii) is a necessary condition for G to be of order $\geq k$.

Before proving the theorem, let us make some remarks. Condition (iv) is a very stringent one since it requires the subgraph G to be fairly condensed. However, combined with the next result from [AHMR] the theorem shows that a substantial number of graphs have order $\geq k$.

We introduce the following partial ordering on graphs. We say that $G \leq_c \tilde{G}$ (G is a *collapse* of $\tilde{G}$) if G can be obtained from $\tilde{G}$ by identifying vertices without identifying edges. For example,

(2.2)



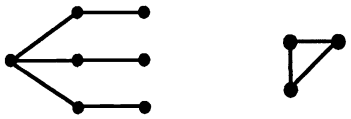
PROPOSITION 2.2. *Let G and $\tilde{G}$ be graphs satisfying $G \leq_c \tilde{G}$. Then*

$$\text{order } G \leq \text{ order } \tilde{G}.$$

This is a restatement of Theorem 4.7 in [AHMR].

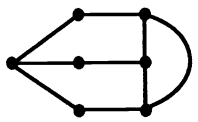
Example. Theorem 2.1 and Proposition 2.2 together prove that the order of

(2.3)



is ≥ 4. Indeed, the graph

(2.4)



has order ≥ 4 by Theorem 2.1, and (2.4) is a collapse of (2.3). Therefore, (2.3) has order ≥ 4. In fact, (2.3) has precisely order 4 since, by Theorem 0.1, a graph of order ≥ 5 should at least have 14 edges, the number of edges of a 5-superblock. □

In order to prove Theorem 2.1, we need some auxiliary results. Recall that an orthogonal representation $f : V(G) \to \mathbb{R}^k$ is in general position if any set of k representing vectors is linearly independent.

LEMMA 2.3. *Let G be a graph that does not contain $K_{p,q}$'s $p + q \geq k + 1$, and let $f: V(G) \to \mathbb{R}^k$ be an orthogonal representation in general position. If $M = M^T \in \mathbb{R}^{k \times k}$ satisfies* (1.1), *then for any $v \in V(G)$ with degree equal to $k - 1$ the vector $f(v)$ is an eigenvector of M.*

Proof. Both $f(v)$ and $Mf(v)$ belong to the orthogonal complement of span $\{f(u) \mid (u, v) \in E(G)\}$. Since deg $v = k - 1$, the vectors $Mf(v)$ and $f(v)$ both belong to a one-dimensional space. □

Recall from [LSS] that an orthogonal representation f of a graph G is called *faithful* if $\langle f(u), f(v)\rangle = 0$ if and only if $(u,v) \in E(G)$.

LEMMA 2.4. *Let G be a graph that contains no $K_{p,q}$'s, $p+q \geq k+1$. Further, suppose that u is a vertex of degree $k-1$ that has r nonadjacent vertices of degree $k-1$. Then for any faithful orthogonal representation f of G in $\mathbb{R}^k$ in general position the set of symmetric matrices M satisfying* (1.1) *is either* $\operatorname{span}\{I_k\}$ *or contains an element of* $\operatorname{rank} < k-r$.

Proof. Let M be symmetric such that (1.1) holds. Since $f(u)$ is an eigenvector of M (at λ_0, say), which is not orthogonal to r other eigenvectors, the dimension of the eigenspace of M at λ_0 is at least the dimension of the span of $f(u)$ and these other r eigenvectors. Since f is in general position we obtain that

$$\operatorname{rank}(M - \lambda_0 I) \leq \max\{0, k-(r+1)\}.$$

Since $M - \lambda_0 I$ is symmetric and satisfies (1.1), the lemma is proved. □

PROPOSITION 2.5. *Let G be a graph that contains no $K_{p,q}$'s, $p+q \geq k+1$. Then for every orthogonal representation $f: V(G) \to \mathbb{R}^k$ there exists a symmetric matrix M of rank* 1 *satisfying* (1.1) *if and only if there is a set V of at most $k-1$ vertices in G such that any edge in G has an endpoint in V.*

Proof. Suppose such a set V exists. Choose $0 \neq w \in \mathbb{R}^k$ such that $\langle w, f(v)\rangle = 0$ for any $v \in V$. It is easy to check that $M := ww^T$ satisfies (1.1).

In order to prove the *only if* part, let f be an orthogonal representation in general position (such an f exists: Theorem 1.1 in [LSS]). Also let $M = ww^T$ with $w \neq 0$ satisfy (1.1). Then for all edges $(i,j) \in E(G)$

$$\langle w, f(i)\rangle\langle w, f(j)\rangle = 0,$$

thus w is orthogonal to one of the endpoints of each edge in G. Since w can be orthogonal to at most $k-1$ linearly independent vectors we obtain the proposition above. □

We are now ready to prove Theorem 2.1.

Proof of Theorem 2.1. Suppose (i)–(iv) hold. We have to prove that order $G \geq k$. Let $f: V(G) \to \mathbb{R}^k$ be in general position and faithful (existence is assured by Theorem 1.1 and (the proof of) Corollary 1.4 in [LSS]). Lemma 2.4 yields, because G satisfies (iv), that either the set of symmetric matrices M satisfying (1.1) is span $\{I_k\}$ or has an element of rank 1. Since G satisfies condition (iii) in the theorem, the latter possibility is ruled out. (Since, if M is symmetric of rank 1 satisfying (1.1), then Proposition 2.5 yields that (2.1) is violated (for $m = k-1$).)

Now suppose that (i), (ii), (iii), and (v) hold. Let $f: V(G) \to \mathbb{R}^k$ be an orthogonal representation in general position. Further suppose that M is positive semidefinite, satisfies (1.1) and has rank d, with $1 < d < k$. (We can always assume that M is positive semidefinite, since if M satisfies (1.1) then any linear combination of M and I_k satisfies (1.1) also.) Since G has k vertices of degree $k-1$, the representing vectors corresponding to these vertices are a basis of eigenvectors of M. But then $k-d$ of these vertices represent the kernel. Delete those vertices. Then from (1.1) it easily follows that $M^{1/2}f$, defined by

$$(M^{1/2}f)(v) = M^{1/2}(f(v)),$$

is an orthogonal representation in general position of the remaining graph. Since (v) holds, this is impossible by Theorem 1.1 in [LSS]. As before, a symmetric M satisfying

(1.1) cannot have rank 1. But then all symmetric M satisfying (1.1) must be in span $\{I_k\}$. □

3. 4-superblocks. In this section we prove Theorem 0.2.

Proof of Theorem 0.2. The implication (i) ⇒ (ii) is merely a restatement of the necessary conditions in Theorem 0.1 in this special case.

For the proof of (ii) ⇒ (iii) we determine the $\leq_c$-minimal elements in the set of graphs $\mathfrak{G}$ described under (ii) in Theorem 0.1, i.e., $\mathfrak{G}$ is the set of graphs with nine edges that are not a collapse of one of the three graphs under (ii). The result is given in the following proposition.

PROPOSITION 3.1. *The $\leq_c$-minimal elements in $\mathfrak{G}$ are the graphs G_i, $i = 1, \ldots, 28$, defined in Theorem* 0.1(iii).

The proof requires a lemma.

LEMMA 3.2. *Let $G \in \mathfrak{G}$ be $\leq_c$-minimal. Then G does not have a vertex u of degree* 1 *and a vertex v of degree ≤ 2 such that the distance $d(u, v)$ between u and v is larger than* 2.

Proof. Suppose that G has vertices u and v with $\deg u = 1$, $\deg v \leq 2$ and $d(u, v) > 2$. By identifying u and v we do not create a degree-4 node. Suppose we create a $K_{2,3}$. Then we must have had

(1)  or (2) 

with three other edges. In case (1) these three other edges do not form a subgraph $K_{1,3}$; otherwise $G \leq_c 3 \times K_{1,3}$, which contradicts $G \in \mathfrak{G}$. Here $3 \times K_{1,3}$ denotes the graph on the right-hand side of (2.2). But then G must be disconnected. This yields that we can identify one of the vertices in a connected component of G not containing u with u and obtain a graph in G that is $\leq_c$-smaller than G. For the second possibility (2), the reasoning is similar.

Suppose that by identifying u and v we create a graph that is $\leq_c$-smaller than $3 \times K_{1,3}$. Since G has no vertices of degree 4, G must, in this case, have vertices of degree ≤ 2 besides u and v. But then u may be identified with one of these other vertices and stay in the class $\mathfrak{G}$. □

Proof of Proposition 3.1. In the reasoning to follow we shall quite frequently use the fact that a nine-edge graph satisfies

$$\sum_{u \in V(G)} \deg(u) = 18. \tag{3.1}$$

Let us now determine the $\leq_c$-minimal elements G in $\mathfrak{G}$.

Case 1. G has a vertex u of degree 1.

Let v denote the vertex adjacent to u. When $\deg v = 1$ the remaining vertices should have degree 3 (Lemma 3.2). This is impossible by (3.1). Consider now the case when $\deg v = 2$, and let $w \neq u$ be the other neighbor of v. The cases, $\deg w = 1$ and $\deg w = 2$, are quickly disregarded again by using Lemma 3.2 and (3.1). When $\deg w = 3$, the only graph one obtains by requiring that all other vertices have degree 3 (which must be the case because of Lemma 3.2) contains a $K_{2,3}$ and is therefore not in $\mathfrak{G}$. Consequently, we are left with the case that $\deg v = 3$. Let $\{u, w_1, w_2\}$ denote the adjacency set of v. When $\deg w_1 = \deg w_2 = 1$ we obtain G_2 as the only possibility. The cases $\{\deg w_1 = 1,\ \deg w_2 = 2\}$, $\{\deg w_1 = \deg w_2 = 2\}$, and

$\{\deg w_1 = \deg w_2 = 3\}$ are quickly discarded, leaving the case $\{\deg w_1 = 2, \deg w_2 = 3\}$. From this we obtain G_1 as the only possibility.

Case 2. All vertices of G have degree 2.

Then G must have nine vertices and consist of circuits. The only possibilities are G_3, G_4, G_5, and G_6.

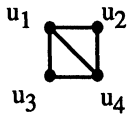
We are left with the cases that G has some vertices of degree 3 and some of degree 2. Because of (3.1), the number of degree-3 vertices must be even.

Case 3. G has two vertices of degree 3 and six of degree 2.

First we consider the case when the vertices of degree 3 are adjacent. Disconnected graphs with these requirements are easily recognized to be G_7 and G_8. Considering the possible paths between the vertices of degree-3 nodes, one obtains the graphs G_9, G_{10}, and G_{11} in case there are three different paths; when there is only one path, the graphs G_{12} and G_{13} are obtained (note that two paths between the degree-3 nodes is impossible). In a similar way, one obtains the graphs G_{14}–G_{18} for the case when vertices of the degree 3 have distance 2 or 3.

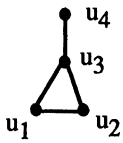
Case 4. G has four vertices of degree 3 and three of degree 2.

Since $G \not\geq_c 3 \times K_{1,3}$, there should be no three vertices of degree 3 that are all nonadjacent. Let u_1, u_2, u_3, and u_4 denote the vertices of degree 3. When $u_1, \ldots, u_4$ are all adjacent to one another, one obtains the only possibility, G_{26}. When they form



,

one obtains G_{25}. When $u_1, \ldots, u_4$ form a square ($K_{2,2}$), one obtains G_{23}. When $u_1, \ldots, u_4$ form



,

one obtains G_{19} and G_{21}. The case when $u_1, \ldots, u_4$ form a connected line gives possibilities G_{20}, G_{22}, and G_{28}. The last case is when $u_1, \ldots, u_4$ form a line of three vertices and an isolated vertex. This gives as the only possibility, G_{24}.

Case 5. G only has vertices of degree 3.

Here G_{27} is the only possibility.

This proves Proposition 3.1. □

This concludes the proof of (ii) $\Longrightarrow$ (iii).

To prove (iii) $\Rightarrow$ (i) we need to show that for the graphs G_1–G_{28} we can find a rank-4 extremal in $M_+(G)$. Then Proposition 2.2 yields that all graphs under (iii) have a rank-4 extremal. The graphs G_1, G_2, G_{19}, G_{20}, G_{21}, G_{22}, G_{24}, G_{27}, and G_{28} have a vertex of degree 3 that is not adjacent to two other vertices of degree 3. Therefore, by Theorem 2.1 (i)–(iv), the order of these graphs is ≥ 4. But then, since the numbers of edges is smaller than 14 we obtain that the order is at most 4, giving equality. The graph G_{26} has as its complement the graph $K_{3,4}$. Consequently, G_{26} has order 4, by Theorem 6.1 in [HPR]. The graphs G_{23} and G_{28} are recognized to have order ≥ 4 by Theorem 2.1 using (i)–(iii) and (v).

The remaining graphs G_3–G_{18} and G_{25} are dealt with by "brute force." A program using Mathematica (using integer arithmetic) produced for us the following rank-4 extremals in $M_+(G)$ for G in G_3–G_{18}, G_{25}. For each G_i this rank-4 extremal is given by $A_i^T A_i$, where A_i, $i = 3, \ldots, 18, 25$, is given by

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & 2 & -2 & -2 & \frac{9}{22} & \frac{37}{4} & -\frac{32}{45} \\ 0 & 2 & -2 & -\frac{5}{2} & -\frac{9}{2} & -3 & \frac{8}{11} & -3 & -\frac{386}{135} \\ 0 & -3 & 3 & -2 & -1 & -3 & 1 & 1 & 2 \\ 0 & 1 & 2 & -1 & -3 & -2 & -3 & -2 & 2 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 1 & 2 & -3 & 0 & 0 & -3 & -5 & -6 & \frac{11}{7} \\ 0 & 1 & 3 & 2 & 4 & 3 & -5 & -1 & \frac{39}{7} \\ 0 & 3 & 2 & 2 & -1 & 3 & 3 & -3 & -3 \\ 0 & 2 & -3 & -3 & 2 & -3 & 3 & 3 & 2 \end{pmatrix}$$

$$A_5 = \begin{pmatrix} 1 & 3 & \frac{1}{3} & -\frac{2}{3} & -12 & 0 & 0 & 6 & \frac{13}{12} \\ 0 & 1 & 2 & -1 & 2 & -2 & \frac{7}{2} & 3 & \frac{1}{2} \\ 0 & -2 & 3 & -2 & 3 & 3 & 2 & -3 & 3 \\ 0 & 1 & 3 & -1 & -3 & 1 & 1 & -1 & -1 \end{pmatrix}$$

$$A_6 = \begin{pmatrix} 1 & 0 & -2 & 1 & \frac{1}{2} & 0 & -3 & -6 & \frac{14}{5} \\ 0 & -2 & -1 & -3 & 1 & 1 & -9 & -2 & -\frac{47}{5} \\ 0 & -3 & 3 & -1 & 2 & 2 & 5 & -1 & 3 \\ 0 & 3 & 2 & 1 & -2 & 1 & -1 & -1 & -1 \end{pmatrix}$$

$$A_7 = \begin{pmatrix} 1 & 2 & \frac{3}{2} & 0 & 0 & \frac{3}{2} & -1 & -1 \\ 0 & -1 & 1 & 3 & \frac{2}{3} & -1 & \frac{1}{2} & -\frac{9}{11} \\ 0 & 1 & -1 & -2 & 2 & -2 & -2 & -\frac{8}{11} \\ 0 & 1 & -1 & 1 & 2 & -2 & 1 & 1 \end{pmatrix}$$

$$A_8 = \begin{pmatrix} 1 & 0 & 0 & -1 & 2 & -14 & 16 & -\frac{9}{50} \\ 0 & -3 & -1 & 3 & -\frac{8}{3} & -9 & 2 & -\frac{103}{50} \\ 0 & -3 & -1 & 2 & 2 & 3 & 3 & 1 \\ 0 & -2 & 3 & 2 & 1 & -2 & 2 & 2 \end{pmatrix}$$

$$A_9 = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & -\frac{15}{2} & -\frac{2}{5} & \frac{97}{8} \\ 0 & 2 & 6 & -\frac{11}{2} & \frac{5}{2} & -1 & 3 & -\frac{1}{20} \\ 0 & -3 & 3 & -3 & -3 & -2 & 1 & 2 \\ 0 & -1 & 3 & -2 & -2 & 2 & 1 & 3 \end{pmatrix}$$

$$A_{10} = \begin{pmatrix} 1 & 0 & 0 & 2 & 1 & 2 & 7 & -\frac{5}{2} \\ 0 & -1 & -3 & -2 & \frac{4}{3} & 0 & -3 & -\frac{35}{6} \\ 0 & 1 & 1 & -1 & 1 & 1 & -1 & 2 \\ 0 & -1 & 3 & 1 & 1 & -3 & -2 & -1 \end{pmatrix}$$

$$A_{11} = \begin{pmatrix} 1 & 0 & 0 & -2 & -\frac{59}{6} & -\frac{10}{3} & \frac{6}{59} & \frac{1062}{277} \\ 0 & -3 & 1 & -\frac{5}{3} & 7 & -2 & 1 & -\frac{2324}{277} \\ 0 & -2 & -1 & 1 & 1 & -1 & -3 & -2 \\ 0 & 1 & -2 & -3 & 3 & 3 & -1 & -2 \end{pmatrix}$$

$$A_{12} = \begin{pmatrix} 1 & 0 & 1 & -3 & 0 & 0 & 9 & \frac{5}{6} \\ 0 & -1 & 5 & 2 & -2 & 4 & -2 & \frac{1}{4} \\ 0 & -3 & -3 & -2 & 2 & 1 & 1 & -3 \\ 0 & 2 & -2 & -2 & -2 & -3 & -2 & 2 \end{pmatrix}$$

$$A_{13} = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & -2 & -1 & \frac{18}{11} \\ 0 & 3 & 3 & 1 & 4 & \frac{2}{3} & \frac{1}{2} & \frac{75}{22} \\ 0 & 2 & -3 & 3 & -3 & 1 & -\frac{1}{2} & 3 \\ 0 & -2 & 3 & -1 & 3 & 2 & -1 & -1 \end{pmatrix}$$

$$A_{14} = \begin{pmatrix} 1 & 0 & 0 & -1 & 3 & 2 & 7 & \frac{16}{11} \\ 0 & 2 & -3 & -4 & 0 & \frac{4}{3} & 1 & -\frac{57}{11} \\ 0 & 2 & -1 & 3 & -1 & -2 & 3 & -1 \\ 0 & -2 & 1 & -1 & -1 & 2 & -2 & 1 \end{pmatrix}$$

$$A_{15} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & -\frac{27}{4} & 2 & -\frac{48}{223} \\ 0 & 2 & 2 & 2 & -\frac{3}{2} & \frac{11}{2} & \frac{5}{2} & -\frac{140}{223} \\ 0 & -3 & -1 & -2 & 3 & 3 & 2 & 2 \\ 0 & 1 & 3 & -3 & -3 & -2 & -1 & 2 \end{pmatrix}$$

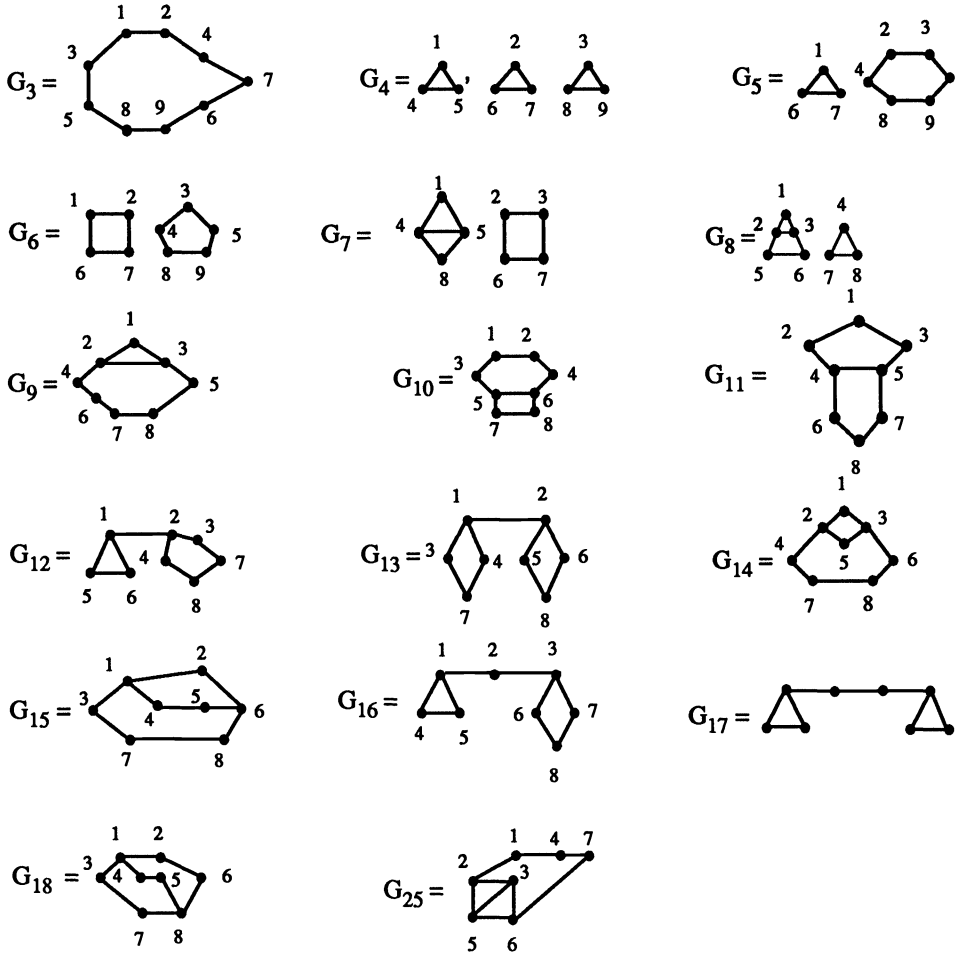
$$A_{16} = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & \frac{25}{2} & \frac{47}{2} & \frac{8}{11} \\ 0 & -1 & 12 & -3 & -5 & -3 & -3 & \frac{48}{11} \\ 0 & -2 & -3 & 3 & -3 & -3 & 3 & -2 \\ 0 & 3 & 2 & -2 & 3 & 1 & -1 & -2 \end{pmatrix}$$

$$A_{17} = \begin{pmatrix} 1 & 0 & -2 & -\frac{5}{2} & 0 & 0 & -\frac{6}{5} & -\frac{80}{61} \\ 0 & 2 & -3 & 3 & -3 & -\frac{5}{3} & -1 & \frac{96}{61} \\ 0 & -3 & -3 & -2 & 2 & 2 & -3 & 2 \\ 0 & 3 & -1 & 2 & -3 & 3 & -3 & -2 \end{pmatrix}$$

$$A_{18} = \begin{pmatrix} 1 & 0 & 0 & 0 & -3 & -1 & 3 & \frac{294}{151} \\ 0 & -1 & -3 & 1 & 12 & 10 & -\frac{5}{3} & -\frac{99}{302} \\ 0 & 3 & 1 & 3 & -2 & 2 & 1 & -\frac{117}{302} \\ 0 & -2 & -3 & 2 & -3 & -2 & 2 & -3 \end{pmatrix}$$

$$A_{25} = \begin{pmatrix} 1 & 0 & -2 & 0 & \frac{2}{3} & \frac{30}{7} & -\frac{81}{20} \\ 0 & 3 & -\frac{4}{3} & -1 & 2 & -\frac{27}{14} & -9 \\ 0 & -1 & 2 & -3 & 3 & 1 & 2 \\ 0 & 3 & 2 & 3 & -1 & 2 & -1 \end{pmatrix}$$

Here the numbering of the nodes is basically from top to bottom and from left to right, or, more explicitly, given by the following.



It is easy to check by hand that the representations of G_3–G_{18} and G_{25}, indicated by A_3–A_{18} and A_{25}, respectively, satisfy the conditions in Step I of the recipe. In order to check that Step II is satisfied, one has to go through more elaborate calculations. These checks were made using the Mathematica program. □

Acknowledgments. We wish to thank Leiba Rodman for attracting our attention to the reference [LSS] and Neola Crimmins for her masterful production of this manuscript.

REFERENCES

[AHMR] J. Agler, J. W. Helton, S. McCullough, and L. Rodman, *Positive semidefinite matrices with a given sparsity pattern*, Linear Algebra Appl., 107 (1988), pp. 101–149.

[GJSW] R. Grone, C. R. Johnson, E. Marques de Sá, and H. Wolkowicz, *Positive definite completions of partial Hermitian matrices*, Linear Algebra Appl., 58 (1984), pp. 109–124.

[HPR] J. W. Helton, S. Pierce, and L. Rodman, *The ranks of extremal positive semidefinite matrices with a given sparsity pattern*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 407–423.

[LSS] L. Lovász, M. Saks, and A. Schrijver, *Orthogonal representations and connectivity of graphs*, Linear Algebra Appl., 114/115 (1989), pp. 439–454.

[PPS] V. I. Paulsen, S. C. Power, and R. R. Smith, *Schur products and matrix completions*, J. Funct. Anal., 85 (1989), pp. 151–178.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 313–320, January 1994

024

NORMS OF HADAMARD MULTIPLIERS*

CARL C. COWEN†, MICHAEL A. DRITSCHEL†, AND RICHARD C. PENNEY†

Abstract. For B a fixed matrix, the authors consider the problem of finding the norm of the map $X \mapsto X \bullet B$, where $\bullet$ is the Hadamard or entrywise product of matrices and the norm of a matrix is its spectral norm. Using techniques from the theory of Kreĭn spaces, the problem is rewritten for Hermitian matrices as a minimization problem whose solution, for small matrices, can be obtained from standard optimization software. The Hadamard multiplier norm for an arbitrary matrix is given in terms of a Hermitian extension. The results are applied to refute a conjecture of R. V. McEachin concerning the value of a constant in an operator inequality.

Key words. Hadamard product, Schur product, Kreĭn space, operator inequality

AMS subject classifications. 15A60, 15A45, 65F35, 15A23

1. Introduction and definitions.

By the *Hadamard product*, also called the Schur product, we mean the entry-wise product of matrices: if A and B are $m \times n$ matrices, their Hadamard product, $A \bullet B$ is the $m \times n$ matrix whose entries are $a_{jk}b_{jk}$. In this paper, we study the norm of the operator on the set $\mathcal{M}_n$ of $n \times n$ matrices given by $X \mapsto X \bullet B$, for a fixed B. Throughout, $\langle \cdot, \cdot \rangle$ will denote the standard Euclidean inner product on $\mathbf{C}^n$ and by the norm of a vector, we will mean the Euclidean norm. The norm of a matrix is its norm as an operator on this Hilbert space (i.e., the spectral norm), and the norm K_B of the Hadamard multiplier is its norm as a linear map on the operators in $\mathcal{M}_n$. If A is in $\mathcal{M}_n$, we will denote its (Hilbert space) adjoint by A^*. For A an $n \times n$ matrix with columns $A_1, A_2, \ldots, A_n$, let

$$c(A) = \max\{\|A_1\|, \|A_2\|, \ldots, \|A_n\|\}.$$

Haagerup [3] (or see [8, pp. 110–116] or [6]) showed that if B is an $n \times n$ matrix, then the norm of B as a Hadamard multiplier is

$$K_B = \min\{c(S)c(R) : S^*R = B\}.$$

We show that when B is Hermitian, there is a Kreĭn space (defined below) associated with B and an optimal factorization for which S^* is $R^\times$, the Kreĭn space adjoint of R. We then obtain K_B as a minimum of $c(UR_0)^2$ over the set of Kreĭn unitaries, U, where $B = R_0^\times R_0$ is a particular factorization of B. In addition, we show that K_B is the value at all local minima of this function, so we can use this factorization and standard software to approximate K_B, confident that the minimizing sequence generated is not approaching an uninteresting local minimum. We can restrict our attention to Hermitian matrices because if A is any matrix, it is easily seen that K_A is the same as K_B, where B is the Hermitian matrix

$$B = \begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix}.$$

A vector space $\mathcal{K}$ with a scalar product $[\cdot, \cdot]$ is a Kreĭn space if there is an inner product $\langle \cdot, \cdot \rangle$ that makes $\mathcal{K}$ a Hilbert space so that $[x, y] = \langle Jx, y \rangle$, where J is self-adjoint and unitary, that is, $J = J^* = J^{-1}$. The matrix J is called a *fundamental*

* Received by the editors July 1, 1991; accepted for publication (in revised form) June 3, 1992. This work was supported in part by National Science Foundation grants.

† Department of Mathematics, Purdue University, West Lafayette, Indiana 47907 (cowen@math.purdue.edu; mad@math.purdue.edu; rcp@math.purdue.edu).

symmetry of the Kreĭn space. The prototypic example of a finite dimensional Kreĭn space is the vector space $\mathbf{C}^n$, together with the scalar product obtained from the diagonal matrix whose first diagonal entries are 1's and whose remaining diagonal entries are -1's. If $J = I$, the resulting Kreĭn space is the Euclidean (Hilbert) space.

If $\mathcal{K}_1$ and $\mathcal{K}_2$ are Kreĭn spaces with fundamental symmetries J_1 and J_2, respectively, and $A: \mathcal{K}_1 \to \mathcal{K}_2$ is a linear map, its Kreĭn space adjoint, denoted $A^\times$, is $J_1 A^* J_2$. Indeed,

$$[Ax, y]_2 = \langle J_2 Ax, y\rangle = \langle x, A^* J_2^* y\rangle = \langle J_1 x, J_1 A^* J_2 y\rangle = [x, J_1 A^* J_2 y]_1.$$

We will need to use the fact that if $\mathcal{K}$, $\mathcal{K}_1$, and $\mathcal{K}_2$ are Kreĭn spaces with fundamental symmetries J, J_1, and J_2 and $R : \mathcal{K} \longrightarrow \mathcal{K}_1$ and $S : \mathcal{K} \longrightarrow \mathcal{K}_2$ are invertible and satisfy $R^\times R = S^\times S$, then there is a Kreĭn unitary $U : \mathcal{K}_1 \longrightarrow \mathcal{K}_2$ so that $S = UR$. Indeed, the invertibility of R and S shows that for v and w in $\mathcal{K}_1$, if $v = Rx$ and $w = Ry$, then $U = SR^{-1}$ is Kreĭn unitary:

$$[v, w]_1 = [Rx, Ry]_1 = [R^\times Rx, y] = [S^\times Sx, y] = [Sx, Sy]_2 = [Uv, Uw]_2.$$

(An infinite dimensional version of this fact is addressed by Theorem 2.12 of [2].) In particular, this means that $\mathcal{K}_1$ and $\mathcal{K}_2$ are equivalent as Kreĭn spaces. For maps between Kreĭn spaces, J-selfadjoint, J-unitary, and so forth, will have the obvious meanings.

2. The results. We begin by showing that when the Hadamard multiplier is Hermitian, there is a solution of the Haagerup extremal problem that has a special symmetry. (This is an extension of Theorem 3 of [1] and its proof.)

THEOREM 2.1. *Let B be a Hermitian matrix. There is a fundamental symmetry J and an $n \times n$ matrix R (regarded as a map from Euclidean space to the Kreĭn space determined by J) so that*

$$K_B = c(R)^2 \quad \textit{and} \quad B = R^\times R.$$

Proof. We first assume that B is invertible.

Paulsen, Power, and Smith prove [9, p. 161] that among all positive matrices of the form

$$P = \begin{pmatrix} X & B \\ B & Y \end{pmatrix},$$

K_B is the smallest $\max\{x_{jj}, y_{jj} : j = 1, \ldots, n\}$ that can occur (see also [6, Thm. 3.1]). If P_0 is a matrix of this form with $K_B = \max\{x_{jj}, y_{jj}\}$ and

$$J_0 = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix},$$

then J_0 is selfadjoint and

$$J_0 P_0 J_0 = \begin{pmatrix} Y & B \\ B & X \end{pmatrix}$$

is positive. Now, $P_1 = (P_0 + J_0 P_0 J_0)/2$ is positive and

$$P_1 = \begin{pmatrix} W & B \\ B & W \end{pmatrix},$$

where $W = (X + Y)/2$. Moreover,

$$K_B \le \max\{w_{jj}\} \le \frac{1}{2}\max\{x_{jj}\} + \frac{1}{2}\max\{y_{jj}\} \le K_B,$$

so P_1 is also an extension of B that gives K_B.

Let

$$\mathcal{E} = \left\{ P = \begin{pmatrix} Z & B \\ B & Z \end{pmatrix} : P \text{ positive and } K_B = \max\{z_{jj}\} \right\}.$$

From the above symmetrization argument, the set $\mathcal{E}$ is nonempty and it is clearly closed. Therefore, $\mathcal{E}$ contains minimal elements with respect to the usual order on positive matrices. We claim that any minimal element has rank n.

To this end, suppose P is a minimal element of $\mathcal{E}$. Let

$$P = \begin{pmatrix} S & R \\ 0 & Q \end{pmatrix}^* \begin{pmatrix} S & R \\ 0 & Q \end{pmatrix}$$

be the Cholesky factorization of P (see, for example, [5, pp. 114 and 407]) in which Q, R, and S are $n \times n$ matrices with Q and S upper triangular having nonnegative diagonal entries. This means

$$\begin{pmatrix} Z & B \\ B^* & Z \end{pmatrix} = \begin{pmatrix} S^*S & S^*R \\ R^*S & R^*R + Q^*Q \end{pmatrix}.$$

Since $B = S^*R$ has rank n, each diagonal entry of S is positive, and it is easily seen that the rank of P is n if and only if $Q = 0$. If Q were not zero, then define $\tilde{P}$ by

$$\tilde{P} = \begin{pmatrix} S & R \\ 0 & 0 \end{pmatrix}^* \begin{pmatrix} S & R \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} S^*S & S^*R \\ R^*S & R^*R \end{pmatrix}, \tag{1}$$

and let $\tilde{P}_0 = (\tilde{P} + J_0\tilde{P}J_0)/2$. Since the diagonal entries of R^*R are no more than those of $R^*R + Q^*Q = Z$, the largest diagonal entry of $\tilde{P}_0$ is no more than K_B. On the other hand, since $\tilde{P}_0$ is a positive matrix with B in the upper right corner, the largest diagonal entry of $\tilde{P}_0$ is at least K_B. Thus, $\tilde{P}_0$ is in $\mathcal{E}$. Since

$$P = \tilde{P}_0 + \frac{1}{2}\begin{pmatrix} Q^*Q & 0 \\ 0 & Q^*Q \end{pmatrix},$$

$\tilde{P}_0$ is strictly less than P unless $Q = 0$. The minimality of P in $\mathcal{E}$ implies $Q = 0$ and rank P is n.

Therefore, a minimal element of $\mathcal{E}$ has the form (1), and there are $n \times n$ matrices R and S satisfying $B = S^*R$ with $S^*S = R^*R$ and $K_B = c(S)c(R)$. Since the diagonal entries of R^*R and S^*S are the squares of the column norms of R and S, respectively, it follows that $c(R) = c(S)$ and $K_B = c(R)^2$. The equality $S^*S = R^*R$ implies that there is a unitary matrix J so that $S = JR$. Since B is Hermitian, we find

$$R^*JR = R^*S = B^* = B = S^*R = R^*J^*R.$$

The invertibility of B implies R is invertible, so the above equality means $J = J^*$. Thus, J is a fundamental symmetry. Now the equality $S = JRI$ gives $S^* = IR^*J = R^\times$, as we were to show.

Now if B is not invertible, we can find a sequence B_k of invertible Hermitian matrices so that $\lim B_k = B$. Continuity gives $\lim K_{B_k} = K_B$. For each k, there are fundamental symmetries J_k and matrices R_k so that $B_k = R_k^\times R_k$, that is, $J_k^* = J_k^{-1} = J_k$ and $B_k = R_k^* J_k R_k$, and $K_{B_k} = c(R_k)^2$. Since all matrices are finite and their entries are uniformly bounded (the J_k's are Euclidean unitary and $\lim c(R_k)^2 = K_B$), we can find subsequences converging to J and R, respectively, and by continuity,

$$J^* = J^{-1} = J, \qquad B = R^* J R = R^\times R,$$

and $K_B = \lim K_{B_k} = \lim c(R_k)^2 = c(R)^2$. Thus, the conclusion holds in this case as well. □

This theorem is an extension of Schur's theorem on the norm of a Hadamard multiplier of a positive matrix. To see this, note that if B is positive, then J is the identity and the theorem reduces to $K_B = c(R)^2$, where $R^*R = B$. Since the diagonal entries of B are just the squares of the norms of the columns of R, as Schur proved [10], K_B is the largest entry on the diagonal of B. (This is not a new proof of the Schur theorem as the Schur theorem was one of the ingredients of the Paulsen, Power, and Smith result.)

Theorem 2.1, together with the observation that $B = R^\times R = R_0^\times R_0$ implies $R = UR_0$ for some Kreĭn unitary map, means we can reformulate the extremal problem of Haagerup.

COROLLARY 2.2. *If B is Hermitian and J is a fundamental symmetry so that $B = R_0^\times R_0$, then*

$$K_B = \min\{c(UR_0)^2 : U \text{ is } J\text{-unitary}\}.$$

Moreover, if B is Hermitian, it is easy to find R_0 satisfying $B = R_0^\times R_0$. There is a unitary U so that UBU^* is diagonal. Now if J is the sign of the diagonal matrix and E is the positive diagonal matrix so that $E^2 J = UBU^*$, then we may take $R_0 = EU$ viewed as an operator from Euclidean space to the Kreĭn space $\mathcal{K}$ determined by J. In this case,

$$R_0^\times R_0 = (EU)^* JEU = U^* EJEU = U^* E^2 JU = B.$$

If J_1 and J are unitarily equivalent, say $W^* J_1 W = J$, then we can use $\tilde{R}_0 = WEU$, viewed as an operator from Euclidean space to $\mathcal{K}_1$, the Kreĭn space determined by J_1. The remarks made at the end of the introduction show that all such factorizations of B arise in this way.

Theorem 2.1 can be extended trivially to a slightly more general setting: if J_0 is a fundamental symmetry whose entries are 1's, 0's, and -1's and B is J_0-selfadjoint, instead of just Euclidean Hermitian (I-selfadjoint), then the conclusion of the theorem holds. This is easily seen to be true since in this case $J_0 B$ is Hermitian and $K_B = K_{J_0 B}$. An important example of such a fundamental symmetry is the matrix J_0 whose nonzero entries are 1's on the cross diagonal, that is, $u_{i,n-i+1} = 1$ and $u_{ij} = 0$ otherwise. In [1], the triangular truncation matrix was viewed as a J_0-selfadjoint matrix and the other Kreĭn space of the factorization $R^\times R$ was also taken to be the space with $J = J_0$.

If we define the function γ on the group $\mathcal{G}$ of Kreĭn unitaries by $\gamma(U) = c(UR_0)^2$, Corollary 2.2 shows that K_B is the minimum of γ on $\mathcal{G}$. In most cases, the minimum is attained on a set that includes a nontrivial subgroup of $\mathcal{G}$, so it is not attained at a unique point. It is computationally important to know whether γ has uninteresting

local minima, that is, local minima at which the value is larger than the value at the absolute minimum. We prove that it does not!

It is well known, and easily shown by taking second derivatives, that if H is Hermitian and v is a vector in $\mathbf{C}^n$ with $Hv \neq 0$, then $t \mapsto \|e^{tH}v\|^2$ is strictly convex.

The following lemma is the particular case of the Cartan decomposition of a semisimple Lie group that applies to our situation (see, for example, [11, p. 162]). We give an elementary proof for the sake of completeness.

LEMMA 2.3. *If U is Kreĭn unitary, then U may be written uniquely as $U = VP$, where V is Kreĭn and Euclidean unitary and P is Kreĭn unitary and Euclidean positive.*

Proof. Let U be Kreĭn unitary and let $U = VP$ be the unique (Euclidean) polar decomposition of U, where V is Euclidean unitary and P is Euclidean positive. Since U is invertible, so are V and P and

$$U = (U^\times)^{-1} = (V^\times)^{-1}(P^\times)^{-1}.$$

Since P is positive and since congruence preserves positivity, $P^\times = JP^*J = J^*PJ$ is positive, and this implies $(P^\times)^{-1}$ is positive. Since V and J are Euclidean unitary and the Euclidean unitaries are a group,

$$(V^\times)^{-1} = (JV^*J)^{-1} = J^{-1}(V^*)^{-1}J^{-1} = JVJ$$

is also Euclidean unitary. This means $U = (V^\times)^{-1}(P^\times)^{-1}$ is a (Euclidean) polar decomposition of U also. Since the polar decomposition is unique, we find $V = (V^\times)^{-1}$ and $P = (P^\times)^{-1}$, which means that V and P are Kreĭn unitary. □

COROLLARY 2.4. *If U is Kreĭn unitary, U can be factored as $U = Ve^H$, where V is Kreĭn and Euclidean unitary and $H = H^* = -H^\times$.*

Proof. We need only prove that $P = e^H$ where H has the required properties. Since P is Euclidean positive, P has a unique Hermitian logarithm, H. The fact that P is Kreĭn unitary means $e^{H^\times} = P^\times = P^{-1} = e^{-H}$ from which it follows that $H = -H^\times$. □

We are now ready to prove that no uninteresting local minima exist.

THEOREM 2.5. *Let R be an $n \times n$ matrix and let γ be the function on the group $\mathcal{G}$ of Kreĭn unitaries given by $\gamma(U) = c(UR)^2$. If γ has a local minimum at U_1, then*

$$\gamma(U_1) = \min\{\gamma(U) : U \in \mathcal{G}\}.$$

Proof. Let U_0 be a Kreĭn unitary so that $\gamma(U_0)$ is the minimum value of γ on $\mathcal{G}$. Replacing R by U_0R in the definition of γ if necessary, we see that, without loss of generality, we may assume $U_0 = I$.

Suppose γ has a local minimum at the Kreĭn unitary U_1. By Corollary 2.4, U_1 can be factored as Ve^H, where V is both Kreĭn and Euclidean unitary and H satisfies $H^* = H = -H^\times$. Let f be defined by $f(t) = \gamma(Ve^{tH})$, that is, $f(t)$ is the maximum of $\|Ve^{tH}R_j\|^2 = \|e^{tH}R_j\|^2$, where $R_1, R_2, \ldots, R_n$ are the columns of R. We noted above that each of the functions $\|e^{tH}R_j\|^2$ is convex. Since the maximum of convex functions is convex, it follows that f is convex. Now V is Euclidean unitary so $\gamma(V) = \gamma(I)$ is the global minimum, and

$$f(0) = \gamma(V) \leq \gamma(U_1) = \gamma(Ve^H) = f(1).$$

If $\gamma(U_1) > \gamma(I)$, the convexity of f implies

$$\gamma(Ve^{tH}) = f(t) \leq (1-t)f(0) + tf(1) < f(1) = \gamma(U_1)$$

for $0 < t < 1$. In particular, this is true for t arbitrarily close to 1, so $\gamma(U_1)$ cannot be a local minimum.

Thus, the only local minima of γ are those points at which γ achieves a global minimum. □

These results may be used for calculating the Hadamard multiplier norm of Hermitian matrices. We begin by factoring B as $B = R_0^{\times} R_0$. Lemma 2.3 permits expressing K_B as the solution of the minimization problem

$$K_B = \min\{c(PR_0)^2 : \ P \text{ is Euclidean positive and Kreĭn unitary}\}, \tag{2}$$

and Corollary 2.4 permits expressing K_B as the solution of the minimization problem

$$K_B = \min\{c(e^H R_0)^2 : H^* = H = -H^{\times}\}. \tag{3}$$

These minimization problems can be approached with standard software since if J is the fundamental symmetry

$$J = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix},$$

then P is Euclidean positive and J–unitary if and only if there is a matrix X so that

$$P = \begin{pmatrix} \sqrt{I + XX^*} & X \\ X^* & \sqrt{I + X^*X} \end{pmatrix},$$

and $H^* = H = -H^{\times}$ if and only if there is a matrix Y so that

$$H = \begin{pmatrix} 0 & Y \\ Y^* & 0 \end{pmatrix}.$$

We have used both methods to approximate K_B for some small matrices B. Our limited experience suggests that using (3) gives slightly better results and that the minimum is relatively shallow, which causes difficulty with the convergence.

Equations (2) and (3) express K_B as minimization problems that can be parametrized as above by matrix variables X and Y. Using $X_0 = Y_0 = (1, 0)$ and $X_1 = Y_1 = (0, 1)$, examples of matrices R_0 can be found to show that these functions are not necessarily convex in the variables X and Y, so the convexity property in the proof of Theorem 2.5 is not the restriction of such a broader convexity property.

We believe the results above form the theoretical basis of an approach for finding the norms of matrices as Hadamard multipliers on $\mathcal{M}_n$.

3. Application to the conjecture of McEachin. In the analysis of a perturbation problem of interest in operator theory, McEachin [7] needed to estimate the Hadamard multiplier norm of matrices such as

$$M = \begin{pmatrix} \frac{1}{7} & \frac{1}{5} & \frac{1}{3} & 1 \\ \frac{1}{5} & \frac{1}{3} & 1 & -1 \\ \frac{1}{3} & 1 & -1 & -\frac{1}{3} \\ 1 & -1 & -\frac{1}{3} & -\frac{1}{5} \end{pmatrix}.$$

McEachin conjectured that

$$K_M = \sqrt{\frac{410 + \sqrt{30}}{864}} + \sqrt{\frac{410 - \sqrt{30}}{864}} > 1.37770223779394.$$

We used MATLAB on a Macintosh II, with the function `minimax` from MATLAB's Optimization Toolbox. We factored M through the Kreĭn space whose fundamental symmetry is the matrix with 1's on the cross diagonal and used the minimization problem from (3). We found that if

```
R =
   0.29345772706387  -0.87151443181904  -0.38760407738941   0.75646065056181
  -0.82107382515359  -0.30980696295447  -0.19650488973146  -0.12891228276962
  -0.78049759723731   0.17648441164311   0.47999514387418   0.82208465702786
   0.09080760857704  -0.70074080677351   0.87900753539208  -0.33619623611316
```

and J is the fundamental symmetry with diagonal 1, 1, -1, and -1, then $M = R^{\times}R = R^*JR$ and the corresponding estimate for K_M is 1.37770218499455. Of course, in the absence of error estimates, this calculation does not really mean a great deal. However, using the calculated value for R and *exact arithmetic* with MAPLE on a Macintosh II to find S satisfying $S^*R = M$, the Haagerup result shows that

$$K_M \leq c(S)c(R) < 1.37770218499457,$$

from which it follows that McEachin's conjectured value for K_M is too large.

While the above calculations do prove McEachin's conjecture is incorrect, they do not really show that our estimate for K_M is close to correct. Because the unitary matrices are the extreme points of the unit ball in the space of matrices, there is a unitary U_0 at which K_M is achieved, that is, $\|M \bullet U_0\| = K_M$. After a slight modification, Corollary 7 of [1] applies to this case and the discussion there indicates how a maximizing unitary may be computed from an optimal factorization $M = R_0^{\times}R_0$. We will use this construction to get a lower bound for K_M.

In [1], the important fundamental symmetry is the matrix with 1's on the cross diagonal, denoted J', and the relation $T^* = J'TJ'$ replaces selfadjointness. Since M and J' each have two positive eigenvalues, the Kreĭn space induced by J' can be used for the Kreĭn space occurring in the factorization of M so that Corollary 7 of [1] applies to $T = J'M$. Suppose $M = R_0^{\times}R_0$ is an optimal factorization of M so that $J'M = J'R_0^{\times}R_0 = S_0^*R_0$ is an optimal factorization of $J'M$ where $S_0^* = J'R_0^{\times}$. It was shown that, in this case, there is a positive diagonal matrix D so that $U_0 = D^{-1}S_0^{-1}R_0J'DJ'$ is a maximizing unitary and $U_0^* = J'U_0J'$. Writing W for $S_0^{-1}R_0$ and d_j for the jth diagonal entry of D, the symmetry relation for U_0 implies

$$d_j^2 = \frac{w_{jn}}{w_{1,n-j+1}}d_1^2.$$

Since we may choose $d_1 = 1$, the optimal factorization leads to D and to U_0. Using our approximate optimal factorization, we computed what we expected to be approximately an optimal unitary. This calculation (using MATLAB on a Macintosh II) produces a matrix U that is close to unitary, and we computed a lower bound for K_M from U. Interestingly, however, better results are achieved by using the unitary V from the polar factorization, $U = VP$ where $P = P^*$, of our approximate unitary:

```
V =
   0.52557371154599   0.29440955823111   0.31639397962744   0.73279610766532
   0.29440955823111   0.28352110536812   0.67353109047329  -0.61586903234281
   0.31639397962744   0.67353109047329  -0.62858513101496  -0.22612265010007
   0.73279610766532  -0.61586903234281  -0.22612265010007  -0.18050968589915
```

From this unitary, we get the estimate

$$K_M \geq \frac{\|V \bullet M\|}{\|V\|} = 1.37770218499280.$$

Combining the two estimates, we find that $K_M = 1.37770218499$ to 11 decimal digits.

REFERENCES

[1] J. R. ANGELOS, C. C. COWEN, AND S. K. NARAYAN, *Triangular truncation and finding the norm of a Hadamard multiplier*, Linear Algebra Appl., 170 (1992), pp. 117–136.

[2] M. A. DRITSCHEL, *The essential uniqueness property for operators on Kreĭn spaces*, preprint, 1990; J. Func. Anal., to appear.

[3] U. HAAGERUP, *Decompositions of completely bounded maps on operator algebras*, preprint, 1984.

[4] R. A. HORN, *The Hadamard product*, Proc. Symposia Appl. Math., 40 (1990), pp. 87–169.

[5] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, New York, 1985.

[6] R. MATHIAS, *Matrix completions, norms and Hadamard products*, Proc. Amer. Math. Soc., 117 (1993), pp. 905–918.

[7] R. V. MCEACHIN, *Analysis of an Inequality Concerning Perturbations of Self-adjoint Operators*, Ph.D. thesis, University of Illinois, 1990.

[8] V. I. PAULSEN, *Completely Bounded Maps and Dilations*, Pitman Research Notes in Mathematics 146, John Wiley and Sons, New York, 1986.

[9] V. I. PAULSEN, S. C. POWER, AND R. R. SMITH, *Schur products and matrix completions*, J. Functional Analysis, 85 (1989), pp. 151–178.

[10] I. SCHUR, *Bemerkungen zur theorie de beschränkten bilinearformen mit unendlich vielen veränderlichen*, J. Reine Angew. Math., 140 (1911), pp. 1–28.

[11] N. R. WALLACH, *Harmonic Analysis on Homogeneous Spaces*, Marcel Dekker, New York, 1973.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 321–330, January 1994

025

CYCLIC REDUCTION FOR SPECIAL TRIDIAGONAL SYSTEMS*

S. BONDELI†‡ AND W. GANDER†

Dedicated to G. H. Golub, on the occasion of his 60th birthday.

Abstract. The solution of linear, tridiagonal systems having real, symmetric, diagonally dominant coefficient matrices with constant diagonals is considered. Details of cyclic reduction to solve such systems are discussed. It is proved that the sequence of the diagonal elements produced by the reduction phase of cyclic reduction converges quadratically. This fact is exploited to reduce the number of steps of the reduction phase (special cyclic reduction). An estimate of the rate of convergence of the diagonal elements will be proved, which can be used to determine the number of steps of the reduction phase.

Several possibilities to compute the diagonal elements are discussed and compared.

Key words. cyclic reduction, diagonally dominant matrices, direct methods, symmetric matrices, Toeplitz matrices, tridiagonal matrices

AMS subject classifications. 65F05, 65K10, 65Q05

1. Introduction.

In this paper we study linear systems of n equations of the form

$$(1)\qquad T\mathbf{x} = \begin{pmatrix} a & 1 & & & \\ 1 & a & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & a & 1 \\ & & & 1 & a \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix} = \mathbf{d} \in \mathbb{R}^n,$$

$|a| \geq 2$. The coefficient matrix T has a very special form: it is tridiagonal, symmetric, and a Toeplitz matrix. Such systems arise, for example, when using finite difference methods to solve linear constant-coefficient boundary-value problems in various contexts [8], [12], [17] or in cubic spline interpolation problems with equidistant knots [7], [16].

A number of methods have been proposed for solving such systems: Toeplitz factorization [1], [2], [5], [6], [15]; cyclic reduction [2], [11], [14], [17], [18]; and specialized variants of Gaussian elimination [7], [13], [18]. Each of these requires $O(n)$ scalar operations. Such systems may also be solved using Fourier transforms, but this leads to an $O(n \log n)$ algorithm [2].

In Toeplitz factorization methods one seeks an LU factorization of T in which L is a lower bidiagonal Toeplitz matrix and U is an upper bidiagonal Toeplitz matrix: $T{=}LU$. The diagonal elements of U form a sequence that converges linearly [7], [13] to

$$u = [\, a - \operatorname{sign}(a)\sqrt{a^2 - 4}\,] \,/\, 2\,,$$

assuming $|a| > 2$. This allows the factorization to be terminated early.

* Received by the editors October 2, 1991; accepted for publication (in revised form) June 25, 1992.

† Institut for Scientific Computing, Eidgenössiche Technische Hochschule Zürich, CH-8092 Zürich, Switzerland (gander@inf.ethz.ch).

‡ Present address. Drosselstrasse 22, CH-8038 Zürich, Switzerland (bondeli@zh007.ubs.ubs.arcom.ch).

Another form of early convergence has been observed for cyclic reduction [9]: in the case where T is a block tridiagonal matrix, the norms of the matrix of the off-diagonal elements decrease quadratically with each reduction step. When this norm is less than the machine precision, then the system can be considered diagonal and back-substitution can be started immediately.

In this paper, however, we prove that the diagonal elements a_i form a sequence that converges quadratically to

$$\lim_{i\to\infty} a_i \;=\; \operatorname{sign}(a)\sqrt{a^2-4}$$

for $|a|>2$; furthermore, we give an explicit formula for a_i.

The major goal of this paper is an analysis of cyclic reduction and, consequently, a better understanding of this widely used method. The paper is arranged as follows. First, we will present cyclic reduction. Then we discuss a special cyclic reduction that reduces the number of reduction steps. In a third part, different algorithms to compute the sequence of diagonal elements are discussed. Finally, we give some concluding remarks.

2. Cyclic reduction. The two fundamental operations of cyclic reduction (cyclic odd-even reduction [9]) are the elimination of odd-indexed unknowns and their eventual recovery through back-substitution.

For simplicity we assume that $n=2^m-1$. (Only a simple modification is necessary to generalize cyclic reduction to any n. For more details see [2].)

We multiply equation 1,3, ..., n of (1) by $-1/a$. If we add to each even-numbered equation the two adjacent equations, the result is

$$(2)\qquad \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_1 & b_1 & & \\ & b_1 & \ddots & \ddots & \\ & & \ddots & a_1 & b_1 \\ & & & b_1 & a_1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_4 \\ \vdots \\ x_{n-3} \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} d_{2,1} \\ d_{4,1} \\ \vdots \\ d_{n-3,1} \\ d_{n-1,1} \end{pmatrix} \in \mathbb{R}^{(n-1)/2},$$

where $a_1 = a - 2/a$, $b_1 = -1/a$ and $d_{j,1} = d_j - (d_{j-1} + d_{j+1})/a$ $(j=2,4,\ldots,n-1)$. This first reduction step decoupled the even-numbered equations from the odd ones, yielding a reduced system of order $(n-1)/2$. The reduced equations have the same form as the original ones: the coefficient matrix is a tridiagonal, symmetric, Toeplitz matrix. Therefore, analogously to the elimination above, the reduction process may be applied on (2). We multiply equation 1,3, ...,$(n-1)/2$ of (2) by $-b_1/a_1$ and add adjacent equations. Again a tridiagonal, symmetric Toeplitz system of order $(n-3)/4$ results with diagonal elements $a_2 = a_1 - 2b_1^2/a_1$ and off-diagonal elements $b_2 = -b_1^2/a_1$. Thus, the reduction process may be applied recursively: after $m-1$ reduction steps, only one equation in the unknown $x_{(n+1)/2}$ is left.

The reduction phase may be summarized as follows.

ALGORITHM 2.1. Cyclic reduction: Reduction phase

```
s = 1
for i = 1, 2, ..., m - 1
    b_i = -b_{i-1}^2 / a_{i-1}
    a_i = a_{i-1} - 2 b_{i-1}^2 / a_{i-1}
    t = s
```

$s = 2s$
for $k = s, 2s, 3s, \ldots, (2^{n2-i} - 1)s = n + 1 - s$
$d_{k,i} = d_{k,i-1} - b_{i-1}(d_{k-t,i-1} + d_{k+t,i-1})/a_{i-1}$
end
end

(Here we have taken $a_0 = a$, $b_0 = 1$, and $d_{k,0} = d_k$.)

The back-substitution phase begins by computing $x_{(n+1)/2}$. All the other unknowns may then be determined by applying back-substitution recursively.

ALGORITHM 2.2. Cyclic reduction: Back substitution
$s = (n + 1)/2$
$x_s = d_{s,m-1}/a_{m-1}$
for $i = m - 2, m - 3, \ldots, 0$
$t = s$
$s = s/2$
for $k = s, s + t, s + 2t, \ldots, n + 1 - s$
$x_k = (d_{k,i} - b_i(x_{k-s} + x_{k+s}))/a_i$
end
end

(Note $x_0 = x_{n+1} = 0$.) Algorithms 2.1 and 2.2 require $O(8n)$ floating-point operations.

Since a_i occurs only in the denominator in the reduction as well as in the back-substitution phase we compute the inverse of a_i. Furthermore, since the sequence a_i converges rapidly in the reduction phase, scalar overhead can be saved by using a special cyclic reduction.

3. Special cyclic reduction.

THEOREM 3.1. *Suppose* $|a| > 2$. *Then the sequence of the diagonal elements* a_i *produced by the reduction phase* (*Algorithm* 2.1) *converges quadratically to*

$$\operatorname{sign}(a)\sqrt{a^2 - 4}\,. \tag{3}$$

Proof. Using the identity $b_i = -b_{i-1}^2/a_{i-1}$ (see Algorithm 2.1), it follows for the computation of the diagonal elements

$$\begin{aligned} a_i &= a_{i-1} - 2b_{i-1}^2/a_{i-1} \;=\; a_{i-1} + 2b_i, \\ (a_i - a_{i-1})/2 &= b_i. \end{aligned} \tag{4}$$

Substituting (4) into $b_i = -b_{i-1}^2/a_{i-1}$, we obtain

$$\begin{aligned} (a_i - a_{i-1})/2 &= -\left(\frac{a_{i-1} - a_{i-2}}{2}\right)^2 /a_{i-1}, \\ a_i &= \frac{a_{i-1}^2 + 2a_{i-1}a_{i-2} - a_{i-2}^2}{2a_{i-1}}, \\ &= a_{i-1} - \frac{(a_{i-1} - a_{i-2})^2}{2a_{i-1}}. \end{aligned} \tag{5}$$

Thus we transform (5) into

$$2a_ia_{i-1} - 2a_{i-1}a_{i-2} - a_{i-1}^2 + a_{i-2}^2 = 0, \tag{6}$$

which is a nonlinear homogeneous difference equation of order-two with initial conditions

$$a_0 = a, \qquad a_1 = \frac{a^2 - 2}{a}.$$

We define for $i = 1, 2, \ldots$

$$r_i = 2a_i a_{i-1} - a_{i-1}^2 , \tag{7}$$

and substituting (7) into (6) gives

$$r_i - r_{i-1} = 0 . \tag{8}$$

This linear homogeneous difference equation (8) has constant solutions: $i = 1, 2, \ldots$

$$r_i = r . \tag{9}$$

The initial conditions yield

$$r = r_1 = 2a_0 a_1 - a_0^2 = 2a\frac{a^2 - 2}{a} - a^2 = a^2 - 4 . \tag{10}$$

Substituting (9) into (7) and rearranging gives

$$\begin{aligned} r &= 2a_i a_{i-1} - a_{i-1}^2, \\ a_i &= \frac{a_{i-1}}{2} + \frac{r}{2a_{i-1}}. \end{aligned} \tag{11}$$

Iteration (11) is Newton's method [10, p. 81] to solve the equation $F(x) = x^2 - r = 0$. Since $F'(\sqrt{r}) = 2\sqrt{r} \neq 0$, the sequence a_i defined by (11) converges quadratically to $\sqrt{r} = \sqrt{a^2 - 4}$.

For every choice of $|a_0| > 2$, the sequence is monotone and thus

$$\lim_{i \to \infty} a_i = \operatorname{sign}(a)\sqrt{a^2 - 4}. \qquad \square \tag{12}$$

Table 1 shows some sequences for different initial values a_0=a>2. The last value of each column is equivalent to $\lim_{i \to \infty} a_i = \sqrt{a^2 - 4}$ of each sequence. The values a_i converge very fast to the limit as $|a|$ moves away from 2 (see Table 1).

TABLE 1
Sequences a_i for some initial values a_0=a.

i	a=3.8	a=3.2	a=2.5	a=2.1	a=2.02
0	3.8000000000000	3.2000000000000	2.5000000000000	2.1000000000000	2.0200000000000
1	3.2736842105263	2.5750000000000	1.7000000000000	1.1476190476190	1.0299009900991
2	3.2313758673210	2.4991504854369	1.5117647058824	0.7524402292037	0.5539833733421
3	3.2310988961517	2.4979994645414	1.5000457770657	0.6486670041268	0.3495570413991
4		2.4979991993594	1.5000000006985	0.6403662256270	0.2897812105012
5			1.5000000000000	0.6403124260035	0.2836159557675
6				0.6403124237433	0.2835489454933
7					0.2835489375752

Before we discuss special cyclic reduction in detail, we show an explicit formula for a_i.

LEMMA 3.2. *Suppose* $|a|>2$. *Then the diagonal elements produced by the reduction phase* (*Algorithm* 2.1) *are given by*

$$a_i = \operatorname{sign}(a)\sqrt{a^2-4}\,\coth\left(|y_0|2^i\right), \tag{13}$$

where

$$y_0 = \begin{cases} \ln\left(\sqrt{a^2-4}+a\right) - \ln 2 & \text{if } a > 2, \\ -\ln\left(\sqrt{a^2-4}-a\right) + \ln 2 & \text{if } a < -2. \end{cases}$$

If $|a| = 2$, *then the diagonal and off-diagonal elements are* ($k = 1, 2, \ldots$)

$$b_k = -\operatorname{sign}(a)/2^k, \tag{14}$$

$$a_k = \operatorname{sign}(a)/2^{k-1}\,. \tag{15}$$

Proof. Equations (14) and (15) follow by induction using the formula for the computation of the diagonal elements a_i and off-diagonal elements b_i in the reduction phase (see Algorithm 2.1). For details see [3, pp. 117ff].

We discuss the case $|a|>2$. Using the identity

$$\coth(2\alpha) = \frac{\coth(\alpha)^2+1}{2\coth(\alpha)},$$

and substituting $a_i = \sqrt{r}\coth(y_i)$ into (11), we obtain

$$\begin{aligned} \sqrt{r}\coth(y_i) &= \frac{\sqrt{r}\coth(y_{i-1})}{2} + \frac{r}{2\sqrt{r}\coth(y_{i-1})}, \\ \coth(y_i) &= \coth(2y_{i-1}). \end{aligned} \tag{16}$$

Since $\coth(x)$ is an injective function for $x\in\mathbb{R}\setminus\{0\}$, (16) implies

$$y_i = 2y_{i-1}, \tag{17}$$

which is a first-order linear homogeneous difference equation with solution

$$y_i = y_0 2^i. \tag{18}$$

The initial value y_0 of (18) is given by

$$\coth(y_0) = a_0/\sqrt{r},$$

$$y_0 = \operatorname{arctanh}\left(\frac{\sqrt{a^2-4}}{a}\right).$$

Using the identity

$$\operatorname{arctanh}(\alpha) = \frac{1}{2}\ln\left(\frac{1+\alpha}{1-\alpha}\right) \quad \text{if } |\alpha|<1,$$

and taking into consideration the sign of the initial value $a_0=a$, it follows that

$$y_0 = \begin{cases} \ln\left(\sqrt{a^2-4}+a\right) - \ln 2 & \text{if } a > 2, \\ -\ln\left(\sqrt{a^2-4}-a\right) + \ln 2 & \text{if } a < -2\,. \end{cases}$$

Finally, we obtain the explicit formula

$$a_i = \operatorname{sign}(a)\sqrt{a^2-4}\,\coth\left(|y_0|2^i\right). \qquad \square \tag{19}$$

LEMMA 3.3. *Suppose* $|a_0|>2$. *On a computer with precision* ε, *the sequence* a_i *becomes constant for* $i \geq k_a$, *where*

$$k_a = \left\lceil \frac{\log\left(\log\left(2/\varepsilon+1\right)\right) - \log\left(\log\left(|a|+\sqrt{a^2-4}\right) - \log\left(2\right)\right)}{\log\left(2\right)} - 1 \right\rceil. \tag{20}$$

Here log *denotes the logarithm to the base* 10 *and* $a = a_0$.

Proof. We will assume that $a_0=a>2$, since if a_i is the sequence corresponding to the initial value a, then $-a_i$ is the sequence corresponding to $a_0 = -a$.

We know that the sequence a_i is monotone and converges quadratically to

$$\lim_{i\to\infty} a_i = \sqrt{a^2-4}\,. \tag{21}$$

Therefore, the sequence has converged numerically if

$$\frac{a_i - \lim_{i\to\infty} a_i}{\lim_{i\to\infty} a_i} \leq \varepsilon. \tag{22}$$

Using the identity

$$\tanh\left(\alpha\right) = \frac{\exp\left(2\alpha\right)-1}{\exp\left(2\alpha\right)+1}$$

and formula (13), we obtain

$$a_i = \sqrt{a^2-4}\,\frac{z_i+1}{z_i-1} \quad \text{with} \quad z_i = \left(\frac{a}{2}+\sqrt{\left(\frac{a}{2}\right)^2-1}\right)^{2^{i+1}}. \tag{23}$$

Using (21) we see that

$$\begin{aligned}\frac{a_i - \lim_{i\to\infty} a_i}{\lim_{i\to\infty} a_i} &= \frac{\sqrt{a^2-4}\left(\dfrac{z_i+1}{z_i-1}-1\right)}{\sqrt{a^2-4}} \\ &= \frac{2}{z_i-1}\,,\end{aligned}$$

and to guarantee inequality (22), we must have

$$z_i \geq \frac{2}{\varepsilon}+1. \tag{24}$$

Using definition (23), some elementary computations give

$$i \geq \frac{\log\left(\log\left(2/\varepsilon+1\right)\right) - \log\left(\log\left(a+\sqrt{a^2-4}\right) - \log\left(2\right)\right)}{\log\left(2\right)} - 1, \tag{25}$$

where log denotes the logarithm to the base 10. $\square$

Table 2 shows k_a for the same initial values $a_0 = a$ that are used in Table 1.

This leads to the following algorithm for the solution to (1) with $|a|>2$.

TABLE 2
Values of the convergence index k_a for ε=1.0e−15.

	a=3.8	a=3.2	a=2.5	a=2.1	a=2.02
$i \geq$	3.81	4.08	4.67	5.81	6.97
$k_a =$	4	5	5	6	7

ALGORITHM 3.4. Special cyclic reduction

$z = \log\big(\log(2/\varepsilon + 1.0)\big) - \log\big(\log(|a| + \sqrt{a^2-4}) - \log(2)\big)$
$k_a = \min\{m-1, \lceil z/\log(2)\rceil - 1\}$
$b_0 = 1$
$\alpha_0 = 1/a$
$s = 1$
for $i = 1, 2, \ldots, k_a$
  $b_i = -\alpha_{i-1} b_{i-1}^2$
  $\alpha_i = \alpha_{i-1}/(1 + 2b_i\alpha_{i-1})$
  $t = s$
  $s = 2s$
  for $k = s, 2s, 3s, \ldots, (2^{n2-i} - 1)s = n + 1 - s$
    $d_{ki} = d_{k,i-1} - \alpha_{i-1} b_{i-1}\,(d_{k-t,i-1} + d_{k+t,i-1})$
  end
end

for $k = s, 2s, 3s, \ldots, n + 1 - s$ $\qquad x_k = \alpha_{k_a} d_{s,k_a}$

for $i = k_a - 1, k_a - 2, \ldots, 0$
  $t = s$
  $s = s/2$
  for $k = s, s + t, s + 2t, \ldots, n + 1 - s$
    $x_k = \alpha_i\,(d_{ki} - b_i(x_{k-s} + x_{k+s}))$
  end
end

Note that we put $x_0 = x_{n+1} = 0$, and to save operations we set $\alpha_i = 1/a_i$. The operation count of this method is reduced to $O((8 - 4/2^{k_a})n)$. Significant savings are obtained only when $|a|$ is so large that k_a is small. This means that the reduction process may stop after a few steps. The reason for this is that since the work is reduced by a factor of two at each step in the reduction phase, most of the work is done in the first few steps.

4. Sequence a_i. In this section we discuss and compare several possibilities to compute the diagonal elements of each reduction step of cyclic reduction and present the results of some time measurements.

Table 3 gives two possibilities to compute the sequence a_i using a nonlinear recurrence of order-two or order-one, respectively (see also (5) and (11)). For $|a|>2$ and i large $a_i \approx a_{i-1}$. Therefore, by computing $a_i - a_{i-1}$ in the numerator of (5), cancellation will occur. For that reason the computation of a_i using the recurrence of order-two is unstable.

However, the nonlinear recurrence of order-one (see Algorithm 2 of Table 3) is stable.

TABLE 3
Computation of the sequence a_i using nonlinear recurrences.

Algorithm 1: Nonlinear recurrence of order 2	Algorithm 2: Nonlinear recurrence of order 1
$a_0 = a$ $a_1 = a - 2/a$ for $i = 2, 3, \ldots, m$ $\quad a_i = a_{i-1} - (a_{i-1} - a_{i-2})^2 / (2a_{i-1})$ end	$r = a^2 - 4$ $a_0 = a$ for $i = 1, 2, \ldots, m$ $\quad a_i = (r/a_{i-1} + a_{i-1})/2$ end

TABLE 4
Explicit formulas for the sequence a_i.

Algorithm 3: Explicit formula using tanh	Algorithm 4: Explicit formula
$q = \sqrt{a^2 - 4}$ $y = \log \lvert a + q \rvert - \log 2$ $a_0 = a$ $t = \tanh y$ for $i = 1, 2, \ldots, m$ $\quad t = 2t/(1 + t^2)$ $\quad a_i = q/t$ end	$q = \sqrt{a^2 - 4}$ $c = (a + q)/2$ if $\lvert c \rvert > 1$ then $\quad c = 1.0/c$ $\quad q = -q$ end for $i = 0, 1, \ldots, m$ $\quad c = c^2$ $\quad a_i = q(c+1)/(c-1)$ end

In Table 4, two different explicit formulas for a_i are shown. Equation (19) leads to Algorithm 3 of Table 4, where the identity

$$\tanh(2\alpha) = \frac{2\tanh(\alpha)}{1 + \tanh^2(\alpha)}$$

is used.

Using (23) we derive the second explicit formula for a_i. If $a_0 = a > 2$, then $c = (a+q)/2 > 1$, and hence formula (23) presents the danger of floating-point overflow. However, this can be turned to our advantage by using the identity

$$\frac{(c+1)}{(c-1)} = -\frac{1/c + 1}{1/c - 1} \quad .$$

Note, with the fact that the sequence a_i is monotone for every initial value $|a_0| > 2$, the computation of a_i by each algorithm in Tables 3 and 4 can be stopped machine independently when the monotonicity is violated.

4.1. Comparison of methods. Table 5 summarizes the floating-point operation counts for the methods shown in Tables 3 and 4. Since the scalar overhead of Algorithm 2 is smaller than the other ones, this method is the fastest on a scalar machine.

On a vector computer, such as the CRAY Y/MP, both algorithms using explicit formulas (Table 4) can be vectorized. Therefore, these two methods are faster on a vector machine than the two others using a nonlinear recurrence (see Fig. 1).

TABLE 5
Operation counts.

Algorithm	Scalar operations
Nonlinear recurrence of order 2	$6m - 4$
Nonlinear recurrence of order 1	$3m + 2$
Explicit formula using tanh	$5m + 5$
Explicit formula	$5m + 10$

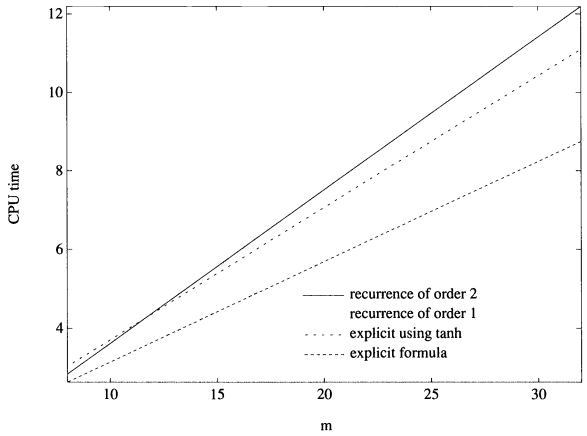


FIG. 1. *Time measurements on a* CRAY Y/MP.

5. Conclusions. For the solution of linear tridiagonal linear symmetric Toeplitz systems we have presented the details of cyclic reduction. We have proved that the sequence of the diagonal elements of the tridiagonal matrices, produced by the reduction phase of cyclic reduction, converge quadratically. This is exploited to reduce the number of steps of the reduction phase of cyclic reduction. In addition, we have developed a formula to compute an a priori estimation of the number of steps of the reduction phase and to compute the diagonal elements to full floating-point precision. Furthermore, we have discussed several algorithms to compute the sequence of the diagonal elements: nonlinear recurrences and two different explicit formulas. On a vector computer the explicit formulas are much faster than the nonlinear recurrences. Without vector facilities, nonlinear recurrence of order-one is the fastest.

Acknowledgment. The authors would like to thank G. H. Golub, one of the inventors of cyclic reduction, for many helpful suggestions.

REFERENCES

[1] M. D. BAKES, *An alternative method of solution of certain tridiagonal systems of linear equations*, Comput. J., 7 (1964), pp. 135–136.

[2] R. F. BOISVERT, *Algorithms for special tridiagonal systems*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 423–442.

[3] S. BONDELI, *Divide and Conquer: Parallele Algorithmen zur Lösung tridiagonaler Gleichungssysteme*, Dissertation ETH Zürich Nr. 9493, Verlag der Fachvereine, Zürich, 1991.

[4] ———, *Divide and conquer: a parallel algorithm for the solution of a tridiagonal linear system of equations*, Parallel Comput., 17(1991), pp 419–434.

[5] D. J. EVANS AND C. D. V. FORRINGTON, *Note on the solution of certain tridiagonal linear systems of linear equations*, Comput. J., 5 (1963), pp. 327–328.
[6] D. FISHER, G. GOLUB, O. HALD, C. LEIVA, AND O. WIDLUND, *On Fourier-Toeplitz methods for separable elliptic problems* , Math. Comp., 28 (1974), pp. 349–368.
[7] W. GANDER, *Latteninterpolation für äquidistante Stützstellen*, in Numerische Prozeduren aus Nachlass und Lehre von Prof. Heinz Rutishauser, W. Gander, L. Molinari, and H. Švecová, eds., Birkhäuser, Basel, 1977, pp. 11–18.
[8] I. GLADWELL AND I. WAIT, *A Survey of Numerical Methods for Partial Differential Equations*, Clarendon Press, Oxford, UK, 1979.
[9] D. HELLER, *Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 484–496.
[10] P. HENRICI, *Elements of Numerical Analysis*, John Wiley & Sons, Inc., New York, 1964.
[11] R. W. HOCKNEY, *A fast direct solution of Poisson's equation using Fourier analysis*, J. Assoc. Comput. Mach., 12(1965), pp. 95–113.
[12] H. B. KELLER, *Numerical Methods for Two-Point Boundary Value Problems*, Blaisdell, London, 1963.
[13] M. A. MALCOLM AND J. PALMER, *A fast method for solving a class of tridiagonal linear systems*, Comm. ACM, 17 (1974), pp. 14–17.
[14] R. REUTER, *Solving tridiagonal systems of linear equations on the* IBM 3090 VF, Parallel Comput., 8(1988), pp. 371–376.
[15] D. J. ROSE, *An algorithm for solving a special class of tridiagonal systems of linear equations*, Comm. ACM, 12 (1969), pp. 234–236.
[16] H. R. SCHWARZ, *Numerische Mathematik*, B. G. Teubner, Stuttgart, 1986, p. 131.
[17] P. N. SWARZTRAUBER, *The methods of cyclic reduction, Fourier analysis, and the FACR algorithm for the discrete solution of Poisson's equation on rectangle*, SIAM Rev., 19 (1977), pp. 490–501.
[18] O. B. WIDLUND, *On the use of fast methods for separable finite difference equations for the solution of general elliptic problems*, in Sparse Matrices and Their Applications, D. J. Rose and R. A. Willoughby, eds., Plenum Press, New York, 1972, pp. 121–131.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 331–346, January 1994

DYNAMIC CONDITION ESTIMATION AND RAYLEIGH–RITZ APPROXIMATION*

PING TAK PETER TANG†

Abstract. It is shown here that the well-known Rayleigh–Ritz approximation method is applicable in dynamic condition estimation. In fact, it can be used as a common framework from which many recently proposed dynamic condition estimators can be viewed and understood. This framework leads to natural generalizations of some existing dynamic condition estimators as well as more convenient alternatives. Numerical examples are also provided to illustrate these claims.

Key words. condition number, singular values, incremental condition estimation

AMS subject classifications. 65F35, 65F05

1. Introduction. Recently a number of dynamic condition estimation schemes have been proposed for a variety of computational situations [1], [9], [6], [10]. These schemes all try to estimate the condition numbers of a sequence of matrices as they evolve in time in some computational process such as QR factorization or recursive least-squares calculations. These estimation schemes are mostly heuristic, and researchers have little theoretical insight on why the schemes work so well in practice. Moreover, although the schemes look alike and are applied to closely related problems, no common framework has been formulated to explain them. Thus, each scheme has been regarded as a different method: incremental condition estimation (ICE), adaptive condition estimation (ACE), adaptive Lanczos estimation (ALE), and ACE for general rank-1 updates (GRACE).

In this paper, we show that many of these dynamic condition-estimation schemes can be viewed from a common and, in fact, well-known framework of Rayleigh–Ritz approximation. This framework provides a number of advantages. The details of the various condition estimators can now be derived naturally and therefore need not be memorized; many known properties of Rayleigh–Ritz approximations can be used directly and thus help in understanding these estimators. The common framework often allows natural generalization of the estimation schemes, and this framework will likely allow relatively easy construction of different estimators for different computational situations.

The rest of the paper is organized as follows. Section 2 discusses the application of Rayleigh–Ritz approximation to condition estimation in general. Section 3 discusses some of the eigenvalue problems that arise from Rayleigh–Ritz approximation; these problems will appear again in subsequent dynamic condition estimators. Section 4 presents the general connection between Rayleigh–Ritz approximations and dynamic condition estimations, and also illustrates how the Rayleigh–Ritz framework can indeed lead to the various dynamic condition estimators. Explanations and generalizations of the various schemes are discussed whenever appropriate. Section 5 presents numerical experiments that illustrate the various points raised in §4. Section 6 offers some concluding remarks.

* Received by the editors February 24, 1992; accepted for publication (in revised form) July 17, 1992. This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U. S. Department of Energy contract W-31-109-Eng-38.

† Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois 60439-4801 (tang@mcs.anl.gov).

2. Condition estimation by Rayleigh–Ritz approximation. The condition number we are concerned with is

$$\kappa_2(R) \equiv \|R\|\|R^{-1}\| = \frac{\sigma_{\max}(R)}{\sigma_{\min}(R)} \quad \text{(all } \|\cdot\| \text{ are 2-norms, unless otherwise stated)},$$

where R is n-by-n upper triangular. In our context, R is always the triangular factor in the QR factorization of an m-by-n matrix A, $m \geq n$; thus $A^TA = R^TR$. Hence, $\sigma_{\max}(R) = \sigma_{\max}(A)$ and $\sigma_{\min}(R) = \sigma_{\min}(A)$. We shall concentrate on estimating the two extreme singular values of A. This task, in turn, can be related to estimating the two extreme eigenvalues of A^TA or the extreme eigenvalues of AA^T (provided the $m - n$ zero eigenvalues are disregarded). It is for these estimations that we employ the Rayleigh–Ritz approximation.

Let the singular value decomposition of A be $A = U\Sigma V^T$, where

$$U = [u_1, u_1, \ldots, u_n], \quad V = [v_1, v_2, \ldots, v_n], \quad \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n),$$
$$\text{and} \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n.$$

Let $F = [x_1, x_2, \ldots, x_k]$ for some $k \leq n$ be a matrix of k orthonormal vectors; thus $F^TF = I$. The Rayleigh–Ritz approximation is obtained by calculating the eigensystem

$$(F^TAA^TF)W = W\operatorname{diag}(\tau_1^2, \tau_2^2, \ldots, \tau_k^2), \quad \tau_1^2 \geq \tau_2^2 \geq \cdots \geq \tau_k^2.$$

The Cauchy interlace theorem tells us that $\sigma_j^2 \geq \tau_j^2$, $j = 1, 2, \ldots, k$. The matrix $F^T(AA^T)F$ is a Rayleigh quotient of AA^T; the vectors $x_j = Fw_j$ are known as Ritz vectors, and the (τ_j^2, x_j)'s are known as Ritz pairs. If $\operatorname{span}(F) \approx \operatorname{span}(U^{(k)})$, $U^{(k)} = [u_1, u_2, \ldots, u_k]$, then the (τ_j, x_j)'s are reasonable estimates of (σ_j, u_j). In particular, we have a reasonable estimate of σ_1. Here we use the convenient notation $\operatorname{span}(\cdot)$ to denote the space spanned by the columns of the matrix inside the parentheses.

Similarly, we can estimate σ_n if we have a corresponding F approximating $[u_n, u_{n-1}, \ldots, u_{n-k+1}]$. The inequalities $\sigma_j^2 \leq \tau_j^2$, $j = n, n-1, \ldots, n-k+1$, would hold as long as $m = n$, or $\operatorname{span}(F)$ is orthorgonal to $\operatorname{span}(U)^\perp$. Analogous estimation can be performed by approximations of V and by the Rayleigh quotients of A^TA.

Various a priori bounds can be derived relating the quality of σ_j's estimates to the closeness of the subspaces $\operatorname{span}(F)$ and $\operatorname{span}(U^{(k)})$. These bounds are not useful in the quantitative sense, however, because we seldom know the closeness of the subspaces. Moreover, even in the case where $\operatorname{span}(F)$ is far from $\operatorname{span}(U^{(k)})$, good approximations to σ_j are still possible. For an extreme example, consider A with equal singular values $\sigma_1 = \sigma_2 = \cdots = \sigma_n$. On the other hand, the kind of analysis giving those bounds is sometimes useful in explaining why specific condition estimators are effective and in identifying situations where they may perform poorly. We shall carry out such analyses whenever they are appropriate. We refer the interested reader to Chapter 11 of [8] for a lucid discussion on Rayleigh–Ritz approximation.

3. Eigenproblems related to Rayleigh quotients. Consider $A = U\Sigma V^T$, $U = [u_1, u_2, \ldots, u_n]$, and

$$U^{(k)} = [u_1, u_2, \ldots, u_k] \quad \text{or} \quad U^{(k)} = [u_n, u_{n-1}, \ldots, u_{n-k+1}]$$

as before. Computations involved in determining the Ritz pairs differ depending on the information determining the space $\mathcal{F} \approx \operatorname{span}(U^{(k)})$. (The situation where we have $\mathcal{G} \approx \operatorname{span}(V^{(k)})$ is analogous, and its discussion is omitted in the rest of this section.)

3.1. Orthonormal basis. The orthornormal basis is the simplest case. $\mathcal{F}$ is given as span(F), where $F^TF = I$. The Ritz pairs are (τ_j^2, x_j), $x_j = Fw_j$, where (τ_j^2, w_j) are the eigenpairs of the Rayleigh quotient $F^T(AA^T)F$.

3.2. General basis. In some situations, $\mathcal{F}$ is given as span(F) but $F^TF \neq I$, although F has full rank. There are two approaches. One is to perform a QR factorization of F:

$$F = QR, \qquad Q \text{ is } n\text{-by-}k \quad \text{and} \quad Q^TQ = I.$$

The Ritz pairs are then (τ_j^2, x_j), where $x_j = Qw_j$ and the (τ_j^2, w_j) are the eigenpairs of $Q^T(AA^T)Q$.

At times, however, the matrix $F^T(AA^T)F$ is readily available. Thus,

$$Q^T(AA^T)Q = R^{-T}F^T(AA^T)FR^{-1}.$$

Then, the Ritz pairs are (τ_j^2, x_j), where $x_j = Fw_j$ and the (τ_j^2, w_j) are the eigenpairs of the generalized eigenvalue problem

$$F^T(AA^T)Fw = \tau^2(R^TR)w = \tau^2(F^TF)w.$$

3.3. General basis and a projection. In certain situations, we may know of an "undesirable" subspace $\mathcal{C}$ given by an orthonormal basis $C = [c_1, c_2, \ldots, c_\ell]$. For example, we may be estimating small singular values and span(C) = span(U)$^\perp$ (in particular $\ell = m-n$). It is quite natural, then, to use the space $\mathcal{F} = \text{span}(F)$ projected to $\mathcal{C}^\perp$. If we assume that $(I - CC^T)F$ has full rank, the Ritz pairs (according to the previous subsection) are obtained either from the eigenpairs of $Q^T(AA^T)Q$, where

$$(I - CC^T)F = QR, \qquad Q \text{ is } n\text{-by-}k \quad \text{and} \quad Q^TQ = I,$$

or from the eigenpairs of the generalized eigenvalue problem

$$\left[F^T(I - CC^T)(AA^T)(I - CC^T)F\right] w = \tau^2 \left[F^T(I - CC^T)F\right] w.$$

3.4. Krylov subspace. The Krylov subspace is the best-known subspace associated with Rayleigh–Ritz approximation. Here, the subspace is

$$\begin{aligned} \mathcal{F} &= \text{span}\{q_1, (AA^T)q_1, \ldots, (AA^T)^{k-1}q_1\}, \\ &= \text{span}\{q_1, q_2, \ldots, q_k\}, \qquad Q^TQ = I. \end{aligned}$$

Moreover, the tridiagonal Rayleigh quotient $T = Q^T(AA^T)Q$ is generated by the Lanczos tridiagonalization procedure. The Ritz pairs are obtained easily from T's eigenpairs.

4. Dynamic condition estimation. In practice, using the Rayleigh–Ritz method for condition estimation is especially attractive when the Ritz pairs can be computed economically. This is indeed often the case in the so-called dynamic condition estimation. A typical scenario is the following. At a certain stage, we have a matrix A and Ritz pairs (τ_j^2, x_j), such that

$$X^T(AA^T)X = \text{diag}(\tau_1^2, \tau_2^2, \ldots, \tau_k^2).$$

The τ_j's approximate A's largest or smallest singular values, or both. (It may also be the case that the Ritz pairs are (μ_j^2, y_j)'s where

$$Y^T(A^TA)Y = \operatorname{diag}(\mu_1^2, \mu_2^2, \ldots, \mu_k^2),$$

but we shall concentrate on AA^T, in general.)

At the next stage, which usually corresponds to new data coming in or a new iteration of some iterative process, we wish to obtain new Ritz pairs for a new matrix $\tilde{A}$. The situation in question usually suggests a convenient basis $\tilde{F} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{\tilde{k}}]$ ($\tilde{k}$ need not be k, and in fact $\tilde{k} = k+1$ frequently) such that the Ritz pairs for $\tilde{A}$ corresponding to the subspace $\tilde{\mathcal{F}} = \operatorname{span}(\tilde{F})$ can be computed economically, usually with the help of A's Ritz pairs. ($\tilde{F}$ is usually closely related to X.) Then, k appropriate Ritz pairs $(\tilde{\tau}_j^2, \tilde{x}_j)$ of $\tilde{A}$ with respect to $\tilde{\mathcal{F}}$ are chosen, and we have

$$\tilde{X}^T(\tilde{A}\tilde{A}^T)\tilde{X} = \operatorname{diag}(\tilde{\tau}_1^2, \tilde{\tau}_2^2, \ldots, \tilde{\tau}_k^2).$$

As pointed out earlier, depending on the nature of $\tilde{\mathcal{F}}$ and $\tilde{F}$, the computation may involve an eigenvalue problem or a generalized eigenvalue problem.

We now show that a number of different dynamic condition estimation schemes are in fact Rayleigh–Ritz methods. This view not only allows us to better understand these schemes, but also suggests generalizations or alternatives.

4.1. ICE. In his paper on ICE [1], Bischof considers the condition number of the growing upper triangular factor in a QR factorization. At a given stage, the subject is an n-by-n upper triangular matrix A. The estimate of A's smallest singular value is given by a large norm solution p to a unit norm right-hand side r, $p^TA = r^T$, $\|r\| = 1$. Thus, $\tau = \|p\|^{-1}$ is an estimate for $\sigma_{\min}(A)$. At the next stage, one more column of the triangular factor is generated:

$$\tilde{A} = \begin{bmatrix} A & a \\ 0^T & \alpha \end{bmatrix}.$$

To obtain a large norm solution $\tilde{p}$ to a new unit norm right-hand side for $\tilde{A}$, Bischof tries to

$$\text{maximize } \left\| [\, sp^T \quad \beta \,] \right\| \quad \text{subject to } \left\| [\, sp^T \quad \beta \,]\, \tilde{A} \right\| = 1.$$

Let $[sp^T\ \beta]\tilde{A} = [sp^TA\ \ c]$. The constraint becomes $s^2+c^2 = 1$. Algebraic manipulation gives

$$\left\| [\, sp^T \quad \beta \,] \right\|^2 = w^TMw, \qquad w^T = [\, s \quad c \,],$$

where M is a positive definite matrix given in terms of p^Ta, α, and τ. Consequently, the maximum is attained at M's largest eigenvalue, and $\tilde{p}^T = [\tilde{s}p^T\ \tilde{\beta}]$ can be computed from the corresponding eigenvector $[\tilde{s}\ \tilde{c}]^T$. The estimate for $\sigma_{\min}(\tilde{A})$ is given by $\tilde{p}$ and $\tilde{\tau} = \|\tilde{p}\|^{-1}$.

We now show that the above process is really a Rayleigh–Ritz method. The estimation of $\sigma_{\min}(A)$ is given by the Ritz pair (τ^2, x), where $x = p/\|p\|$, of the Rayleigh quotient $x^T(AA^T)x$. To estimate $\sigma_{\min}(\tilde{A})$, we simply use the subspace

$$\tilde{\mathcal{F}} = \operatorname{span}(\tilde{F}), \qquad \tilde{F} = \begin{bmatrix} x & 0 \\ 0 & 1 \end{bmatrix}.$$

Clearly, $\tilde{F}^T\tilde{F} = I$, and the two Ritz pairs are $(\tilde{\tau}_j^2, \tilde{x}_j = \tilde{F}\tilde{w}_j)$, where $(\tilde{\tau}_j^2, \tilde{w}_j)$ are the eigenpairs of

$$\tilde{F}^T(\tilde{A}\tilde{A}^T)\tilde{F} = \operatorname{diag}(\tau^2, 0) + zz^T, \qquad z = \begin{bmatrix} x^T a \\ \alpha \end{bmatrix}.$$

The new estimate $(\tilde{\tau}_1, \tilde{x}_1)$ is derived from the small Ritz pair $(\tilde{\tau}_1^2, \tilde{x}_1)$.

To show that ICE is equivalent to the Rayleigh–Ritz method just described, we first show that $\tilde{\tau}_1 = \tilde{\tau}$:

$$\begin{aligned} 1 &= \left\|\tilde{w}_1^T\tilde{F}^T\tilde{A}\right\| \cdot \tilde{\tau}_1^{-1}, \quad \tilde{w}_1^T = [\gamma_1 \quad \gamma_2], \quad \text{and} \quad \|\tilde{q}_1\| = 1 \\ &= \left\|[\gamma_1 p^T/\|p\| \quad \gamma_2]\,\tilde{A}\right\| \cdot \tilde{\tau}_1^{-1}. \end{aligned}$$

Thus,

$$1 = \left\|[(\gamma_1/\tilde{\tau}_1\|p\|)p^T \quad (\gamma_2/\tilde{\tau}_1)]\,\tilde{A}\right\|,$$

giving

$$\left\|[(\gamma_1/\tilde{\tau}_1\|p\|)p^T \quad (\gamma_2/\tilde{\tau}_1)]\right\| \le \tilde{\tau}^{-1}.$$

Consequently, $\tilde{\tau}_1 \ge \tilde{\tau}$. On the other hand,

$$\begin{aligned} 1 &= \left\|\tilde{p}^T\tilde{A}\right\| \\ &= \left\|[\tilde{s}\|p\| \quad \tilde{\beta}]\,\tilde{F}^T\tilde{A}\right\| \end{aligned}$$

implies $\tilde{\tau} = \|\tilde{w}^T\tilde{F}^T\tilde{A}\|$, $\tilde{w}^T = [\tilde{s}\|p\| \ \tilde{\beta}]^T$ normalized. Thus $\tilde{\tau} \ge \tilde{\tau}_1$. Thus, we have $\tilde{\tau} = \tilde{\tau}_1$. We omit the proof that $\tilde{x}_1 = \tilde{p}/\|\tilde{p}\|$, which is straightforward. Similar arguments show that the estimation of $\sigma_{\max}(\tilde{A})$ in [1] is also a Rayleigh–Ritz method with an initial Ritz pair (τ^2, x) where $\tau^2 = x^T(AA^T)x \approx \sigma_{\max}^2(A)$. The Ritz pair for $\tilde{A}$ is obtained by using the subspace

$$\tilde{\mathcal{F}} = \operatorname{span}(\tilde{F}), \qquad \tilde{F} = \begin{bmatrix} x & 0 \\ 0 & 1 \end{bmatrix}.$$

Indeed, viewing ICE as a Rayleigh–Ritz method leads to convenient generalizations (some of which were explored in [2]). At a given stage, we have Ritz pairs (τ_j^2, x_j) such that

$$X^T(AA^T)X = \operatorname{diag}(\tau_1^2, \tau_2^2, \ldots, \tau_k^2), \quad \tau_1 \ge \tau_2 \ge \cdots \ge \tau_k \ge 0,$$

where $\tau_1 \approx \sigma_{\max}(A)$, $\tau_k \approx \sigma_{\min}(A)$, and the interlacing property

$$\sigma_{\max-j+1}(A) \ge \tau_j \ge \sigma_{\min+k-j}(A), \qquad j = 1, 2, \ldots, k$$

holds. At the next stage, we have

$$\tilde{A} = \begin{bmatrix} A & a \\ 0^T & \alpha \end{bmatrix}.$$

The subspace is chosen to be the $\tilde{k} = k + 1$ dimensional

$$\tilde{\mathcal{F}} = \operatorname{span}(\tilde{F}), \qquad \tilde{F} = \begin{bmatrix} X & 0 \\ 0^T & 1 \end{bmatrix}.$$

Clearly, $\tilde{F}^T\tilde{F} = I$, and the problem becomes that of determining the eigensystem of the Rayleigh quotient

$$\tilde{F}^T(\tilde{A}\tilde{A}^T)\tilde{F} = \operatorname{diag}(\tau_1^2, \tau_2^2, \ldots, \tau_k^2, 0) + zz^T, \qquad z = \begin{bmatrix} X^T a \\ \alpha \end{bmatrix}.$$

Thus, both the Rayleigh quotient and its eigensystem can be determined economically (see [4] and [11]). Note that if we were to formulate this generalization following the approach in [1] as a maximization problem in terms of $k+1$ parameters and try to realize the objective function as a quadratic form, the algebraic manipulation would have been quite involved.

Understanding ICE as a Rayleigh–Ritz method sheds some light on its effectiveness as well as its ineffectiveness. The analysis usually assumes that some of the extreme left singular vectors of A (or, equivalently, the extreme eigenvectors of AA^T) are close to $\mathcal{F} = \operatorname{span}(X)$; thus $\angle(u_j, \mathcal{F}) = \theta_j$ is small for $j = 1, 2, \ldots, \ell$ or $j = n, n-1, \ldots, n-\ell+1$, or both, for some ℓ. We then try to identify situations where $\angle(\tilde{u}_j, \tilde{\mathcal{F}})$ will also be small, where

$$\tilde{A} = \tilde{U}\tilde{\Sigma}\tilde{V} \quad \text{and} \quad \tilde{A}\tilde{A}^T = \tilde{U}\tilde{\Sigma}^2\tilde{U}^T.$$

Since the nature of these analyses is more qualitative than quantitative, we shall not call the results theorems or lemmas.

PHENOMENON 1. Suppose $\sigma_{\min}(A) = \sigma_n$ is not small (say, $\sigma_n \geq 1$) and that there is a drastic drop of the smallest singular value of $\tilde{A}$ (say, $\sigma_{\min}(\tilde{A}) = \tilde{\sigma}_{n+1} \leq \sqrt{\varepsilon}$, where ε is the machine precision). Furthermore, assume that the vector a has a moderate norm (say, $\|a\| \leq 2$). Then, ICE is effective in tracking $\tilde{\sigma}_{n+1}$.

Explanation. Let M be the matrix

$$M = \begin{bmatrix} AA^T & \\ & 0 \end{bmatrix}.$$

Then,

$$\tilde{A}\tilde{A}^T = M + \tilde{a}\tilde{a}^T, \qquad \tilde{a} = \begin{bmatrix} a \\ \alpha \end{bmatrix}.$$

Let U be the left singular vectors of A. Therefore, the eigenvalues of $\tilde{A}\tilde{A}^T$ are the eigenvalues of

$$\begin{bmatrix} U & 0 \\ 0^T & 1 \end{bmatrix}^T (M + \tilde{a}\tilde{a}^T) \begin{bmatrix} U & 0 \\ 0^T & 1 \end{bmatrix}$$
$$= \operatorname{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2, 0) + bb^T, \qquad b^T = [\beta_1 \beta_2 \cdots \beta_{n+1}] = [a^T U \ \alpha].$$

Thus, $\tilde{\sigma}_{n+1}$ satisfies the equation (see [7])

$$\begin{aligned} \frac{\alpha^2}{\tilde{\sigma}_{n+1}^2} &= 1 + \sum_{j=1}^{n} \frac{\beta_j^2}{\sigma_j^2 - \tilde{\sigma}_{n+1}^2} \\ &\leq 1 + \sum_{j=1}^{n} \frac{\beta_j^2}{1 - \tilde{\sigma}_{n+1}^2} \\ &\leq \left(1 + \frac{\|a\|^2}{1 - \tilde{\sigma}_{n+1}^2}\right) \tilde{\sigma}_{n+1}^2 \\ &\leq 4\tilde{\sigma}_{n+1}^2. \end{aligned}$$

Now, since ICE applies Rayleigh–Ritz method to $\tilde{A}\tilde{A}^T$ with span($\tilde{F}$) that contains $[0\ 1]^T$, we have

$$\begin{aligned}\tilde{\tau}_{\min}^2 &\le [0\ 1]^T \tilde{A}\tilde{A}^T \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &\le \alpha^2 \\ &\le 4\tilde{\sigma}_{n+1}^2.\end{aligned}$$

Moreover, we may even say something about the direction of the corresponding left singular vector by invoking well-known results in [5] on rotation of eigenvectors by perturbation. Now,

$$[0\ 1]^T M \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0,$$

and the residual

$$(M + \tilde{a}\tilde{a}^T) \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot 0 = \alpha\tilde{a}$$

has norm less than $5\sqrt{\varepsilon}$. Furthermore, $M + \tilde{a}\tilde{a}^T$ has a large separation between the smallest and the next smallest eigenvalue. We have, therefore, by the "sin θ" theorem in [5]

$$\sin\theta \le 5\sqrt{\varepsilon}/(1-\varepsilon),$$

where θ is the angle between $\tilde{A}\tilde{A}^T$'s smallest eigenvector and $[0\ 1]^T$.

Thus ICE is effective in tracking $\tilde{\sigma}_{\min}$ whenever there is a sudden big drop of the smallest singular value from A to $\tilde{A}$, *regardless* of the quality of the estimate of $\sigma_{\min}(A)$. This phenomenon can be generalized.

PHENOMENON 2. Suppose $\sigma_{\min-\ell}(A) = \sigma_{n-\ell} \ge 1$; $\sigma_j \le \sqrt{\varepsilon}$ for $j > n-\ell$ for some $\ell < k$, the dimension of the space $\mathcal{F}$. Let $\angle(u_j, \mathcal{F}) \le \theta$, $j = n, n-1, \ldots, n-\ell+1$ for some small θ (say, $\theta \le \sqrt{\varepsilon}$). If $\tilde{\sigma}_{n-\ell+1}$ is also small (say, $\tilde{\sigma}_{n-\ell+1} \le \sqrt{\varepsilon}$), ICE is effective in tracking $\tilde{\sigma}_{n+1}, \ldots, \tilde{\sigma}_{n-\ell+1}$ and the corresponding singular vectors $\tilde{u}_{n+1}, \ldots, \tilde{u}_{n-\ell+1}$.

The analysis is similar to the preceding one, and the crux is to show that $\sum_{j=n-\ell+1}^{n} \beta_j^2$ is small where $b^T = [U^T a\ \alpha]$. Note that the direction of the vector a, where

$$\tilde{A}\tilde{A}^T = \begin{bmatrix} AA^T & \\ & 0 \end{bmatrix} + \begin{bmatrix} a \\ \alpha \end{bmatrix} [a^T \quad \alpha],$$

does not play any direct role in the above phenomena. Moreover, the analyses so far indicate that ICE is reasonably effective *in general* in tracking small singular values when a large gap separates them from the large singular values. But we must emphasize the phrase "in general" because our analyses assume that k is large enough to cover all the values on the left of the gap, and that a gap in the singular values of a triangular matrix does not necessarily give existence of similar well-defined gaps in the singular value spectra of its principal submatrices.

The analysis above also suggests that the problem is likely to arise when $\|a\|$ is disproportionately large. This observation is confirmed in Experiment 1 in §5.

What about the effectiveness of ICE in estimating large singular values? Since the Rayleigh–Ritz method is known to be effective in tracking large eigenvalues, ICE

is generally effective in tracking large singular values. This fact is illustrated by the many experiments in [1]. We must, however, be aware that the Rayleigh–Ritz method need not give a good approximation to the corresponding large eigenvectors. Consider, for example, a matrix AA^T with one small eigenvalue close to $\sqrt{\varepsilon}$, while all the rest of the eigenvalues are large, but all within a factor of four to each other. Then any space $\mathcal{F}$ with dimension two or more will give a reasonable estimate to the maximum eigenvalue, but the corresponding Ritz vector may be even orthogonal to the exact largest eigenvector. Along this line, we also point out an obviously bad situation for ICE in estimating large singular values. Suppose $\dim(\mathcal{F}) = k$ is close to the k largest eigenvectors of AA^T. Furthermore, $\sigma_{\max}(A) \approx 1$. Now, if we have

$$\tilde{A} = \begin{bmatrix} A & a \\ 0^T & \alpha \end{bmatrix}$$

as usual, but $\angle(a, \mathrm{span}([u_1 u_2 \cdots u_k])) \approx 90°$, u_j's being A's exact singular vectors, $\|a\| \approx 10^5$, and $|\alpha| \approx 1$, then ICE will estimate $\sigma_{\max}(\tilde{A})$ poorly. One can argue, of course, that this is a rare case from a probablistic point of view.

Before discussing the next condition estimation scheme, we mention that the simple choice of α^2 as an (upper) estimate of $\lambda_{\min}(\tilde{A}\tilde{A}^T) = \sigma^2_{\min}(\tilde{A})$ is also a Rayleigh–Ritz method. The subspace is simply $\mathrm{span}([0\ 1]^T)$. This is the estimation used implicitly in the QR factorization with column pivoting. In particular, because the subspace used in ICE always contains $[0\ 1]^T$, the estimate of ICE is always better than α^2.

4.2. ACE. ACE was designed by Pierce and Plemmons [9] to track the condition number of the information matrix (or, equivalently, that of the covariance matrix) in recursive least-squares problems using a forgetting factor. At a certain stage, one has an n-by-n upper triangular matrix A, which is the Cholesky factor of the information matrix. An estimate of $\sigma_{\min}(A)$ is given by a unit norm vector x such that $\|x^T A\| \approx \sigma_{\min}(A)$. At the next stage, a row is added to A:

$$\tilde{A} = \begin{bmatrix} A \\ a^T \end{bmatrix} \quad \text{and} \quad Q \begin{bmatrix} A \\ a^T \end{bmatrix} = \begin{bmatrix} \tilde{R} \\ 0^T \end{bmatrix},$$

where $\tilde{R}$ is the Cholesky factor for $\tilde{A}^T\tilde{A}$. The goal is to estimate $\sigma_{\min}(\tilde{A})$, and the means is by minimizing $\|z^T\tilde{R}\|/\|z\|$ over some convenient choices of z. The choice used in [9] is

$$z = \text{first } n \text{ component of } Q \begin{bmatrix} \alpha x \\ \beta \end{bmatrix}.$$

To solve this minimization problem requires some algebra. Since the last row of $Q\tilde{A}$ is zero, it is obvious that the last row of Q must be $q^T = [-c^T\gamma\ \gamma]$, where $\gamma = (1 + c^T c)^{-1/2}$ and $c^T A = a^T$. (Computing c requires $O(n)$ operations and is relatively inexpensive.) Because Q is orthogonal and $\|x\| = 1$, we have

$$\left\| Q \begin{bmatrix} \alpha x \\ \beta \end{bmatrix} \right\|^2 = \alpha^2 + \beta^2.$$

Thus,

$$\|z\|^2 = \alpha^2 + \beta^2 - \left\| q^T \begin{bmatrix} \alpha x \\ \beta \end{bmatrix} \right\|^2$$

$$= [\alpha \quad \beta] \begin{bmatrix} 1-\gamma^2(c^Tx)^2 & \gamma^2 c^Tx \\ \gamma^2 c^Tx & 1-\gamma^2 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$
$$= [\alpha \quad \beta] N \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Finally,

$$\|z^T\tilde{R}\|^2 = \left\| [\alpha x^T \quad \beta] Q^T \begin{bmatrix} \tilde{R} \\ 0^T \end{bmatrix} \right\|^2$$
$$= \left\| [\alpha x^T \quad \beta] \begin{bmatrix} A \\ a^T \end{bmatrix} \right\|^2$$
$$= [\alpha \quad \beta] \begin{bmatrix} b^Tb & b^Ta \\ a^Tb & a^Tb \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \qquad b = Ax,$$
$$= [\alpha \quad \beta] M \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Thus, the minimization problem is equivalent to

$$\text{minimize}_{\alpha,\beta} \frac{[\alpha \quad \beta] M \begin{bmatrix} \alpha \\ \beta \end{bmatrix}}{[\alpha \quad \beta] N \begin{bmatrix} \alpha \\ \beta \end{bmatrix}},$$

which is the minimum eigenpair of the generalized eigenvalue problem $Mw = \lambda Nw$. (Note that N is positive definite whenever a is not a multiple of b, cf. [9].) To estimate $\sigma_{\max}(\tilde{A})$, one simply starts with x such that $\|x^TA\| \approx \sigma_{\max}(A)$ and carries out the algebra. The result is the problem of determining the maximum eigenpair of $Mw = \lambda Nw$.

We now show that this method is also a Rayleigh–Ritz method. First, consider the estimation of $\sigma_{\min}(\tilde{A})$. A convenient space for $\tilde{A}\tilde{A}^T$ is

$$\tilde{\mathcal{F}} = \text{span}(\tilde{F}), \qquad \tilde{F} = \begin{bmatrix} x & 0 \\ 0 & 1 \end{bmatrix}.$$

The $(n+1)$th eigenvalue $\lambda_{n+1}(\tilde{A}\tilde{A}^T) = 0$ may cause a gross underestimate of $\sigma_n(\tilde{A})$. An obvious remedy is to further project $\tilde{\mathcal{F}}$ onto the orthogonal complement of the eigenspace of $\lambda_{n+1}(\tilde{A}\tilde{A}^T)$. Recall the vector q, $q^T = [-c^T\gamma \ \ \gamma]$ in the preceding discussion, where $q^T\tilde{A} = 0$. Clearly,

$$\tilde{A}\tilde{A}^Tq = 0 = \lambda_{n+1}(\tilde{A}\tilde{A}^T)q,$$

and the desired subspace is thus $\tilde{\mathcal{F}}$ projected onto $\text{span}(q)^\perp$. From §3.3, this results in the generalized eigenvalue problem

$$\tilde{F}^T(I-qq^T)(\tilde{A}\tilde{A}^T)(I-qq^T)\tilde{F}w = \lambda\tilde{F}(I-qq^T)\tilde{F}w.$$

But $q^T\tilde{A} = 0$, and we have

$$\tilde{F}^T(I-qq^T)(\tilde{A}\tilde{A}^T)(I-qq^T)\tilde{F} = \tilde{F}^T(\tilde{A}\tilde{A}^T)\tilde{F}$$
$$= \begin{bmatrix} b^Tb & b^Ta \\ a^Tb & a^Ta \end{bmatrix}$$

and

$$\tilde{F}^T(I - qq^T)\tilde{F} = I - (\tilde{F}^T q)(q^T \tilde{F}) = \begin{bmatrix} 1 - \gamma^2(c^T x)^2 & \gamma^2 c^T x \\ \gamma^2 c^T x & 1 - \gamma^2 \end{bmatrix}.$$

The derivation for estimating $\sigma_{\max}(\tilde{A})$ is exactly the same.

Using the Rayleigh–Ritz framework, we can easily combine the estimations of largest and smallest singular values and also generalize them. At any given stage, we have k Ritz pairs (τ_j^2, x_j), $j = 1, 2, \ldots, k$, such that

$$X^T(AA^T)X = B^T B = \operatorname{diag}(\tau_1^2, \tau_2^2, \ldots, \tau_k^2),$$

where some of the τ_j approximates A's large singular values, while the others approximate A's small singular values. At the next stage, we have

$$\tilde{A} = \begin{bmatrix} A \\ a^T \end{bmatrix},$$

and we use the subspace $\tilde{\mathcal{F}}$ projected to $\operatorname{span}(q)^\perp$, where

$$\tilde{\mathcal{F}} = \operatorname{span}(\tilde{F}), \qquad \tilde{F} = \begin{bmatrix} X & 0 \\ 0^T & 1 \end{bmatrix}.$$

The corresponding generalized eigenvalue problem is $Mw = \lambda N w$, where

$$M = \begin{bmatrix} \operatorname{diag}(\tau_1^2, \tau_2^2, \ldots, \tau_k^2) & B^T a \\ a^T B & a^T a \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} I - \gamma^2(X^T c)(c^T X) & \gamma^2 X^T c \\ \gamma^2 c^T X & 1 - \gamma^2 \end{bmatrix}.$$

The problem is that, unlike the generalization of ICE, which leads to an eigenproblem of rank-1 perturbed diagonal matrices, computational schemes that can exploit the structure of the matrix pencil here seem to be lacking.

There is, however, a rather natural alternative in view of the Rayleigh–Ritz approximation. If we use the space $\mathcal{H} = \operatorname{span}[\tilde{F}\ q]$, we will have $(0, q)$ as the smallest Ritz pair—which can be discarded because it corresponds to the spurious zero singular value of $\tilde{A}$. Moreover, because the remaining Ritz vectors are orthogonal to q, they will be reduced back to n-vectors from $(n+1)$-vectors after a transformation by Q, where

$$Q\tilde{A} = \begin{bmatrix} R \\ 0^T \end{bmatrix}.$$

Let us now consider the calculation of these Ritz pairs. A sequence of $k+2$ Householder transformations can be used to determine a vector g such that $[\tilde{F}\ g]$ forms an orthonormal basis of $\mathcal{H}$. This requires only order $n+1$ amount of work. The Rayleigh quotient is thus

$$\begin{bmatrix} \tilde{F}^T \\ g^T \end{bmatrix} (\tilde{A}\tilde{A}^T) [\tilde{F} \quad g] = \begin{bmatrix} \tilde{F}^T(\tilde{A}\tilde{A}^T)\tilde{F} & \tilde{F}^T(\tilde{A}\tilde{A}^T)g \\ g^T(\tilde{A}\tilde{A}^T)\tilde{F} & g^T(\tilde{A}\tilde{A}^T)g \end{bmatrix}.$$

The eigensystem can be determined by an efficient (order k^2 work) rank-1 update method as follows. First,

$$\tilde{F}^T(\tilde{A}\tilde{A}^T)\tilde{F} = \begin{bmatrix} \operatorname{diag}(\tau_1^2, \tau_2^2, \ldots, \tau_k^2) & B^T a \\ a^T B & a^T a \end{bmatrix}.$$

Let $\Sigma = \operatorname{diag}(\tau_1, \tau_2, \ldots, \tau_k)$ and let $z = \Sigma^{-1} B^T a$. Since $\Sigma^{-1} B^T B \Sigma^{-1} = I_k$, z is an othorgonal projection of a to a k-dimension subspace. Thus, $a^T a \geq z^T z$ and we can define $\alpha^2 = a^T a - z^T z$. Then,

$$\tilde{F}^T (\tilde{A}\tilde{A}^T)\tilde{F} = \begin{bmatrix} \Sigma^2 & \Sigma z \\ z^T \Sigma & z^T z \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \end{bmatrix} \cdot [0^T \quad \alpha].$$

The eigensystem of the first matrix on the right can be solved efficiently by the method given in [3]. In particular, when all the τ_j's are distinct, the $k+1$ eigenvalues λ_j are zero and the k roots of a secular equation

$$1 + \sum_{j=1}^{k} \frac{\zeta_j^2}{\tau_j^2 - \lambda}, \quad \text{where } [\zeta_1 \zeta_2 \cdots \zeta_k] = z^T.$$

The eigenvectors are given by

$$\begin{bmatrix} \left(\Sigma^2 - \lambda_j I\right)^{-1} \Sigma z \\ -1 \end{bmatrix}.$$

The case of $\tau_j = \tau_{j+1}$ for some j can be handled easily by deflation techniques described in [3]. Once these quantities are computed, the eigensystem of $\tilde{F}^T (\tilde{A}\tilde{A}^T)\tilde{F} = U\tilde{\Sigma}^2 U^T$ can be determined by applying another round of rank-1 update technique. Finally,

$$\begin{bmatrix} \tilde{F}^T \\ g^T \end{bmatrix} (\tilde{A}\tilde{A}^T) [\tilde{F} \quad g] = \begin{bmatrix} U & \\ & 1 \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}^2 & \tilde{\Sigma}\tilde{z} \\ \tilde{z}^T \tilde{\Sigma} & \tilde{z}^T \tilde{z} \end{bmatrix} \begin{bmatrix} U^T & \\ & 1 \end{bmatrix},$$

where $\tilde{z} = U^T g$. The rank-1 update method in the preceding discussions is clearly applicable here, too.

We see two advantages of this alternative that do not require solving a generalized eigenvalue problem. First, exploiting the structure of the resulting eigenvalue problem as suggested above leads to an order k^2 computation, as opposed to an order k^3 computation for the generalized eigenvalue problem. This advantage is likely to be minor since k is usually very small. The second advantage, however, is more significant. The original scheme that requires solving $Mw = \lambda N w$ is quite complicated in its additional processing when N is ill conditioned. In fact, it is unclear if this processing can be generalized when N has dimension higher than two. Our alternative here, however, is free from such complications. The reason is as follows. Even under conditions that would lead to an ill-conditioned N, our computation that tries to find an orthonormal basis for span$([\tilde{F}, q])$ would still produce a set of orthonormal vectors. This set is all that is required for our Rayleigh–Ritz method to be valid numerically. Thus, the proposed alternative does not have numerical breakdown.

4.3. ALE. ALE is a condition estimation scheme proposed recently by Ferng, Golub, and Plemmons [6]. At a given stage, the subject is an upper triangular matrix A (the Cholesky factor of a covariance matrix). Estimates of A's largest singular values are given by approximate left and right singular vectors $x_1, x_2, \ldots, x_k$ and $y_1, y_2, \ldots, y_k$, respectively, satisfying

$$X^T A Y = \operatorname{diag}(\tau_1, \tau_2, \ldots, \tau_k), \quad \tau_1 \geq \tau_2 \cdots \geq \tau_k,$$

where

$$X = [x_1 x_2 \cdots x_k], \quad Y = [y_1 y_2 \cdots y_k], \quad X^T X = Y^T Y = I.$$

Estimates on A's smallest singular values are given by A^{-1}'s approximate largest singular vectors satisfying analogous conditions.

At the next stage, $\tilde{A}$ is the Cholesky factorization of A^TA+aa^T or A^TA-aa^T. To obtain new estimates of $\tilde{A}'s$ largest singular values, we apply k steps of the Lanczos algorithm to the matrix B with the starting vector q_1:

$$B = \begin{bmatrix} & \tilde{A} \\ \tilde{A}^T & \end{bmatrix}, \qquad q_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}.$$

It is therefore abundantly clear that ALE is a Rayleigh–Ritz method using the Krylov subspace $\text{span}\{q_1, Bq_1, \ldots, B^{k-1}q_1\}$ given by an orthonormal basis. Because the Krylov subspaces are used, this method is effective only in estimating B's extreme eigenvalues. Because B's eigenvalues are $\{\pm\sigma_1(\tilde{A}), \pm\sigma_2(\tilde{A}), \ldots, \pm\sigma_n(\tilde{A})\}$, this method on $\tilde{A}$ is effective only in estimating $\tilde{A}$'s largest singular values. To estimate the smallest singular values, Ferng, Golub, and Plemmons apply the Lanczos method to $\tilde{A}^{-1}$ (using a back-solve involving $\tilde{A}$ instead of $\tilde{A}^{-1}$ explicitly).

The generalization of this scheme is obvious. In fact, the discussion in [6] is for a general k, although its implementation restricts k to two so that eigenproblems of the Rayleigh quotients are reduced to finding roots of quadratic equations.

The advantages of this method are several. Many well-established a priori or a posterori bounds can be obtained (see [8] for an excellent discussion on the Lanczos algorithm) if one cares to pay the price for them. Since estimation of small singular values is achieved by estimating the large singular values of $\tilde{A}^{-1}$, this method generally yields the best estimate compared to other condition estimation schemes discussed so far. Furthermore, by suitably applying a posterori bounds on the large and small singular values, rather reliable upper and lower bounds on $\tilde{A}$'s condition number can be obtained.

The main disadvantage of this method is obviously its cost: while the preceding schemes require order n amount of work, ALE requires order n^2. Judging the price one must pay for this estimation, it seems natural that one should increase the value of k to at least three or four. The resulting eigenproblem can be readily solved by standard QR iteration, or using rank-1 update techniques as the (implicit) tridiagonal Rayleigh quotient (or the explicit bidiagonal) is being built column by column.

4.4. GRACE. GRACE is developed by Shroff and Bischof in [10] to estimate the condition of $QR + wv^T$. The method essentially combines two ICE steps and one ACE step. Consequently, we can also use our Rayleigh–Ritz framework to understand GRACE. It is worthwhile, however, to investigate if one Rayleigh–Ritz step can be applied to obtain condition estimate for $QR + wv^T$.

5. Numerical experiments. We present a number of experiments that illustrate the preceding discussions.

Experiment 1. This experiment relates to the direction and scaling of the added column a in ICE. Here, ICE is applied to a sequence of updates: $A_1 \in \mathbf{R}$,

$$\tilde{A}_\ell := \begin{bmatrix} A_\ell & a_\ell \\ 0^T & \alpha_\ell \end{bmatrix}; \quad A_{\ell+1} := \tilde{A}_\ell, \quad \ell = 1, 2, \ldots, n-1.$$

We noted previously that as long as $\|a\|$ remains moderate compared to $\|A\|$, the directions of those vectors do not, in general, affect the quality of the ICE estimates. But if $\|a\|$ is large, it is crucial that the subspace used in the approximation not be far away from $\|a\|$. To illustrate this point, we do the following. A_n, $n = 30$, is an

upper triangular matrix generated by applying a QR factorization to $U\Sigma V^T$, where U and V are random orthogonal matrices generated by the method described in [12], and Σ is a diagonal matrix with entries chosen randomly in $[0, 1]$.

We apply ICE with $k = 2$ (that is, one vector to estimate $\sigma_{\max}$ and one vector to estimate $\sigma_{\min}$). After the estimate is obtained for A_{n-1}, we modify a_n by

$$a_n := \text{normalize}\left((I - 0.999XX^T)a_n\right),$$

where $X = [x_1 x_2]$ are the two approximate vectors generated by ICE to estimate A's extreme singular values. We denote the resulting matrix B. This way, the last added column is nearly orthogonal to the subspace to be used in B's estimate. Because $\|a_n\| = 1$ is moderate, we do not expect great degradation in the ICE estimate. Indeed,

$$\sigma_{\max}(B)/\tau_{\max} = 1.3 \quad \text{and} \quad \tau_{\min}/\sigma_{\min}(B) = 2.6.$$

Next, we scale a_n up by a factor of 10^3. We call the resulting matrix $B^{(s)}$:

$$B^{(s)} = \begin{bmatrix} A_{n-1} & 10^3 a_n \\ 0^T & \alpha_n \end{bmatrix}.$$

The result of ICE is drastically changed:

$$\sigma_{\max}(B^{(s)})/\tau_{\max} = 1100 \quad \text{and} \quad \tau_{\min}/\sigma_{\min}(B^{(s)}) = 480.$$

Finally, we increase the dimension of the subspace used by two; that is, we use two vectors to estimate the maximum singular values and two vectors to estimate the minimum singular values. The quality of the ICE estimates are restored:

$$\sigma_{\max}(B^{(s)})/\tau_{\max} = 3.4 \quad \text{and} \quad \tau_{\min}/\sigma_{\min}(B^{(s)}) = 1.3.$$

Experiment 2. We apply ACE and our alternative to track the condition numbers of a sequence of Cholesky factors produced by recursive least-squares computations employing an exponential window. The process is initiated by a 30-by-30 identity matrix, and the forgetting factor used is 0.95. Three condition estimators are used. The first one is ACE. The second one is our alternative with $k = 2$: one vector for $\sigma_{\max}$ and one vector for $\sigma_{\min}$. The third one is our alternative with $k = 3$: one vector for $\sigma_{\max}$, but two vectors for $\sigma_{\min}$.

We test two cases. In the first test case, 60 steps of updates are performed where the 60 rows of data are numbers chosen randomly in $[-1, 1]$ with outliers of magnitude 10^4 added to six random places. In the second test case, 30 steps of updates are performed where 30 rows of data are $s_j a_j^T$, where a_j's entries are randomly chosen from $[-1, 1]$ and $s_j = 10^{4j/30}$. One can consider the incoming data as exponentially scaled. The results are summarized in Fig. 1 by the ratios of

$$\frac{\text{actual condition number}}{\text{estimated condition number}}.$$

The experiment illustrates that our alternative is comparable to ACE. Moreover, by increasing k—which is made practical by not having to solve a generalized eigenvalue problem—the estimates can become more reliable.

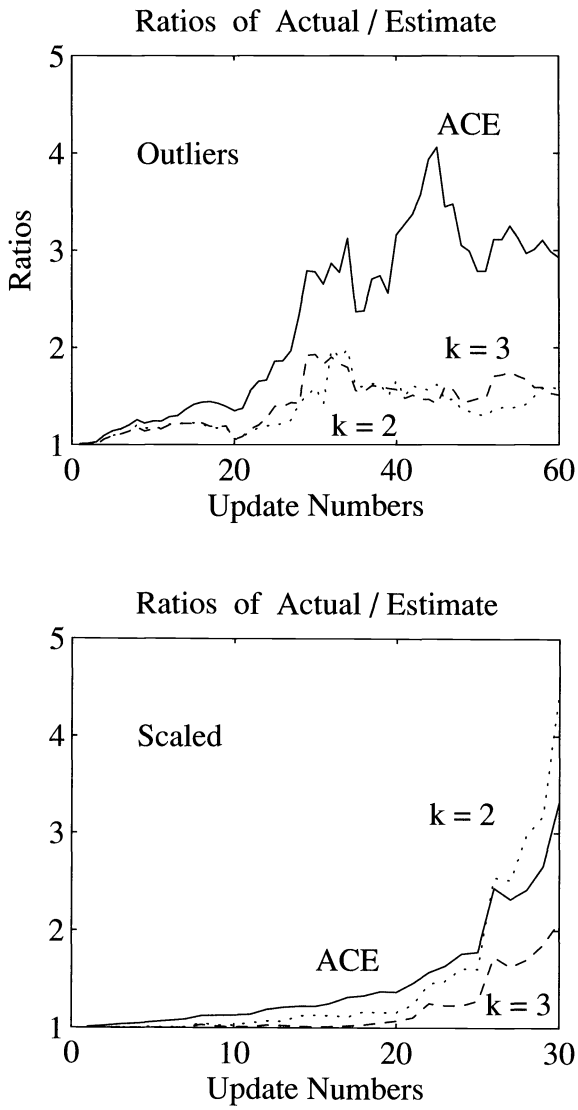


FIG. 1. ACE *on random matrices with outliers added or exponentially scaled.*

Experiment 3. As pointed out earlier, ALE is generally accurate. This experiment illustrates the benefit of increasing the number of Lanczos iterations k (which is the dimension of the Krylov subspace). Here, ALE is applied to estimate the condition numbers of a sequence of Cholesky factors generated by recursive least-squares computations employing a sliding window. Hence, both up and downdatings are involved. The process is initiated by a 20-by-20 identity matrix, and the up and downdatings are performed on 20 rows of data composed of numbers chosen randomly in $[-1, 1]$ with outliers of magnitude 10^4 added to six random places. The results are summarized in

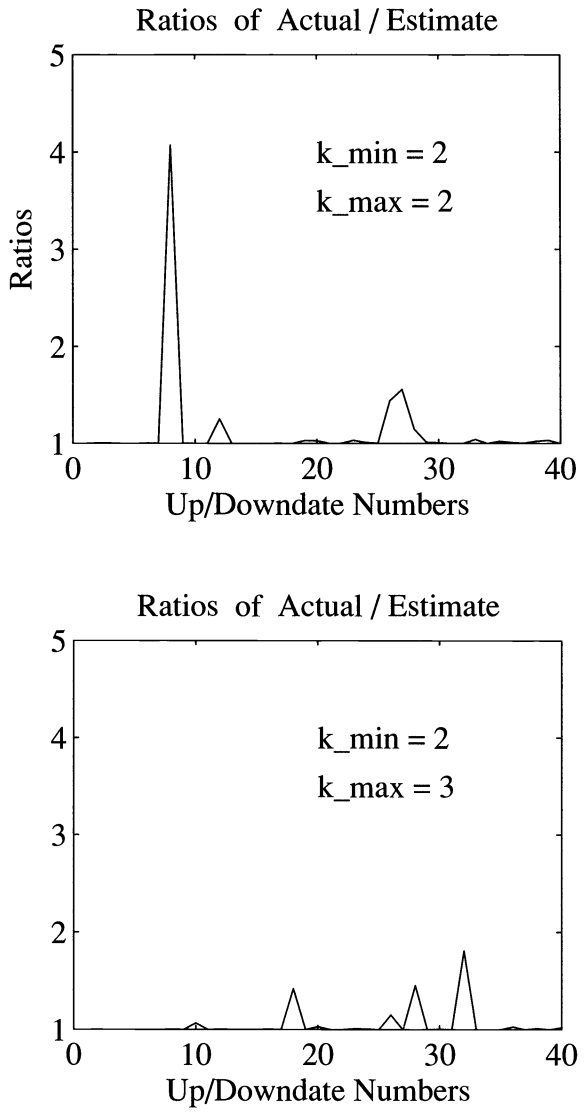


FIG. 2. ALE *on random matrices with outliers added.*

Fig. 2 by the ratios of

$$\frac{\text{actual condition number}}{\text{estimated condition number}}.$$

6. Conclusions. We have shown that the Rayleigh–Ritz approximation is applicable to dynamic condition estimations. Viewing many existing dynamic condition estimators as Rayleigh–Ritz approximation allows better understanding of, and natural generalization to, such estimators. Moreover, this viewpoint is likely to lead to other condition estimators tailored to other specific situations.

Acknowledgments. This work was motivated by a discussion with Professor Robert Plemmons during a matrix theory conference held at the University of Hong Kong in the Summer of 1991. The author also acknowledges the hospitality of the organizers of that conference, Drs. Yik-Hoi Au-Yeung and Raymond Chan. The source codes of ACE and ALE are provided by Drs. Daniel Pierce and William Ferng, respectively. The author has also benefitted from discussions with Drs. Christian Bischof and Daniel Pierce.

REFERENCES

[1] C. H. BISCHOF, *Incremental condition estimation*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 312–322.

[2] C. H. BISCHOF AND P. T. P. TANG, *Generalizing incremental condition estimation*, J. Numer. Linear Algebra, to appear.

[3] J. R. BUNCH AND C. R. NIELSEN, *Updating the singular value decomposition*, Numer. Math., 31 (1978), pp. 111–129.

[4] J. R. BUNCH, C. R. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

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

[6] W. FERNG, G. H. GOLUB, AND R. J. PLEMMONS, *Adaptive Lanczos methods for recursive condition estimation*, in SPIE Vol. 1348, Advanced Signal-Processing Algorithms, Architectures, and Implementations, Washington, DC, The Internat. Society for Optical Engineering, 1990, pp. 326–337.

[7] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.

[8] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[9] D. J. PIERCE AND R. J. PLEMMONS, *Fast adaptive condition estimation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 274–291.

[10] G. M. SHROFF AND C. H. BISCHOF, *Adaptive condition estimation for rank-one updates of QR factorizations*, Preprint MCS-P166-0790, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, July 1990.

[11] D. SORENSEN AND P. T. P. TANG, *On the orthogonality of eigenvectors computed by a divide-and-conquer method*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.

[12] G. W. STEWART, *The efficient generation of random orthogonal matrices with an application to condition estimators*, SIAM J. Numer. Anal., 17 (1980), pp. 403–409.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 1, pp. 347–358, January 1994

027

ON THE STRUCTURE OF GENERALIZED SINGULAR VALUE AND QR DECOMPOSITIONS*

BART DE MOOR†

Abstract. This paper analyzes in detail the structure of generalizations of the singular value decomposition and the QR decomposition for any number of matrices. The structure is completely determined as a function of the ranks of the matrices or their products and concatenations.

Key words. ordinary, product, quotient, restricted singular value decomposition, QR decomposition, complete orthogonal factorization

AMS subject classifications. 15A09, 15A18, 15A21, 15A24, 65F20

1. Introduction.

In a previous paper [4], we introduced an infinite tree of generalizations of the ordinary singular value decomposition (OSVD) and we derived a constructive proof of it. All decompositions in this tree are considered as generalized singular value decompositions (GSVDs) and it was shown in [4] how all of them can be labeled with a sequence of the letters P and Q, where P stands for *product* and Q stands for *quotient*. In [5], we introduced a corresponding set of generalizations of the QR decomposition, which could be denoted by appropriate enumerations of the letters L (lower) and U (upper). It is the purpose of this paper to discuss in more detail the structure of these generalizations. In particular, we shall derive formulas for the dimensions of the blocks in the quasi-diagonal matrices of the GSVDs of [4] (Theorem 1.1 of this paper), or the triangular matrices in the GQRDs (generalized QR decompositions) of [5] (Theorem 1.2 of this paper), in terms of the ranks of the matrices involved and concatenations and products of these matrices.

This paper is organized as follows. In the remainder of this section, we summarize the main results on generalized SVDs and QRDs obtained in [4] and [5]. Since there is a one-to-one correspondence between these two generalizations, we will concentrate on the generalizations of the SVD, while the results will apply for the GQRDs as well. In §2, we analyze in detail the structure of a GSVD that only consists of P-steps. In §3, we analyze GSVDs that only contain Q-steps. In §4, we discuss the general case where we exploit the obtained insights from §§2 and 3. Instead of providing rigorous proofs, we have chosen to indicate our methods of deriving these results with illustrative examples.

Let us first state the main result of [4] in the following theorem.

THEOREM 1.1 (GSVDs for k matrices). *Consider a set of k matrices with compatible dimensions: A_1 $(n_0 \times n_1), A_2$ $(n_1 \times n_2), \ldots, A_{k-1}$ $(n_{k-2} \times n_{k-1}), A_k$ $(n_{k-1} \times n_k)$. Then there exist*

- *unitary matrices U_1 $(n_0 \times n_0)$ and V_k $(n_k \times n_k)$;*

* Received by the editors June 24, 1991; accepted for publication (in revised form) February 21, 1992. The results presented in this paper have been obtained in the framework of the Belgian Program on Concerted Research Actions and on the Interuniversity Attraction Poles initiated by the Belgian State, Prime Minister's Office, Science Policy Programming, and in the framework of the European Community Research Programme ESPRIT (European Strategic Programme for Research and Development in Information Technology) Basic Research Action 3280. The scientific responsibility rests with its author.

† This author is a research associate of the Belgian National Fund for Scientific Research (NFWO, Nationaal Fonds voor Wetenschappelijk Onderzoek) and associate professor at ESAT (Electronics, Systems, Automation and Technology), Department of Electrical Engineering, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, B-3001 Leuven, Belgium (demoor@esat.kuleuven.ac.be).

- *matrices* $D_j, j = 1, 2, \ldots, (k-1)$ *of the form*

$$
\begin{array}{c}
D_j \\
n_{j-1} \times n_j
\end{array}
=
\begin{array}{c}
\begin{array}{c} r_j^1 \\ r_{j-1}^1 - r_j^1 \\ r_j^2 \\ r_{j-1}^2 - r_j^2 \\ r_j^3 \\ \ldots \\ \ldots \\ r_j^j \\ n_{j-1} - r_{j-1} - r_j^j \end{array}
\begin{array}{c}
\begin{array}{ccccccc} r_j^1 & r_j^2 & r_j^3 & \ldots & \ldots & r_j^j & n_j - r_j \end{array} \\
\begin{pmatrix}
I & 0 & 0 & \ldots & \ldots & 0 & 0 \\
0 & 0 & 0 & \ldots & \ldots & 0 & 0 \\
0 & I & 0 & \ldots & \ldots & 0 & 0 \\
0 & 0 & 0 & \ldots & \ldots & 0 & 0 \\
0 & 0 & I & \ldots & \ldots & 0 & 0 \\
0 & \ldots & \ldots & \ldots & \ldots & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & \ldots & \ldots & \ldots & \ldots & I & 0 \\
0 & 0 & \ldots & \ldots & \ldots & 0 & 0
\end{pmatrix}
\end{array}
\end{array},
$$

where the integers r_j *are the ranks of the matrices* A_j*, satisfying*

$$r_j = \operatorname{rank}(A_j) = \sum_{i=1}^{j} r_j^i;$$

- *a matrix* S_k *of the form*

$$
\begin{array}{c}
S_k \\
n_{k-1} \times n_k
\end{array}
=
\begin{array}{c}
\begin{array}{c} r_k^1 \\ r_{k-1}^1 - r_k^1 \\ r_k^2 \\ r_{k-1}^2 - r_k^2 \\ r_k^3 \\ \ldots \\ \ldots \\ r_k^k \\ n_{k-1} - r_{k-1} - r_k^k \end{array}
\begin{array}{c}
\begin{array}{ccccccc} r_k^1 & r_k^2 & r_k^3 & \ldots & \ldots & r_k^k & n_k - r_k \end{array} \\
\begin{pmatrix}
S_k^1 & 0 & 0 & \ldots & \ldots & 0 & 0 \\
0 & 0 & 0 & \ldots & \ldots & 0 & 0 \\
0 & S_k^2 & 0 & \ldots & \ldots & 0 & 0 \\
0 & 0 & 0 & \ldots & \ldots & 0 & 0 \\
0 & 0 & S_k^3 & \ldots & \ldots & 0 & 0 \\
0 & \ldots & \ldots & \ldots & \ldots & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & \ldots & \ldots & \ldots & \ldots & S_k^k & 0 \\
0 & 0 & \ldots & \ldots & \ldots & 0 & 0
\end{pmatrix}
\end{array}
\end{array}.
$$

The $r_k^i \times r_k^i$ *matrices* S_k^i *are diagonal with positive diagonal elements.*

- *Nonsingular matrices* X_j $(n_j \times n_j)$ *and* Z_j*,* $j = 1, 2, \ldots, (k-1)$*, where* Z_j *is either* $Z_j = X_j^{-*}$ *or either* $Z_j = X_j$ *(i.e., both choices are always possible), such that the given matrices can be factorized as*

$$
\begin{aligned}
A_1 &= U_1 D_1 X_1^{-1}, \\
A_2 &= Z_1 D_2 X_2^{-1}, \\
A_3 &= Z_2 D_3 X_3^{-1}, \\
\cdots &= \cdots, \\
A_i &= Z_{i-1} D_i X_i^{-1}, \\
\cdots &= \cdots, \\
A_k &= Z_{k-1} S_k V_k^*.
\end{aligned}
$$

Expressions for the integers r_j^i *are given below; they are ranks of certain matrices in the constructive proof of this theorem* [4].

Observe that the matrices D_j and S_k are generally not diagonal. Their only nonzero blocks however are diagonal block matrices. Observe that we always take the

last factor in every factorization as the inverse of a nonsingular matrix, which is only a matter of convention. (Another convention would result in a modified definition of the matrices Z_i.) As to the name of a certain GSVD, we propose to adopt the following convention.

DEFINITION 1 (Nomenclature for GSVDs). If $k = 1$ in Theorem 1.1, then the corresponding factorization of the matrix A_1 will be called the OSVD. If for a matrix pair $A_i, A_{i+1}, 1 \le i \le k-1$ in Theorem 1.1, we have that $Z_i = X_i$ then, the factorization of the pair will be said to be of P-type. If, on the other hand, for a matrix pair $A_i, A_{i+1}, 1 \le i \le k-1$ in Theorem 1.1, we have that $Z_i = X_i^{-*}$ the factorization of the pair will be said to be of Q-type. The name of a GSVD of the matrices A_i, $i = 1, 2, \ldots, k > 1$ as in Theorem 1.1, is then obtained by simply enumerating the different factorization types.

We now give some examples.

Example 1. Consider two matrices A_1 $(n_0 \times n_1)$ and A_2 $(n_1 \times n_2)$. Then, we have the following two possible GSVDs.

	P-type	Q-type
A_1	$U_1 D_1 X_1^{-1}$	$U_1 D_1 X_1^{-1}$
A_2	$X_1 S_2 V_2^*$	$X_1^{-*} S_2 V_2^*$

The P-type factorization corresponds to the PSVD (product singular value decomposition) as in [9] (called ΠSVD there) and in [1] and [3], while the Q-type factorization is nothing else than the QSVD (quotient singular value decomposition) in [8], [10], and [11] (called generalized SVD there). A P-type factorization is precisely the kind of transformation that occurs in the PSVD while a Q-type factorization occurs in the QSVD.

Example 2. Let us write down the PQQP-SVD for five matrices:

$$\begin{aligned} A_1 &= U_1 D_1 X_1^{-1}, \\ A_2 &= X_1 D_2 X_2^{-1}, \\ A_3 &= X_2^{-*} D_3 X_3^{-1}, \\ A_4 &= X_3^{-*} D_4 X_4^{-1}, \\ A_5 &= X_4 S_5 V_5^*. \end{aligned}$$

In [5], we derived the following generalization of the QR decomposition for a chain of k matrices.

THEOREM 1.2 (Generalized QR decompositions for k matrices). *Given k complex matrices A_1 $(n_0 \times n_1)$, A_2 $(n_1 \times n_2)$, ..., A_k $(n_{k-1} \times n_k)$. There always exist unitary matrices $Q_0, Q_1, \ldots, Q_k$ such that $\tilde{T}_i = Q_{i-1}^* A_i Q_i$, where $\tilde{T}_i$ is a lower triangular or upper triangular matrix (both cases are always possible) with the following structure.*

Lower triangular (which will be denoted by a superscript l):

$$\tilde{T}_i^l = \begin{array}{c} \\ r_{i-1}^1 \\ r_{i-1}^2 \\ \vdots \\ r_{i-1}^i \end{array} \begin{array}{c} \begin{array}{cccccc} r_i^1 & r_i^2 & \ldots & r_i^i & r_i^{i+1} \end{array} \\ \begin{pmatrix} R_{i,1} & 0 & \ldots & 0 & 0 \\ * & R_{i,2} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & \ldots & R_{i,i} & 0 \end{pmatrix} \end{array}, \quad \text{where } R_{i,j} = \begin{pmatrix} 0 \\ R_{i,j}^l \end{pmatrix},$$

and $R_{i,j}^l$ is a square nonsingular lower triangular matrix.

Upper triangular (which will be denoted by a superscript u):

$$\tilde{T}_i^u = \begin{array}{c} \\ r_{i-1}^1 \\ r_{i-1}^2 \\ \vdots \\ r_{i-1}^i \end{array} \begin{array}{c} \begin{array}{ccccc} r_i^{i+1} & r_i^1 & \dots & r_i^{i-1} & r_i^i \end{array} \\ \begin{pmatrix} 0 & R_{i,1} & * & \dots & * \\ 0 & 0 & R_{i,2} & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & R_{i,i} \end{pmatrix} \end{array}, \quad \text{where } R_{i,j} = \begin{pmatrix} R_{i,j}^u \\ 0 \end{pmatrix},$$

and $R_{i,j}^u$ is a square nonsingular upper triangular matrix. The block dimensions r_i^j coincide with those of Theorem 1.1.

As to the nomenclature of these GQRDs, we propose the following definition.

DEFINITION 2 (Nomenclature for GQRD). The name of a GQRD of k matrices of compatible dimensions is generated by enumerating the letters L (for lower) and U (for upper), according to the lower or upper triangularity of the matrices $\tilde{T}_i, i = 1, \dots, k$ in the decomposition of Theorem 1.2.

For k matrices, there are 2^k different sequences with two letters. For instance, for $k = 3$, there are eight GQRDs (LLL, LLU, LUL, LLU, ULL, ULU, UUL, UUU).

The relation between the two generalizations, the GSVDs and the GQRDs, is the following:

(i) A pair of identical letters, i.e., L-L or U-U that occurs in the factorization of A_i, A_{i+1} corresponds to a P-type factorization of the pair;

(ii) A pair of alternating letters, i.e., L-U or U-L that occurs in the factorization of A_i, A_{i+1} corresponds to a Q-type factorization of the pair.

As an example, for a PQP-SVD of four matrices, there are two possible corresponding GQRDs, namely, an LLUL decomposition and an UULU decomposition. As with the GSVD, we can also introduce the convention to use powers of (a sequence of) letters. For instance, for a P^3Q^2-SVD (which is short for a PPPQQ-SVD), there are two QR decompositions, namely, an L^4UL-QR and an U^4LU-QR.

2. Structure of a GSVD with only P-steps. The main purpose of this section is to derive expressions for the block dimensions r_p^q when all steps in the GSVD are P-steps. These block dimensions will be expressed as a function of the ranks of products of the form

$$\operatorname{rank}(A_i A_{i+1}, \dots, A_{j-1} A_j),$$

which will be denoted by $r_{i(i+1)\dots(j-1)j}$. This will be done in two steps. First, we derive an implicit characterization of the block dimensions. This leads directly to an explicit determination of these block dimensions.

LEMMA 2.1. *The rank of the product of the matrices $D_i, D_{i+1}, \dots, D_j$ that appears in a* P^{k-1}-SVD (*or the rank of the product $A_i A_{i+1}, \dots, A_j$ in an* L^k-QR *or a* U^k-QR) *is given by*

$$\operatorname{rank}(D_i D_{i+1} \dots D_j) = r_{(i)(i+1)\dots(j)} = r_j^1 + r_j^2 + \dots + r_j^i.$$

As the examples will reveal, the following theorem follows directly from this lemma.

THEOREM 2.2. *Consider a* P^{k-1}-SVD *of the matrices $A_1, A_2, \dots, A_k$. Then, the block dimensions r_p^q, $p = 1, \dots, k$, $q = 1, \dots, p$ are given by*

$$r_j^1 = r_{(1)(2)\dots(j)},$$
$$r_j^i = r_{i(i+1)\dots(j)} - r_{(i-1)(i)\dots(j)},$$

with $r_{(i)\ldots(j)} = r_i$ *if* $i = j$.

Let us analyze an example from which we will see the general result.

Example 3 (P^3-SVD). Let us derive expressions for the block dimensions $r_4^1, r_4^2, r_4^3, r_4^4$ of the matrix S_4 in terms of $r_1, r_2, r_3, r_{(3)(4)}, r_{(2)(3)(4)}, r_{(1)(2)(3)(4)}$. The matrices D_1, D_2, D_3, S_4 have the following structure:

$$D_1 = \begin{array}{c} \\ r_1^1 \\ n_0 - r_1 \end{array} \begin{array}{c} \begin{array}{cc} r_1^1 & n_1 - r_1^1 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \end{array}, \qquad D_2 = \begin{array}{c} \\ r_2^1 \\ r_1^1 - r_2^1 \\ r_2^2 \\ n_1 - r_2^2 - r_1^1 \end{array} \begin{array}{c} \begin{array}{ccc} r_2^1 & r_2^2 & n_2 - r_2 \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array},$$

$$D_3 = \begin{array}{c} \\ r_3^1 \\ r_2^1 - r_3^1 \\ r_3^2 \\ r_2^2 - r_3^2 \\ r_3^3 \\ n_2 - r_2 - r_3^3 \end{array} \begin{array}{c} \begin{array}{cccc} r_3^1 & r_3^2 & r_3^3 & n_3 - r_3 \end{array} \\ \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array},$$

$$S_4 = \begin{array}{c} \\ r_4^1 \\ r_3^1 - r_4^1 \\ r_4^2 \\ r_3^2 - r_4^2 \\ r_4^3 \\ r_3^3 - r_4^3 \\ r_4^4 \\ n_3 - r_3 - r_4^4 \end{array} \begin{array}{c} \begin{array}{ccccc} r_4^1 & r_4^2 & r_4^3 & r_4^4 & n_4 - r_4 \end{array} \\ \begin{pmatrix} S_4^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & S_4^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & S_4^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & S_4^4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

From the structure and dimensions of these matrices, we see that (we only show block dimensions that are relevant)

$$D_3 S_4 = \begin{array}{c} \\ r_4^1 \\ r_3^1 - r_4^1 \\ r_2^1 - r_3^1 \\ r_4^2 \\ r_3^2 - r_4^2 \\ r_2^2 - r_3^2 \\ r_4^3 \\ r_3^3 - r_4^3 \\ n_2 - r_2 - r_3^3 \end{array} \begin{array}{c} \begin{array}{ccccc} r_4^1 & r_4^2 & r_4^3 & & \end{array} \\ \begin{pmatrix} S_4^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & S_4^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & S_4^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}, \qquad D_2 D_3 S_4 = \begin{array}{c} \\ r_4^1 \\ \\ \\ r_4^2 \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} r_4^1 & r_4^2 & & & \end{array} \\ \begin{pmatrix} S_4^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & S_4^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array},$$

$$D_1 D_2 D_3 S_4 = \begin{array}{c} \\ r_4^1 \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{c} r_4^1 \end{array} \\ \begin{pmatrix} S_4^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

We see by inspection that

$$\begin{aligned} r_{(1)(2)(3)(4)} &= r_4^1, \\ r_{(2)(3)(4)} &= r_4^1 + r_4^2, \\ r_{(3)(4)} &= r_4^1 + r_4^2 + r_4^3, \\ r_4 &= r_4^1 + r_4^2 + r_4^3 + r_4^4, \end{aligned}$$

from which it follows that

$$\begin{aligned} r_4^1 &= r_{(1)(2)(3)(4)}, \\ r_4^2 &= r_{(2)(3)(4)} - r_{(1)(2)(3)(4)}, \\ r_4^3 &= r_{(3)(4)} - r_{(2)(3)(4)}, \\ r_4^4 &= r_4 - r_{(3)(4)}. \end{aligned}$$

The same expressions apply for the blocks in the corresponding U^4- or L^4-QR.

Observe that in the product D_3 and S_4, only the diagonal blocks of S_4 with dimensions r_4^1, r_4^2, r_4^3 survive. In the product $D_2D_3S_4$ only the blocks with dimensions r_4^1 and r_4^2 survive, and in $D_1D_2D_3S_4$ only the block with dimension r_4^1 survives. This observation can easily be generalized to the following survival rule for a pure PSVD, which is the essence of the proof of Lemma 2.1.

In a product of matrices $D_i, D_{i+1}, \ldots, D_j$ (or S_k) only the blocks with block dimensions $r_j^1, r_j^2, \ldots, r_j^i$ survive.

Once this observation has been established, a proof of Theorem 3.2 is straightforward.

3. A GSVD with only Q-steps. Let us now look closer at the structure of a GSVD with only Q-steps. We will see that we can derive expressions for the block dimensions $r_p^q, p = 1, \ldots, k, q = 1, \ldots, p$ in two steps. First, we obtain an implicit formula where the required block dimensions are unknowns in a set of linear equations. In a second step, these are solved to obtain an expression for the block dimensions r_p^q in terms of the ranks of the block matrices

$$\begin{pmatrix} A_i & 0 & 0 & \ldots & 0 & 0 & 0 \\ A_{i+1}^* & A_{i+2} & 0 & \ldots & 0 & 0 & 0 \\ 0 & A_{i+3}^* & A_{i+4} & \ldots & 0 & 0 & 0 \\ 0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\ 0 & \ldots & \ldots & \ldots & A_{j-3}^* & A_{j-2} & 0 \\ 0 & \ldots & \ldots & \ldots & 0 & A_{j-1}^* & A_j \end{pmatrix}.$$

Their rank is denoted by $r_{i|i+1|\ldots|j-1|j}$.

We will proceed in the same way as in §2. Instead of proving our results rigorously, we prefer to reveal the mechanisms by some clarifying examples. First, we obtain the following implicit characterization.

LEMMA 3.1. *Consider a* Q^{k-1}-SVD *of the matrices* $A_1, A_2, \ldots, A_k$. *Then if* $j - i$ *even*

$$r_{i|\ldots|j} = r_{i|\ldots|j-1} + (r_j^1 + r_j^2 + \cdots + r_j^i) + r_j^{i+2} + r_j^{i+4} + \cdots + r_j^{j-2} + r_j^j;$$

if $j - i$ *odd*

$$r_{i|\ldots|j} = r_{i|\ldots|j-1} + (r_j^{i+1} + r_j^{i+3} + \cdots + r_j^{j-2} + r_j^j).$$

As will be shown, this lemma leads to the following theorem.

THEOREM 3.2. *Consider a* Q^{k-1}*-SVD of the matrices* $A_1, A_2, \dots, A_k$*. Then*

$$
\begin{aligned}
r_k^1 &= (-1)^{k+1}(r_{1|\dots|k} - r_{1|\dots|k-1} - r_{2|\dots|k} + r_{2|\dots|k-1}),\\
r_k^j &= (-1)^{j+k+1}(r_{(j+1)|\dots|k} - r_{(j+1)|\dots|k-1} - r_{(j-1)|\dots|k} + r_{(j-1)|\dots|k-1})\\
&\qquad \textit{for } 2 \le j \le k-2,\\
r_k^{k-1} &= r_k - r_{k-2|k-1|k} + r_{k-2|k-1},\\
r_k^k &= r_{k-1|k} - r_{k-1}.
\end{aligned}
$$

Observe that in all cases, no more than four ranks $r_{i|\dots|j}$ are involved. Also, the third case may be recognized as Grassman's dimension theorem, giving the dimension of the intersection of the column spaces of the matrices

$$
\begin{pmatrix} A_{k-2} \\ A_{k-1}^* \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 \\ A_k \end{pmatrix}.
$$

Let us derive the result of Theorem 3.2 by an example.

Example 4 (QQ-SVD).[1] A QQ-SVD of three matrices A_1 $(n_0 \times n_1)$, $A_2(n_1 \times n_2)$, and A_3 $(n_2 \times n_3)$ takes the form

$$
\begin{aligned}
A_1 &= U_1 D_1 X_1^{-1},\\
A_2 &= X_1^{-*} D_2 X_2^{-1},\\
A_3 &= X_2^{-*} S_3 V_3^*,
\end{aligned}
$$

where

$$
D_1 = \begin{array}{c} \\ r_1^1 \\ n_0 - r_1 \end{array} \begin{array}{c} \begin{array}{cc} r_1^1 & n_1 - r_1^1 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \end{array}, \qquad
D_2 = \begin{array}{c} \\ r_2^1 \\ r_1^1 - r_2^1 \\ r_2^2 \\ n_1 - r_2^2 - r_1 \end{array} \begin{array}{c} \begin{array}{ccc} r_2^1 & r_2^2 & n_2 - r_2 \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array},
$$

$$
S_3 = \begin{array}{c} \\ r_3^1 \\ r_2^1 - r_3^1 \\ r_3^2 \\ r_2^2 - r_3^2 \\ r_3^3 \\ n_3 - r_2 - r_3^3 \end{array} \begin{array}{c} \begin{array}{cccc} r_3^1 & r_3^2 & r_3^3 & n_3 - r_3 \end{array} \\ \begin{pmatrix} S_3^1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & S_3^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & S_3^3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.
$$

Observe that

$$
\begin{pmatrix} A_1 & 0 \\ A_2^t & A_3 \end{pmatrix} = \begin{pmatrix} U_1 & 0 \\ 0 & X_2^{-*} \end{pmatrix} \begin{pmatrix} D_1 & 0 \\ D_2^t & S_3 \end{pmatrix} \begin{pmatrix} X_1^{-1} & 0 \\ 0 & V_3^* \end{pmatrix}.
$$

[1] A complete detailed analysis of the QQ-SVD (which is also called the restricted singular value decomposition (RSVD)) together with numerous applications can be found in [2].

The left and right factors are nonsingular. Hence, we can obtain expressions for all dimensions involved by analyzing the block bidiagonal matrix

$$\left(\begin{array}{c|c} D_1 & 0 \\ \hline D_2^t & S_3 \end{array}\right) = \begin{array}{l} r_3^1 \\ r_2^1 - r_3^1 \\ r_1 - r_2^1 \\ n_0 - r_1 \\ \hline r_3^1 \\ r_2^1 - r_3^1 \\ r_3^2 \\ r_2^2 - r_3^2 \\ r_3^3 \\ n_2 - r_2 - r_3^3 \end{array} \left(\begin{array}{ccc|ccc|cccc} I & 0 & 0 & & & & & & & \\ 0 & I & 0 & & & & & & & \\ 0 & 0 & I & & & & & & & \\ 0 & 0 & 0 & & & & & & & \\ \hline I & 0 & 0 & & & & S_3^1 & 0 & 0 & 0 \\ 0 & I & 0 & & & & 0 & 0 & 0 & 0 \\ & & & I & 0 & 0 & 0 & S_3^2 & 0 & 0 \\ & & & 0 & I & 0 & 0 & 0 & 0 & 0 \\ & & & & & & 0 & 0 & S_3^3 & 0 \\ & & & & & & 0 & 0 & 0 & 0 \end{array}\right),$$

where we have used the finest possible subdivision of matrices (i.e., a partitioning based upon the block dimensions r_3^1, r_3^2, r_3^3). All nonzero blocks are diagonal. Elements not shown are zero. First, it is straightforward to see that $r_3 = r_3^1 + r_3^2 + r_3^3$. Next, we concentrate on the submatrix $\begin{pmatrix} D_2^t & S_3 \end{pmatrix}$. In this matrix, the block columns with the matrices S_3^1 and S_3^2 are linearly dependent on the previous ones. The block column with S_3^3 is linearly independent. Hence $\operatorname{rank}\begin{pmatrix} D_2^t & S_3 \end{pmatrix} = r_{2|3} = r_2 + r_3^3$. Next, we will relate the rank $r_{1|2} = \operatorname{rank}\begin{pmatrix} D_1^t & D_2 \end{pmatrix}^t$ to

$$r_{1|2|3} = \operatorname{rank}\begin{pmatrix} D_1 & 0 \\ D_2^t & S_3 \end{pmatrix}.$$

It can be seen that when the block column with S_3^1 is appended to $\begin{pmatrix} D_1^t & D_2 \end{pmatrix}^t$, the rank will increase with

$$r_3^1 \times \left[\operatorname{rank}\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} - \operatorname{rank}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right] = r_3^1.$$

If the block column with S_3^2 is appended to $\begin{pmatrix} D_1^t & D_2 \end{pmatrix}^t$, the rank will not increase. Finally, if the block column with S_3^1 is appended, then the rank will increase with $r_3^3 \times [\operatorname{rank}(0 \;\; 1) - \operatorname{rank}(0)] = r_3^3$. Hence

$$\begin{aligned} r_{1|2|3} = r_{1|2} &+ r_3^1 \times \left[\operatorname{rank}\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} - \operatorname{rank}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right] \\ &+ r_3^2 \times [\operatorname{rank}(1 \;\; 1) - \operatorname{rank}(1)] \\ &+ r_3^3 \times [\operatorname{rank}(0 \;\; 1) - \operatorname{rank}(0)] = r_{1|2} + r_3^1 + r_3^3. \end{aligned}$$

We can now set up a set of linear equations as

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_3^1 \\ r_3^2 \\ r_3^3 \end{pmatrix} = \begin{pmatrix} r_3 \\ r_{2|3} - r_2 \\ r_{1|2|3} - r_{1|2} \end{pmatrix},$$

which can be solved as

$$\begin{pmatrix} r_3^1 \\ r_3^2 \\ r_3^3 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_3 \\ r_{2|3} - r_2 \\ r_{1|2|3} - r_{1|2} \end{pmatrix} = \begin{pmatrix} r_{1|2|3} - r_{1|2} + r_2 - r_{2|3} \\ r_3 + r_{1|2} - r_{1|2|3} \\ r_{2|3} - r_2 \end{pmatrix}.$$

The same expressions will appear in the ULU- or LUL-QR.

Example 5 (Q^7-SVD). The courageous reader may wish to verify that for $k = 7$, the following set of linear equations needs to be solved:

$$\begin{pmatrix} r_7 \\ r_{6|7} - r_6 \\ r_{5|6|7} - r_{5|6} \\ r_{4|5|6|7} - r_{4|5|6} \\ r_{3|4|5|6|7} - r_{3|4|5|6} \\ r_{2|3|4|5|6|7} - r_{2|3|4|5|6} \\ r_{1|2|3|4|5|6|7} - r_{1|2|3|4|5|6} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_7^1 \\ r_7^2 \\ r_7^3 \\ r_7^4 \\ r_7^5 \\ r_7^6 \\ r_7^7 \end{pmatrix}.$$

This set of equations can be solved as

$$\begin{pmatrix} r_7^1 \\ r_7^2 \\ r_7^3 \\ r_7^4 \\ r_7^5 \\ r_7^6 \\ r_7^7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} r_7 \\ r_{6|7} - r_6 \\ r_{5|6|7} - r_{5|6} \\ r_{4|5|6|7} - r_{4|5|6} \\ r_{3|4|5|6|7} - r_{3|4|5|6} \\ r_{2|3|4|5|6|7} - r_{2|3|4|5|6} \\ r_{1|2|3|4|5|6|7} - r_{1|2|3|4|5|6} \end{pmatrix}.$$

The pattern of the inverse matrix now becomes clear. We have a triantidiagonal matrix with a sequence of alternating 1 and -1, ending in a 1 in the top right-hand corner. As a matter of fact, this observation constitutes the essence of a proof of Theorem 3.2.

4. On the structure of a GSVD. For the analysis of the structure of a completely general GSVD, in which the letters P and Q can appear in any order, we need a mixture of the two preceding notations for block bidiagonal matrices, the blocks of which can be products of matrices, such as

$$\begin{pmatrix} A_{i_0} A_{i_0+1} \dots A_{i_1-1} & 0 & 0 & \dots & 0 \\ (A_{i_1} \dots A_{i_2-1})^* & A_{i_2} \dots A_{i_3-1} & 0 & \dots & 0 \\ 0 & (A_{i_3} \dots A_{i_4-1})^* & A_{i_4} \dots A_{i_5-1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & A_{i_l} \dots A_j \end{pmatrix},$$

where $1 \le i_0 < i_1 < i_2 < i_3 < \cdots < i_l \le j \le k$. Their rank will be denoted by

$$r_{(i_0)\dots(i_1-1)|i_1\dots(i_2-1)|\dots|i_l\dots(j)}.$$

For instance, the rank of the matrix

$$\begin{pmatrix} A_2 A_3 & 0 & 0 \\ A_4^* & A_5 A_6 A_7 & 0 \\ 0 & (A_8 A_9)^* & A_{10} \end{pmatrix}$$

will be represented by $r_{(2)(3)|4|(5)(6)(7)|(8)(9)|(10)}$.

In the following theorem, we derive an implicit expression of the block dimensions $r_p^q, p = 1, \dots, k, q = 1, \dots, p$ of a **GSVD** of $A_1, A_2, \dots, A_k$. We proceed in two steps. The first part of the theorem is based on the survival rule described in Lemma 2.1, and the second part is then an application of the pure Q-step SVD in Lemma 3.1.

THEOREM 4.1 (On the structure of a GSVD or GQRD). *The rank*

$$r_{(i_0)(i_0+1)\dots(i_1-1)|i_1\dots(i_2-1)|\dots|i_l\dots j}$$

can be expressed as follows:

1. *Calculate the* $l+1$ *integers* $s_j^i, i = 1, 2, \dots, l+1$.

$$\begin{aligned} s_j^1 &= r_j^1 + r_j^2 + \cdots + r_j^{i_0}, \\ s_j^2 &= r_j^{i_0+1} + r_j^{i_0+2} + \cdots + r_j^{i_1}, \\ \cdots &= \cdots, \\ s_j^{l+1} &= r_j^{i_{l-1}+1} + r_j^{i_{l-1}+2} + \cdots + r_j^{i_l}. \end{aligned}$$

2. *Depending on* l *even or* l *odd there are two cases.*

l *even*:

$$\begin{aligned} & r_{(i_0)\dots(i_1-1)|(i_1)\dots(i_2-1)|\dots|(i_l)\dots j} \\ & = r_{(i_0)\dots(i_1-1)|(i_1)\dots(i_2-1)|\dots|(i_{l-1})\dots(i_l-1)} + s_j^1 + s_j^3 + \cdots + s_j^{l+1}. \end{aligned}$$

l *odd*:

$$\begin{aligned} & r_{(i_0)\dots(i_1-1)|(i_1)\dots(i_2-1)|\dots|(i_l)\dots j} \\ & = r_{(i_0)\dots(i_1-1)|(i_1)\dots(i_2-1)|\dots|(i_{l-1})\dots(i_l-1)} + s_j^2 + s_j^4 + \cdots + s_j^{l+1}. \end{aligned}$$

Again, we will not give an unreadable algebraic proof of this theorem, but instead we illustrate it with an example.

Example 6 (QPPQ-SVD). A QPPQ-SVD of five matrices A_1, A_2, A_3, A_4, A_5 can be analyzed in terms of the ranks of the matrices

$$\begin{pmatrix} D_1 & 0 \\ (D_2D_3D_4)^t & S_5 \end{pmatrix}, \quad \begin{pmatrix} (D_2D_3D_4)^t & S_5 \end{pmatrix}, \quad \begin{pmatrix} (D_3D_4)^t & S_5 \end{pmatrix}, \quad \begin{pmatrix} D_4^t & S_5 \end{pmatrix}.$$

Let us first consider the first matrix

$$\left(\begin{array}{c||c} D_1 & 0 \\ \hline\hline (D_2D_3D_4)^t & S_5 \end{array}\right) =$$

$$\begin{array}{c} r_5^1 \\ \\ \\ \\ \\ \\ \hline\hline r_5^1 \\ r_4^1 - r_5^1 \\ \hline r_5^2 \\ r_4^2 - r_5^2 \\ \hline r_5^3 \\ \\ \hline r_5^4 \\ \\ \hline r_5^5 \\ \\ \hline \end{array} \left(\begin{array}{ccccc|ccccc||cccccc} 1 & 0 & 0 & 0 & 0 & & & & & & & & & & & \\ 0 & 1 & 0 & 0 & 0 & & & & & & & & & & & \\ 0 & 0 & 1 & 0 & 0 & & & & & & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & & & & & & & & & & & \\ \hline\hline 1 & 0 & 0 & 0 & 0 & & & & & & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ & & & & & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & & & & & & & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & & & & & & & 0 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & & & & & & & 0 & 0 & 0 & 0 & 1 & 0 \\ & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

Elements not shown are represented by 0 while 1 represents a nonzero square diagonal matrix. Obviously, $r_5 = r_5^1+r_5^2+r_5^3+r_5^4+r_5^5$. Also, $r_{(2)(3)(4)|5} = r_{(2)(3)(4)}+(r_5^3+r_5^4+r_5^5)$. Using the notation of Theorem 4.1, we have $s_5^1 = r_5^1 + r_5^2$ and $s_5^2 = r_5^3 + r_5^4 + r_5^5$, so that indeed $r_{(2)(3)(4)|5} = r_{(2)(3)(4)} + s_5^2$. Also,

$$r_{1|(2)(3)(4)|5} = r_{1|(2)(3)(4)} + r_5^1 \times \left[\operatorname{rank}\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} - \operatorname{rank}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right] + r_5^2 \times [\operatorname{rank}(1\ \ 1) - \operatorname{rank}(1)] + (r_5^3 + r_5^4 + r_5^5).$$

With the notation of Theorem 4.1, we have for this case $s_5^1 = r_5^1$, $s_5^2 = r_5^2$, $s_5^3 = r_5^3+r_5^4+r_5^5$, so that indeed $r_{1|234|5} = r_{1|234}+s_5^1+s_5^3$. Up to now, we have three implicit equations for the five unknowns $r_5^1, r_5^2, r_5^3, r_5^4, r_5^5$. The remaining two are found from the matrix $(D_4^t \ \ S_5)$ as $r_{4|5} = r_4 + r_5^5$ and from

$$((D_3D_4)^t \ \ S_5) = \begin{array}{l} r_5^1 \\ r_4^1 - r_5^1 \\ \hline r_5^2 \\ r_4^2 - r_5^2 \\ \hline r_5^3 \\ r_4^3 - r_5^3 \\ \hline r_5^4 \\ \\ \hline r_5^5 \\ \\ \hline \end{array} \left(\begin{array}{ccc|ccc|ccc|c||cccccc} 1 & 0 & 0 & & & & & & & & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & 1 & 0 & 0 & & & & & 0 & 1 & 0 & 0 & 0 & 0 \\ & & & 0 & 1 & 0 & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & & & 1 & 0 & 0 & & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & & & & 0 & 1 & 0 & & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & & & & & & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & & & & & & & 0 & 0 & 0 & 0 & 1 & 0 \\ & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

From this we find that

$$r_{34|5} = r_{3|4} + (r_5^1 + r_5^2 + r_5^3) \times [\operatorname{rank}(1\ \ 1) - \operatorname{rank}(1)] + r_5^4 + r_5^5 = r_{3|4} + (r_5^4 + r_5^5).$$

With the notation of Theorem 4.1 we have $s_5^1 = r_5^1 + r_5^2 + r_5^3$ and $s_5^2 = r_5^4 + r_5^5$, so that indeed $r_{(3)(4)|5} = r_{3|4} + s_5^2$. From these equations we now find

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} r_5^1 \\ r_5^2 \\ r_5^3 \\ r_5^4 \\ r_5^5 \end{pmatrix} = \begin{pmatrix} r_5 \\ r_{234|5} - r_{234} \\ r_{1|234|5} - r_{1|234} \\ r_{4|5} - r_4 \\ r_{34|5} - r_{34} \end{pmatrix},$$

which, upon solution, results in

$$\begin{aligned} r_5^1 &= r_{1|234|5} - r_{1|234} - r_{234|5} + r_{234}, \\ r_5^2 &= r_5 - r_{1|234|5} + r_{1|234}, \\ r_5^3 &= r_{234|5} - r_{234} - r_{34|5} + r_{34}, \\ r_5^4 &= r_{34|5} - r_{34} - r_{4|5} + r_4, \\ r_5^5 &= r_{4|5} - r_4. \end{aligned}$$

5. Conclusions. In this paper, we have analyzed in detail the structure of some recently introduced generalizations of the singular value and the QR decomposition. The structure is completely determined in terms of the ranks of the involved matrices

and other matrices that are formed from products and concatenations of these matrices. Some more examples and details can be found in the technical report [6] and the papers [2]–[5], and [7].

REFERENCES

[1] B. De Moor and G. H. Golub, *Generalized singular value decompositions: A proposal for a standardized nomenclature*, ESAT-SISTA Report 1989-10, Apr. 1989, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, Belgium; Also, Numerical Analysis Project Manuscript NA-89-04, Dept. of Computer Science, Stanford University, Stanford, CA.

[2] ———, *The restricted singular value decomposition: properties and applications*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 401–425.

[3] B. De Moor, *On the structure and geometry of the product singular value decomposition*, Linear Algebra Appl., 168 (1992).

[4] B. De Moor and H. Zha, *A tree of generalizations of the ordinary singular value decomposition*, Linear Algebra Appl., special issue on Canonical Forms of Matrices, 147 (1991), pp. 469–500.

[5] B. De Moor and P. Van Dooren, *Generalizations of the QR and the singular value decomposition*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 993–1014.

[6] B. De Moor, *On the structure of generalized singular value and QR decompositions*, ESAT-SISTA Report 1990-12, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, Belgium, 1990.

[7] ———, *Generalizations of the singular value and QR decompositions*, Signal Processing, 25 (1991), pp. 135–146.

[8] G. H. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[9] K. V. Fernando and S. J. Hammarling, *A product induced singular value decomposition for two matrices and balanced realisation*, in Linear Algebra in Signal Systems and Control, B. N. Datta, C. R. Johnson, M. A. Kaashoek, R. Plemmons, and E. Sontag, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1988, pp. 128–140,

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

[11] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 359–365, April 1994

STRONG HALL MATRICES*

RICHARD A. BRUALDI† AND BRYAN L. SHADER‡

Abstract. The authors develop an inductive structure for nonsquare strong Hall matrices that is quite analogous to the well-known inductive structure of square strong Hall (i.e., fully indecomposable) matrices. Other properties of strong Hall matrices are also discussed.

Key words. Hall matrices, trees, matchings, bipartite graphs

AMS subject classifications. 05B20, 05C50, 15A21

1. Introduction. Let M be an m by n matrix with $m \geq n$. Then M is a *Hall matrix* provided there does not exist a zero submatrix of M of size r by s with $r + s > m$. If in addition there does not exist a zero matrix with $1 \leq s \leq n - 1$ and $r + s = m$, then M is a *strong Hall matrix.* For example, each of the matrices

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

is a strong Hall matrix. Replacing any 1 in the first matrix with a 0 gives a Hall matrix that is not a strong Hall matrix. A square strong Hall matrix is usually called a *fully indecomposable matrix.* Strong Hall matrices are important in sparse matrix analysis. For instance, in [4] an algorithm is given that correctly predicts the nonzero structure of the upper triangular factor R in the QR factorization of a strong Hall matrix. In [5] and [7], the nonzero structure of Q is correctly computed. Strong Hall matrices also arise in computing the fill in the Cholesky factorization [3].

Since whether or not a matrix is a Hall matrix or strong Hall matrix depends only on the locations of its 0's, we henceforth consider only (0,1)-matrices.

Let M be an m by n (0,1)-matrix with $m \geq n$. A *k-diagonal* of M is a collection of k ones of M with no two in the same row or column. An n-diagonal contains a unique one from each column of M and is also called a *column-diagonal.* A *cover* of M is a set of rows and columns of M that contain all the ones of M. A *minimum cover* of M is a cover of smallest cardinality. A *proper cover* is a cover with at least one row and at least one column. The next two lemmas follow from the well-known theorem of Hall(see, e.g., [2]) and the above definitions.

LEMMA 1.1. *Let M be an m by n $(0,1)$-matrix. Then the following are equivalent:*

(i) *M is a Hall matrix;*

(ii) *M has a column-diagonal;*

* Received by the editors January 20, 1992; accepted for publication (in revised form) September 21, 1992.

† Department of Mathematics, University of Wisconsin, Madison, Wisconsin 53706 (brualdi@math.wisc.edu). This paper was written while the author was a member of the Institute for Mathematics and Its Applications, University of Minnesota, Minneapolis, Minnesota 55455. This author's research was partially supported by National Science Foundation grant DMS-8901445 and National Security Agency grant MDA904-89-H-2060.

‡ Department of Mathematics, University of Wyoming, Laramie, Wyoming 82071 (bshader@corral.uwyo.edu). This paper was written while the author was a postdoctoral fellow at the Institute of Mathematics and Its Applications, University of Minnesota, Minneapolis, Minnesota 55455. This author's research was partially supported by a University of Wyoming Basic Research Grant.

(iii) *Every cover of M has cardinality at least n;*

(iv) *For each k with $1 \leq k \leq n$, every set of k columns of M contain 1's in at least k different rows.*

If $n = 1$, then M is a strong Hall matrix if and only if M is a Hall matrix.

LEMMA 1.2. *Let M be an m by n $(0,1)$-matrix with $n \geq 2$. Then the following are equivalent:*

(i) *M is a strong Hall matrix;*

(ii) *Every $m-1$ by $n-1$ submatrix of M is a Hall matrix;*

(iii) *M is a Hall matrix and no minimum cover is proper;*

(iv) *For each k with $1 \leq k \leq n-1$, every set of k columns of M contain 1's in at least $k+1$ different rows.*[1]

An immediate consequence of (ii) in Lemma 1.2 is that every 1 of a strong Hall matrix belongs to at least one column-diagonal.

Let M be an m by n (0,1)-matrix with no zero rows. Then it follows from Lemmas 1.1 and 1.2 that if $m = n$, then M is a strong Hall matrix if and only if the cover consisting of all the rows and the cover consisting of all the columns are the only minimum covers of M, and if $m > n$, then M is a strong Hall matrix if and only if the cover consisting of all the columns is the only minimum cover of M. This implies that the direct sum of two strong Hall matrices neither of which has a zero row is a strong Hall matrix if and only if neither is a square matrix.

We now define the usual bipartite graph associated with a matrix. Let $M = [m_{ij}]$ be an m by n (0,1)-matrix. Let $G = G(M)$ be the bipartite graph with vertices $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$ and an edge joining x_i and y_j if and only if $m_{ij} = 1$ $(1 \leq i \leq m, 1 \leq j \leq n)$. We call the vertices in X the *row vertices* and the vertices in Y the *column vertices* of G. Note that there is a one-to-one correspondence between the k-diagonals of M and *matchings* of G with k edges. A *column-matching* of G is a matching with n edges and hence corresponds to a column-diagonal of M. If M is a Hall matrix, respectively, a strong Hall matrix, then we say that $G(M)$ is a *Hall graph*, respectively, *strong Hall graph*. Let S be a subset of the edges of G and let $\overline{S}$ denote the set of edges of G not in S. A path from a vertex v to a vertex w is an *S-alternating path* provided that the first, third, ... edges belong to S and the second, fourth, ... edges belong to $\overline{S}$. Thus if v is a row vertex and w is a column vertex (or the other way around) the last edge of an S-alternating path belongs to S. If both v and w are row vertices or both are column vertices, the last edge of an S-alternating path belongs to $\overline{S}$.

A square strong Hall matrix is also known as a *fully indecomposable matrix*. A square matrix M is fully indecomposable if and only if the graph $G(M)$ is elementary, where a bipartite graph is *elementary* provided it is connected and each edge is in a column matching. There is a well-known inductive structure for fully indecomposable matrices M (see, e.g., [2]). This inductive structure is equivalent to an *ear decomposition* of $G(M)$ as defined in [6]. Our main purpose is to develop an inductive structure for nonsquare strong Hall matrices M. This inductive structure could be derived from either the inductive structure of M or an ear decomposition of $G(M)$, but we have taken a more revealing approach using the concept of a strong Hall tree defined

[1] Because of (iv) in Lemma 1.2, our definition of a strong Hall matrix is equivalent to the original definition given in [3]. There is another definition of a strong Hall matrix that is used in the literature[4],[5],[7],[8]. It is that property (iv) holds for each k with $1 \leq k < m$. The only difference is that an m by n matrix with $m > n$, which has $m - n$ zero rows and is a strong Hall matrix by our definition, is not a strong Hall matrix by the other definition. All of the results in §§2 and 3 hold regardless of which definition is used.

in this paper. As an application, we strengthen and make more transparent some results of Gilbert [4] concerning S-alternating paths where S is a column-matching of a bipartite graph with the strong Hall property.[2]

In §2 we study strong Hall matrices whose graphs are trees and characterize them in three ways. In §3 we use strong Hall trees to obtain two charaterizations of strong Hall matrices. We also obtain an upper bound on the number of 1's in strong Hall matrices with n columns that are minimal in a sense to be made precise later. We assume that the reader is familiar with standard graph theory terminology.

2. Strong Hall trees. In this section we study strong Hall matrices whose associated bipartite graphs are trees, that is, strong Hall trees. Our first result gives a simple characterization of strong Hall trees. Recall that a *leaf* (*pendant vertex*) of a tree is a vertex that belongs to exactly one edge.

THEOREM 2.1. *Let M be an m by n $(0,1)$-matrix with $m+n \geq 3$, and assume that the bipartite graph $G(M)$ is a tree. Then $G(M)$ is a strong Hall graph if and only if no column vertex of $G(M)$ is a leaf.*

Proof. If some column vertex of $G(M)$ is a leaf, then M has a minimum cover that is proper, and hence by Lemma 1.2 $G(M)$ is not strong Hall. Now assume that $G(M)$ is not a strong Hall graph. It follows from Lemma 1.2 that after row and column permutations,

$$M = \begin{bmatrix} M_1 & O \\ M_2 & M_3 \end{bmatrix},$$

where M_3 is a square matrix of order $b \geq 1$. The bipartite graph $G(M_3)$ is an induced subgraph of $G(M)$ and hence has no cycles. Since $G(M_3)$ has $2b$ vertices, it has at most $2b-1$ edges and hence some column of M_3 and the corresponding column of M contains a unique 1. Therefore some column vertex of $G(M)$ is a leaf. □

Note that the theorem implies that if M is an m by n strong Hall matrix with $m+n \geq 3$ such that $G(M)$ is a tree, then $m > n$. Using Theorem 2.1 we now obtain an inductive structure for strong Hall trees. This inductive structure is a consequence of being able to carry out the algorithm described below.

ALGORITHM

Let M be an m by n $(0,1)$-matrix such that $G(M)$ is a tree.

(0) Set $i = 0$.

(1) Choose a path γ_0 joining two row vertices and let T_0 be the subgraph of $G(M)$ determined by γ_0.

(2) While there exists a row vertex not in T_i, do: Choose a path γ_{i+1} joining a row vertex not in T_i to a vertex in T_i, all of whose intermediate vertices are not in T_i, and let T_{i+1} be the graph obtained by adjoining γ_{i+1} to T_i. Replace i by $i+1$.

(3) Let $k = i$.

THEOREM 2.2. *Let M be an m by n strong Hall $(0,1)$-matrix with $m+n \geq 3$ and assume that $G(M)$ is a tree. Then each of the trees $T_0, T_1, \ldots, T_k$ constructed in the above algorithm is a strong Hall tree and $T_k = G(M)$.*

Proof. Since $m+n \geq 3$, there is a path γ_0 joining two row vertices. It follows inductively that each T_i is a tree in which no column vertex is a leaf and hence by

[2] This was our original motivation for considering strong Hall matrices.

Theorem 2.1 each is a strong Hall tree. Suppose that $T_k \neq G(M)$. Then there exists at least one vertex y of $G(M)$ that is not a vertex of T_k and all such vertices are column vertices. Since $G(M)$ is a tree, this implies that y is a leaf, contradicting Theorem 2.1. □

COROLLARY 2.3. *Let M be an m by n strong Hall $(0,1)$-matrix such that $G(M)$ is a tree. Let γ_0 be any path from a row vertex v to a row vertex w. Then there exists a column matching F such that γ_0 is an F-alternating path.*

Proof. We refer to the above algorithm. Let F be the set of edges consisting of the first, third, ... edges of the paths γ_i of even length and the second, fourth, ... edges of the paths γ_i of odd length. The result now follows by induction. □

We now obtain two more characterizations of strong Hall trees.

THEOREM 2.4. *Let M be an m by n $(0,1)$-matrix with $m+n \geq 3$ and assume that $G(M)$ is a tree. Then the following are equivalent:*

(i) *M is a strong Hall matrix;*

(ii) *Given any two distinct vertices v and w there exist column-matchings F_1 and F_2 such that the path from v to w in $G(M)$ is F_1-alternating and $\overline{F}_2$-alternating.*

(iii) *Every edge of $G(M)$ is in a column-matching.*

Proof. We first suppose that (i) holds and prove (ii). Let v and w be distinct row vertices. By Corollary 2.3 F_1 exists, and by interchanging the roles of v and w we see that F_2 also exists. By Theorem 2.1, no column vertex of $G(M)$ is a leaf and this implies that each path of $G(M)$ is a subpath of a path joining two row vertices. Hence (ii) holds. Clearly (ii) implies (iii). We now suppose that (iii) holds and prove that M is a strong Hall matrix. Let O be an r by s zero submatrix of M. Since $G(M)$ is connected, the submatrix complementary to O contains a 1. By (iii) the $m-1$ by $n-1$ matrix obtained from M by deleting the row and column of this 1 is a Hall matrix. By Lemma 1.1, $r+s \leq m-1$, and hence by Lemma 1.2, M is a strong Hall matrix. □

We conclude this section with the following observation about extremal strong Hall trees. The identity matrix of order k is denoted by I_k.

THEOREM 2.5. *Let M be an m by n strong Hall $(0,1)$-matrix with $n \geq 3$. Assume that $G(M)$ is a tree and that each matrix obtained from M by deleting a row is not a strong Hall matrix. Then $m \leq 2n-2$ with equality if and only if there exist permutation matrices P and Q so that*

$$PMQ = \left[\begin{array}{c|c} & 1 \\ I_{n-1} & \vdots \\ & 1 \\ \hline & 0 \\ I_{n-1} & \vdots \\ & 0 \end{array}\right]. \tag{1}$$

Proof. By Theorem 2.1 no column vertex of $G(M)$ is a leaf. A simple calculation shows that the number k of row vertices of $G(M)$ that are leaves satisfies $k \geq m-n+1$. Let x be a row vertex that is a leaf and let y be the unique column vertex adjacent to x. If the degree of y is greater than 2, then it follows from Theorem 2.1 that the matrix obtained from M by deleting the row corresponding to x is a a strong Hall matrix. Since $n \geq 2$, we conclude that the degree of y equals 2 and no column vertex

is adjacent to two leaves and hence $k \leq n$. Therefore $m \leq 2n-1$. Suppose that $m = 2n-1$. Then $k = n$ and the degree of each column vertex equals 2, implying that the number of edges of $G(M)$ equals $2n$. Since the number of edges of $G(M)$ also equals $3n-2$, we contradict our assumption that $n \geq 3$. Hence $m \leq 2n-2$. Assume that $m = 2n-2$. Then $k = n-1$ and the column sums of M are $2, \ldots, 2, n-1$. The column vertex with degree equal to $n-1$ is adjacent to the $n-1$ row vertices that are not leaves, and it is easy to see that there exist permutation matrices P and Q such that (1) holds. Since the matrix in (1) satisfies the assumptions in the theorem when $n \geq 3$, the theorem now follows. □

3. Strong Hall matrices with connected graphs. Let M be an m by n strong Hall (0,1)-matrix with $m > n$. We show that if the graph $G(M)$ is connected, then $G(M)$ has a strong Hall spanning tree and this enables us to apply many of the results of the previous section. We first review a special form for strong Hall matrices given in [1] and [2; Exer. 1, p. 117] (although not called strong Hall matrices in these references).

By Lemma 1.2, M has a column diagonal and after row and column permutations, we may assume that M has the form

$$\begin{bmatrix} B_2' & * \\ F_2' & B_1 \\ O & F_1 \end{bmatrix},$$

where B_1 and B_2' are square matrices and have only 1's on their main diagonals, and where F_1 is a nonvacuous matrix with at least one 1 in each column. By Lemma 1.2, M has no minimum cover, which is proper and hence the matrix F_2', if not vacuous, contains at least one 1. Hence we may further permute the rows and columns of M so that M has the form

$$\begin{bmatrix} B_3' & * & * \\ F_3' & B_2 & * \\ O & F_2 & B_1 \\ O & O & F_1 \end{bmatrix},$$

where B_2 and B_3' are square matrices and have only 1's on their main diagonals, and where F_2 is a nonvacuous matrix with at least one 1 in each column. If the matrix F_3' is not vacuous, then it contains at least one 1. We may continue like this and eventually obtain, after row and column permutations, a matrix of the form

$$\begin{bmatrix} B_k & * & * & \cdots & * & * \\ F_k & B_{k-1} & * & \cdots & * & * \\ O & F_{k-1} & B_{k-2} & \cdots & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & \cdots & B_2 & * \\ O & O & O & \cdots & F_2 & B_1 \\ O & O & O & \cdots & O & F_1 \end{bmatrix}, \tag{2}$$

where k is a positive integer, the square matrices $B_1, B_2, \ldots, B_k$ have only 1's on their main diagonals, and the matrices $F_1, F_2, \ldots, F_k$ are nonvacuous matrices with at least one 1 in each column. Conversely, a matrix of the form (2) satisfying these

conditions is a strong Hall matrix. This implies that any matrix M' obtained from M by replacing all the off-diagonal 1's of each B_i with 0's, all but exactly one 1 in each column of each F_i with 0's, and all submatrices marked with a $*$ with zero matrices is a strong Hall matrix. Such a matrix M' has exactly two 1's in each column.

An almost immediate corollary of the form (2) is the following theorem in [5,Thm. 3.1].

COROLLARY 3.1. *Let M be an m by n strong Hall $(0,1)$-matrix with $m > n$. Let k be an integer with $1 \leq p \leq n$ and assume that column p of M contains at least three 1's. Then for all but at most one 1 in column p of M, the matrix obtained from M by replacing the 1 with 0 is a strong Hall matrix.*

Proof. Without loss of generality we assume that M has the form (2). Let M' be any matrix as defined above. Then M' is a strong Hall matrix and hence all but at most two 1's in column p of M have the property stated in the corollary. Consider a 1 in column p of M that is not a 1 of M'. By Lemma 1.2, this 1 belongs to some column diagonal of M. Taking the form (2) with respect to this column diagonal we now see that at most one 1 in column p does not have the property stated in the corollary. □

THEOREM 3.2. *Let M be an m by n strong Hall $(0,1)$-matrix with $m > n$ and assume that the graph $G(M)$ is connected. Then there exists a $(0,1)$-matrix M^* obtained from M by replacing certain 1's by 0's such that M^* is a strong Hall matrix and the graph $G(M^*)$ is a tree.*

Proof. We may assume that M has the form (2), and we let M' be a matrix as defined above. Then M' is a strong Hall matrix and the graph $G(M')$ is a spanning forest of $G(M)$. Since $G(M)$ is connected, it is possible to replace certain 0's of M' by 1's so that the bipartite graph of the resulting matrix M^* is a spanning tree of $G(M)$. Since M' is a strong Hall matrix, so is M^*. □

Theorems 2.2 and 3.2 provide an inductive structure for any nonsquare strong Hall matrix with a connected graph that is quite analogous to that of a square strong Hall (i.e., fully indecomposable) matrix. The role of cycles in the square case is played by paths in the nonsquare case.

Theorems 2.4 and 3.2 immediately imply the following result.

COROLLARY 3.3. *Let M be an m by n strong Hall $(0,1)$-matrix with $m > n$ and assume that $G(M)$ is connected. Then given any two distinct vertices v and w there exist column matchings F' and F'' and a path from v to w in $G(M)$ that is both F'-alternating and $\overline{F''}$-alternating.*

We now obtain two new characterizations of strong Hall matrices whose graphs are connected. The fact that (ii) below is a property of such matrices is equivalent to Lemma 2.5 in [4].

THEOREM 3.4. *Let M be an m by n $(0,1)$-matrix with $m \geq n$ and assume that $G(M)$ is connected. Then the following are equivalent:*

(i) *M is a strong Hall matrix;*

(ii) *Given any column vertex v and row vertex w there exists a column matching F and a path γ from v to w in $G(M)$ that is F-alternating.*

(iii) *Every edge of $G(M)$ is in a column-matching.*

Proof. We first assume that (i) holds and prove that (ii) holds. If $m > n$, then (ii) follows from Corollary 3.3. Now assume that $m = n$. Let M' be the $n+1$ by n matrix obtained from M by appending a new row whose only nonzero entry is a 1 in the column corresponding to v. Then M' is a strong Hall matrix and the graph $G(M')$ is the connected graph obtained from $G(M)$ by adjoining a new row vertex

u and a new edge joining u and v. By Corollary 3.3 there is column-matching F of $G(M')$ and an F-alternating path γ in $G(M')$ from v to w. Since the first edge of γ belongs to F, F is a column-matching of $G(M)$. Since u is a leaf of $G(M')$, γ is a path of $G(M)$. Thus (ii) holds.

We now assume that (ii) holds and show that (iii) also holds. Let α be an edge joining a column vertex v to a row vertex w. Let γ and F be as guaranteed by (ii). If α is the only edge of γ, then α is in F. Otherwise, γ and α form a cycle, and by interchanging matching and nonmatching edges on this cycle, we obtain a column-matching containing α. Hence (iii) holds.

Just as in the proof of Theorem 2.4, (iii) implies (ii). □

The implication (i) implies that (ii) in Theorem 3.4 is equivalent to Lemma 2.5 in [4]. The fact that (i) and (iii) are equivalent has also been noted by Gilbert (private communication).

We conclude with the following result.

THEOREM 3.5. *Let M be an m by n strong Hall matrix with $n \geq 3$. Then there exists an m' by n strong Hall submatrix M' of M with $m' \leq 2n$. If $G(M)$ is connected, then M' can be chosen so that $G(M')$ is connected and $m' \leq 2n - 2$.*

Proof. The first assertion is trivial if m does not exceed $2n$. We may assume that M has the special form (2), where B_1 has order $p \leq n$. There exists a submatrix F_1' of F_1 of order p that has at least one 1 in each of its columns. The matrix obtained from M by deleting the $m - n - p$ rows that intersect F_1, but not F_1', is a strong Hall matrix with $n + p \leq 2n$ rows. The second conclusion of the theorem follows from Theorems 2.5 and 3.2. □

Direct sums of matrices, each of which equals the 2 by 1 matrix of all 1's, show that the first inequality in Theorem 3.5 is best possible.

Acknowledgment. It is a pleasure to thank John Gilbert for a stimulating lecture at the IMA on strong Hall matrices and for helpful comments.

REFERENCES

[1] R. A. BRUALDI, *Term rank of the direct product of matrices*, Canad. J. Math., 18 (1966), pp. 126–138.

[2] R. A. BRUALDI AND H.J. RYSER, *Combinatorial Matrix Theory*, Cambridge University Press, New York, 1991.

[3] T. F. COLEMAN, A. EDENBRANDT, AND J.R. GILBERT, *Predicting fill for sparse orthogonal factorization*, J. Assoc. Comput. Math., 33 (1986), pp. 517–532.

[4] J. R. GILBERT, *An efficient parallel sparse partial pivoting algorithm*, preprint.

[5] D. R. HARE, C.R. JOHNSON, D.D. OLESKY, AND P. VAN DEN DRIESSCHE, *Sparsity analysis of the QR factorization*, SIAM J. Matrix Anal., 14 (1993), pp. 655–669.

[6] L. LOVÁSZ AND M.D. PLUMMER, *Matching Theory*, North-Holland, Amsterdam, 1986.

[7] A. POTHEN, *Predicting the structure of sparse orthogonal factors*, Linear Algebra Appl., 194 (1993), pp. 183–203.

[8] A. POTHEN AND C.-J. FAN, *Computing the block triangular form of a sparse matrix*, ACM Trans. Math. Software, 16 (1990), pp. 303–324.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 366–382, April 1994

002

COMPUTING THE PSVD OF TWO 2×2 TRIANGULAR MATRICES*

GARY E. ADAMS†, ADAM W. BOJANCZYK†, AND FRANKLIN T. LUK‡

Abstract. In this paper, the authors propose a method for computing the singular value decomposition (SVD) of a product of two 2×2 triangular matrices. The method shown is numerically desirable in that all relevant residual elements will be numerically small.

Key words. singular value decomposition, Jacobi methods

AMS subject classifications. 65F15, 15A18, 15A23

1. Introduction.

The problem of computing the SVD of a product of two matrices has many applications; see, e.g., [4] and [5]. The problem is also closely related to finding a generalized SVD of two matrices (cf. [6]). A crucial step in either the product SVD (PSVD) or the generalized SVD (GSVD) problem is the accurate computation of the PSVD of two 2×2 triangular matrices.

We wish to achieve two objectives: first, to ensure that the transformations applied to the triangular matrices must leave the matrices triangular and, second, to ensure that the SVD of the product is computed accurately. As discussed in a recent paper by Bai and Demmel [1], these two properties are essential to guarantee the stability of the GSVD method [6]. Several strategies have been proposed to preserve these two properties. In [1], examples are presented where these strategies can fail and a new method that overcomes the exposed drawbacks is then proposed.

In this paper we propose an alternative approach. Our new method, which we will call a *half-recursive* method, is a slight variation of the *fully recursive* method proposed in [2] for computing the SVD of a product of several matrices. An alternative method for the SVD of several matrices appears in [7]. We show that our algorithm is *simpler* to implement and enjoys the same nice numerical properties as the method in [1].

Our paper is organized as follows. In §2 we describe the PSVD of two 2×2 upper triangular matrices. A criterion for numerical stability is given in §3. We present our new algorithm in §4, and an error analysis in §5. Finally, some detailed proofs can be found in Appendices A and B and a numerical example in Appendix C.

2. Problem definition.

Given two upper triangular matrices:

$$A_1 = \begin{pmatrix} a_1 & b_1 \\ 0 & d_1 \end{pmatrix} \cdot \text{ and } \quad A_2 = \begin{pmatrix} a_2 & b_2 \\ 0 & d_2 \end{pmatrix},$$

we call the product A

$$A = A_1 A_2 \ ,$$

* Received by the editors October 21, 1991; accepted for publication (in revised form) September 22, 1992. G. E. Adams and F. T. Luk were supported in part by the Army Research Office under grant DAAL03-90-G-0104 and the Joint Services Electronics Program contract F49620-90-C-0039 at Cornell University and A. W. Bojanczyk by the Army Research Office grant DAAL03-90-G-0092 and the Joint Services Electronics Program contract F49620-90-C-0039 at Cornell University.

† School of Electrical Engineering, Cornell University, Ithaca, New York 14853 (adams@ee.cornell.edu, adamb@ee.cornell.edu)

‡ Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180 (luk@cs.rpi.edu).

and let

$$A = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix} .$$

Our objective is to find three orthogonal matrices Q_1, Q_2, Q_3, such that

$$A' = Q_1 A Q_3^T = \begin{pmatrix} a' & 0 \\ 0 & d' \end{pmatrix} \tag{2.1}$$

and

$$A'_i = Q_i A_i Q_{i+1}^T = \begin{pmatrix} a'_i & b'_i \\ 0 & d'_i \end{pmatrix} , \tag{2.2}$$

for $i = 1, 2$. Equations (2.1) and (2.2) imply that

$$A' = A'_1 A'_2 .$$

In other words, we would like to find *three* transformations Q_1, Q_2, and Q_3 to zero out *four* elements, namely, the off-diagonal elements of A and the subdiagonal elements of A_1 and A_2. The extra requirement, although mathematically feasible, may cause numerical difficulty if not treated with care; see examples in [1] and [2]. Our goal is to develop an algorithm so that properties (2.1) and (2.2) will be satisfied except for very small numerical errors. In this paper, we use the vector and matrix 2-norms

$$\| \cdot \| = \| \cdot \|_2 .$$

2.1. Relationship with GSVD. The basic step in a GSVD of two 2×2 triangular matrices A_1 and A_2 is to compute the SVD of the product $A_1 \cdot \mathrm{adj}(A_2)$, where adj denotes the adjoint of a matrix. We have

$$\mathrm{adj}(A_2) = \begin{pmatrix} d_2 & -b_2 \\ 0 & a_2 \end{pmatrix} .$$

It is therefore obvious that our two-by-two PSVD method can also be applied to the two-by-two GSVD problem.

3. Criterion for numerical stability. For the purpose of this analysis, an overbar ($\bar{}$) denotes a computed quantity that is perturbed as the result of inexact arithmetic. We assume that exact arithmetic may be performed using these perturbed values. The tilde symbol ($\tilde{}$) is used to denote conceptual values computed exactly from perturbed data. For example, let $\tilde{A}$ be the exactly computed A using quantities $\bar{A}_1$ and $\bar{A}_2$. Let $\bar{Q}_1$, $\bar{Q}_2$, and $\bar{Q}_3$ be transformations computed with floating point arithmetic. Recall that A'_1, A'_2, and A' denote the three matrices A_1, A_2, and A, respectively, after the equivalence transformations as defined in (2.1) and (2.2) have been performed. Define

$$\tilde{A}' := \bar{Q}_1 \bar{A} \bar{Q}_3^T = \begin{pmatrix} \tilde{a}' & \tilde{b}' \\ \tilde{e}' & \tilde{d}' \end{pmatrix} \tag{3.1}$$

and

$$\tilde{A}'_i := \bar{Q}_i A_i \bar{Q}_{i+1}^T = \begin{pmatrix} \tilde{a}'_i & \tilde{b}'_i \\ \tilde{e}'_i & \tilde{d}'_i \end{pmatrix} . \tag{3.2}$$

Let ϵ denote the relative machine precision. The best that we can aim for is to compute $\bar{A}_i'$, such that

$$\| \tilde{A}_i' - A_i' \| = O(\epsilon \ \| A_i \|) . \tag{3.3}$$

The relation (3.3) implies that the (2,1) element $\tilde{e}_i'$ of $\tilde{A}_i'$ will satisfy

$$| \tilde{e}_i' | = O(\epsilon \ \| A_i \|) , \tag{3.4}$$

for $i = 1, 2$. Condition (3.4) implies that e_i' may be safely truncated to zero. Thus, $\tilde{e}'$ is also forced to zero.

We prove in §5 that by using our new method, the computed matrices $\tilde{A}_1'$ and $\tilde{A}_2'$ will satisfy condition (3.4) and $\tilde{A}'$ will satisfy the conditions that

$$|\tilde{b}'| = O(\epsilon \ \| \bar{A} \|) \tag{3.5}$$

and

$$|\tilde{e}'| = O(\epsilon \ \| \bar{A} \|) . \tag{3.6}$$

The conditions proposed in [1] for computing the GSVD of two matrices, A_1 and $\mathrm{adj}(A_2)$, follow from (3.4), (3.5), (3.6), and the similar construction of the two algorithms.

4. New algorithm. In this section, we propose a new algorithm for the PSVD problem. Our algorithm is a modification of the algorithm presented in [2] for a product of several matrices. The tool we use is a transformation discussed in Charlier, Vanbegin, and Van Dooren [3]:

$$Q = \begin{pmatrix} s & c \\ -c & s \end{pmatrix} , \tag{4.1}$$

where $c^2 + s^2 = 1$. We may regard the transformation as a permuted reflection

$$Q = \begin{pmatrix} c & s \\ s & -c \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} .$$

The reason behind using permuted reflections is that we actually deal with an $n \times n$ problem. The permutation that is incorporated into Q corresponds to the so-called odd-even order of eliminations in one sweep of a Jacobi SVD procedure.

While each transformation Q_i is defined by the cosine-sine pair

$$c_i = \cos\theta_i \quad \text{and} \quad s_i = \sin\theta_i ,$$

we also associate Q_i with the tangent

$$t_i = \tan\theta_i .$$

Given t_i, we can easily recover c_i and s_i using the relations

$$c_i = \frac{1}{\sqrt{1 + t_i^2}} \quad \text{and} \quad s_i = t_i c_i . \tag{4.2}$$

Following the exposition in [2], we consider the result of applying the left and right transformations Q_l and Q_r to a 2 × 2 upper triangular matrix A:

$$A' = Q_l A Q_r^T = \begin{pmatrix} a' & b' \\ e' & d' \end{pmatrix} = \begin{pmatrix} s_l & c_l \\ -c_l & s_l \end{pmatrix} \begin{pmatrix} a & b \\ 0 & d \end{pmatrix} \begin{pmatrix} s_r & c_r \\ -c_r & s_r \end{pmatrix}^T . \tag{4.3}$$

We can derive from (4.3) these four relations:

$$e' = c_l c_r(-at_r + dt_l - b) , \tag{4.4}$$

$$b' = c_l c_r(-at_l + dt_r + bt_l t_r) , \tag{4.5}$$

$$a' = c_l c_r(bt_l + d + at_l t_r) , \tag{4.6}$$

$$d' = c_l c_r(a - bt_r + dt_l t_r) , \tag{4.7}$$

where $t_l = \tan\theta_l$ and $t_r = \tan\theta_r$. The postulates that both e' and b' be zeros define two conditions on t_l and t_r, so that (4.3) represents an SVD of A. The postulate that e' be zero defines a condition relating θ_l to θ_r, so that if one is known the other can be computed to reduce A' to an upper triangular form. For ease of exposition, assume for now that $abd \neq 0$. This condition will be removed in §5.2. It implies that $c_l c_r \neq 0$, and so the postulate that $e' = 0$ in (4.4) becomes

$$-at_r + dt_l - b = 0 . \tag{4.8}$$

The consequence of (4.8) is that (4.6) and (4.7) simplify to

$$a' = c_l c_r(t_l^2 + 1)d \tag{4.9}$$

and

$$d' = c_l c_r(t_r^2 + 1)a , \tag{4.10}$$

respectively. The relations (4.9) and (4.10) imply that

$$a'd' = ad .$$

For the SVD problem, both e' and b' are zeros, and we can use (4.8) to reduce (4.5) either to an equation in t_l:

$$b' = c_l c_r\left(\frac{bd}{a}\right)\left(t_l^2 + 2t_l\sigma_l - 1\right) , \tag{4.11}$$

where

$$\sigma_l = \frac{1}{2d}\left(\frac{d^2 - a^2}{b} - b\right)$$

or to an equation in t_r:

$$b' = c_l c_r\left(\frac{ab}{d}\right)\left(t_r^2 + 2t_r\sigma_r - 1\right) , \tag{4.12}$$

where

$$\sigma_r = \frac{1}{2a}\left(\frac{d^2 - a^2}{b} + b\right).$$

From (4.11) we get a quadratic equation by setting b' to zero:

$$t_l^2 + 2\sigma_l t_l - 1 = 0 \tag{4.13}$$

and from (4.12) we get

$$t_r^2 + 2\sigma_r t_r - 1 = 0\,. \tag{4.14}$$

Equations (4.13) and (4.14) are solved by the formulas given in [2]:

$$r = \frac{(d-a)(d+a)}{b}\,, \tag{4.15}$$

$$\sigma_l = \frac{r-b}{2d}\,, \tag{4.16}$$

$$\sigma_r = \frac{r+b}{2a}\,, \tag{4.17}$$

$$t_l = \frac{1}{\sigma_l + \operatorname{sign}(\sigma_l)\sqrt{\sigma_l^2 + 1}}\,, \tag{4.18}$$

$$t_r = \frac{1}{\sigma_r + \operatorname{sign}(\sigma_r)\sqrt{\sigma_r^2 + 1}}\,. \tag{4.19}$$

In finite-precision arithmetic, either one of t_l and t_r can be computed with a higher relative precision. In particular, if

$$\operatorname{sign}(r) = -\operatorname{sign}(b)\,,$$

then (4.18) will produce a very accurate t_l; whereas if

$$\operatorname{sign}(r) = \operatorname{sign}(b)\,,$$

then (4.19) will produce a very precise t_r. If $r = 0$, then both t_l and t_r will be computed with the same relative accuracy.

Now, let $r \neq 0$. We first present a lemma relating the sizes of t_l and t_r to those of a and d.

LEMMA 4.1. *Let abdr* $\neq 0$. *If* $|\, a \,| > |\, d \,|$, *then* $|\, \sigma_l \,| > |\, \sigma_r \,|$ *and* $|\, t_l \,| < |\, t_r \,|$. *Conversely, if* $|\, a \,| < |\, d \,|$, *then* $|\, \sigma_l \,| < |\, \sigma_r \,|$ *and* $|\, t_l \,| > |\, t_r \,|$.

Proof. See [2]. □

We are ready to present an algorithm for computing the three orthogonal matrices Q_1, Q_2, and Q_3, such that (2.1) and (2.2) are satisfied. The algorithm proceeds in two stages. In the first stage, we calculate the product A explicitly:

$$a = a_1 a_2\,, \tag{4.20}$$

$$b = a_1 b_2 + b_1 d_2 \,, \tag{4.21}$$

$$d = d_1 d_2 \,. \tag{4.22}$$

We use (4.15) to calculate r, and then we compute either σ_l or σ_r so that the corresponding tangent defines the smaller angular rotation. Hence we obtain either t_1 or t_3. In the second stage, we use the relation (4.8) with t_1 or t_3 as the reference tangent to compute the remaining transformations. Suppose that t_1 is known, then t_2 and t_3 are generated by the forward substitutions

$$t_2 = \frac{d_1 t_1 - b_1}{a_1} \,, \tag{4.23}$$

$$t_3 = \frac{d t_1 - b}{a} \,. \tag{4.24}$$

On the other hand, if t_3 is known, then t_2 and t_1 are generated by the backward substitutions

$$t_2 = \frac{a_2 t_3 + b_2}{d_2} \,, \tag{4.25}$$

$$t_1 = \frac{a t_3 + b}{d} \,. \tag{4.26}$$

If t_1 is computed first as the reference tangent, then (4.23) will guarantee that A_1' will be numerically upper triangular and (4.24) will guarantee that A' will be numerically diagonal. As will be shown later, these two properties will guarantee that A_2' will be numerically upper triangular and hence (3.4), (3.5), and (3.6) will be satisfied.

We refer to the method defined by (4.23)–(4.24) or (4.25)–(4.26) as *half recursive*, to differentiate it from the *fully recursive* method proposed in [2] for computing the PSVD of several matrices. The fully recursive method also picks the smaller outer angular rotation as the starting point for the recursion, from which all remaining rotations are computed. However in [2], the other outer rotation is computed from the previous rotation in the sequence. For example, in the case of a product of two matrices, the tangent t_3 in (4.24) would be computed from t_2 using (4.8):

$$t_3 = \frac{d_2 t_2 - b_2}{a_2} \,. \tag{4.27}$$

Note how (4.24) uses the product A whereas (4.27) uses the matrix A_2. It was shown in [1] that the fully recursive method may fail to satisfy (3.5) and (3.6) and thus is not recommended for the GSVD problem. On the other hand, the fully recursive method easily extends to any number of factors in the product. It is not clear what is an appropriate extension of the half-recursive method for the case of a product of more than two matrices.

Our half-recursive method is equivalent to the method proposed by Bai and Demmel in [1] in the sense that it also computes a very accurate PSVD of $A_1 A_2$, and that it uses essentially the same criterion in choosing whether to compute the middle transformation Q_2 from Q_1 or from Q_3. A proof that the two methods use the same condition for computing Q_2 is given in Appendix B.

5. Backward error analysis. In this section, we present a backward error analysis of our computation. The function $\mathrm{fl}(a)$ will be used to denote the floating point approximation of a. For example, instead of a, b, and d, we have the perturbed values $\bar{a}$, $\bar{b}$, and $\bar{d}$, which result from floating point computation, $\mathrm{fl}(\ A_1A_2)$. Recall that a tilde denotes a quantity computed exactly from perturbed data. For example, $\tilde{r}$ denotes the result of using formula (4.15) in exact arithmetic with the perturbed data $\bar{a}$, $\bar{b}$, and $\bar{d}$.

In our error analysis, we adopt a convention that involves a liberal use of Greek letters. For example, by α we mean a relative perturbation of an absolute magnitude not greater than ϵ, where ϵ denotes the machine precision. All terms of order ϵ^2 or higher will be ignored in this first-order analysis.

We start our procedure by computing elements of the product matrix A:

$$\bar{a} := \mathrm{fl}(a_1a_2) = a_1a_2(1+\alpha)\ , \tag{5.1}$$

$$\bar{d} := \mathrm{fl}(d_1d_2) = d_1d_2(1+\delta)\ , \tag{5.2}$$

$$\bar{b} := \mathrm{fl}(a_1b_2 + b_1d_2) = a_1b_2(1+2\beta_1) + b_1d_2(1+2\beta_2)\ , \tag{5.3}$$

where, according to our convention, the parameters α_1, δ_1, β_1, β_2, and β_3 are all quantities whose absolute values are bounded by ϵ. From (5.1)–(5.3) it follows that

$$\bar{A} = (A_1 + \delta A_1)(A_2 + \delta A_2)\ ,$$

with $\| \delta A_i \| \leq \epsilon \ \| A_i \|$. This property, which generally does not hold for a product of more than two 2×2 upper triangular matrices, will allow us to prove backward error type assertions on the half-recursive method.

Our analysis is divided into two parts. In §5.1, we consider a regular case where all elements of the computed matrix product are numerically significant with respect to the maximal-in-magnitude element; i.e.,

$$\min(|\ \bar{a}\ |\ ,\ |\ \bar{b}\ |\ ,\ |\ \bar{d}\ |\ ) > \epsilon\ \max(|\ \bar{a}\ |\ ,|\ \bar{b}\ |\ ,|\ \bar{d}\ |\ )\ . \tag{5.4}$$

In §5.2, we consider special cases where at least one element of the computed A is numerically insignificant.

5.1. Regular case. Without loss of generality, we assume that $rb < 0$; i.e., $\mathrm{sign}(r) = -\mathrm{sign}(b)$. Thus we compute t_1 first as the reference tangent from which t_2 and t_3 will be next determined via (4.23) and (4.24), respectively. We recall several lemmas from [2].

LEMMA 5.1. *Let $\tilde{t}_1$ and $\bar{t}_1$ be the exact and computed solutions, respectively, of* (4.18) *with data $\bar{a}, \bar{b}, \bar{d}$. Moreover, let $\tilde{c}_1, \tilde{s}_1$ and $\bar{c}_1, \bar{s}_1$ be the exact and computed cosines and sines using* (4.2) *with the tangent value $\bar{t}_1$. Then*

$$\bar{t}_1 = \tilde{t}_1(1 + 10\epsilon_1)\ , \tag{5.5}$$

$$\bar{c}_1 = \tilde{c}_1(1 + 3\mu_1)\ , \tag{5.6}$$

$$\bar{s}_1 = \tilde{s}_1(1 + 4\nu_1)\ , \tag{5.7}$$

where $|\epsilon_1| < \epsilon$, $|\mu_1| < \epsilon$, *and* $|\nu_1| < \epsilon$.

Proof. See [2]. □

In other words, Lemma 5.1 states that the procedure (4.15)–(4.19) for solving (4.13) is numerically stable in the forward sense. Two lemmas follow, leading to our main result of Theorem 5.5.

LEMMA 5.2. *The recurrences* (4.23) *and* (4.24) *yield* $\bar{t}_2$ *and* $\bar{t}_3$, *such that*

$$\tilde{a}_1\bar{t}_2 - \tilde{d}_1\bar{t}_1 + b_1 = 0\,, \tag{5.8}$$

$$\tilde{a}\bar{t}_3 - \tilde{d}\bar{t}_1 + \bar{b} = 0\,, \tag{5.9}$$

with

$$\tilde{a}_1 = a_1(1+2\psi_1)\,, \qquad \tilde{d}_1 = d_1(1+\phi_1)\,, \tag{5.10}$$

$$\tilde{a} = \bar{a}(1+2\psi)\,, \qquad \tilde{d} = \bar{d}(1+\phi). \tag{5.11}$$

Proof. The proof easily follows from (4.23) and (4.24). □

LEMMA 5.3. *The recurrence* (4.24) *yields* $\bar{t}_3$, *such that* $\bar{t}_3 = \tilde{t}_3(1+13\gamma)$.

Proof. From (4.24), it holds that

$$\begin{aligned}
\bar{t}_3 &= \left(\frac{\bar{d}\tilde{t}_1(1+11\psi) - \bar{b}}{\bar{a}}\right)(1+2\gamma_1) \\
&= \left(\frac{\bar{d}\tilde{t}_1 - \bar{b}}{\bar{a}} + \frac{11\psi\bar{d}\tilde{t}_1}{\bar{a}}\right)(1+2\gamma_1) \\
&= \left(\tilde{t}_3 + 11\psi\tilde{t}_3\frac{\bar{d}}{\bar{a}}\frac{\tilde{t}_1}{\tilde{t}_3}\right)(1+2\gamma_1)\,.
\end{aligned}$$

Since $|\bar{d}/\bar{a}| \le 1$ and $|\tilde{t}_1/\tilde{t}_3| \le 1$, we get $\bar{t}_3 = \tilde{t}_3(1+13\gamma)$. □

We now show that $\bar{a}'$ and $\bar{d}'$ are computed with high relative precision.

LEMMA 5.4. *Let* $\tilde{a}'$ *and* $\tilde{d}'$ *be the exact singular values of the computed product* $\bar{A}$. *If* $\tilde{a}'$ *and* $\tilde{d}'$ *are computed via relations* (4.6) *and* (4.7), *then the computed singular values* $\bar{a}'$ *and* $\bar{d}'$ *satisfy the following relations*:

$$\bar{a}' = \tilde{a}'(1+\alpha_4)\,, \qquad \bar{d}' = \tilde{d}'(1+\delta_4)\,. \tag{5.12}$$

Proof. From (4.9) and (4.10), we get

$$\tilde{a}' = \bar{d}(\tilde{t}_1^2+1)\hat{c}_1\hat{c}_3 \quad \text{and} \quad \tilde{d}' = \bar{a}(\tilde{t}_3^2+1)\hat{c}_1\hat{c}_3\,,$$

where $\tilde{t}_1$ and $\tilde{t}_3$ are the exact tangents corresponding to the data $\bar{a}$, $\bar{b}$, and $\bar{d}$, and $\tilde{t}_i = \hat{s}_i/\hat{c}_i$. Thus, the lemma follows from Lemmas 5.1 and 5.3. □

THEOREM 5.5. *Suppose that the computed tangent values are* $\bar{t}_1$ *and* $\bar{t}_3$. *Let* $\tilde{c}_1$, $\tilde{s}_1$, $\tilde{c}_3$, *and* $\tilde{s}_3$ *be the corresponding exact cosine and sine values. Let*

$$\tilde{e}' := \tilde{c}_1\tilde{c}_3[-\bar{a}\bar{t}_3 + \bar{d}\bar{t}_1 - \bar{b}]\,, \tag{5.13}$$

$$\tilde{b}' := \tilde{c}_1\tilde{c}_3[-\bar{a}\bar{t}_1 + \bar{d}\bar{t}_3 + \bar{b}\bar{t}_1\bar{t}_3]\,. \tag{5.14}$$

That is, $\tilde{e}'$ and $\tilde{b}'$ are the exact values of e' and b', respectively, corresponding to the computed data $\bar{a}$, $\bar{b}$, $\bar{d}$, $\bar{t}_1$, and $\bar{t}_3$. Then

$$| \tilde{e}' | \leq K_1 \epsilon \ \| \bar{A} \| \ , \tag{5.15}$$

$$| \tilde{b}' | \leq K_2 \epsilon \ \| \bar{A} \| \ , \tag{5.16}$$

where K_1 and K_2 are some positive constants.

Proof. See Appendix A. □

Lemma 5.4 and Theorem 5.5 together state that the SVD of the upper triangular matrix $\bar{A}$ is computed very accurately. We now justify why the (2,1) element in the computed matrix A_i' can be set to zero by showing that $| \tilde{e}_i' |$ corresponds to a relative and elementwise perturbation of A_i' of the order of ϵ. Let the cosine and sine pairs $\tilde{c}_i$ and $\tilde{s}_i$ satisfy $\bar{t}_i = \tilde{s}_i / \tilde{c}_i$ for $i = 1, 2, 3$. From (4.2) we can derive that

$$\bar{c}_i := \mathrm{fl}(\tilde{c}_i) = \tilde{c}_i(1 + 3\mu_i) \ , \tag{5.17}$$

$$\bar{s}_i := \mathrm{fl}(\tilde{s}_i) = \tilde{s}_i(1 + 4\nu_i) \ . \tag{5.18}$$

Let $\tilde{A}_i'$ denote the exact updated matrix derived from A_i, $\bar{c}_i$, $\bar{s}_i$, $\bar{c}_{i+1}$, and $\bar{s}_{i+1}$. Our next results provide a bound on the element $\tilde{e}_i'$, $i = 1, 2$, defined by the relation

$$\tilde{e}_i' := -\bar{c}_i \bar{s}_{i+1} a_i + \bar{s}_i \bar{c}_{i+1} d_i - \bar{c}_i \bar{c}_{i+1} b_i \ . \tag{5.19}$$

THEOREM 5.6. *The matrices $\tilde{A}_1'$ and $\tilde{A}_2'$ are almost upper triangular in that their* (2.1) *elements $\tilde{e}_1'$ and $\tilde{e}_1'$ satisfy the inequalities*

$$| \tilde{e}_1' | \ \leq 3 \ \epsilon \ \| A_1 \| \tag{5.20}$$

and

$$| \tilde{e}_2' | \ \leq K_3 \ \epsilon \ \| A_2 \| \ . \tag{5.21}$$

Proof. Note that $\tilde{A}_1'$ is the same for both fully recursive and half-recursive methods. The proof that $\tilde{A}_1'$ is almost upper triangular in the sense that (5.20) holds can be found in [2].

To prove the second part of the theorem from (5.8)–(5.11) and (5.1)–(5.3), we get the following two relations to first order of the machine precision:

$$a_1(1 + 2\psi_1)\bar{t}_2 - d_1(1 + \phi_1)\bar{t}_1 + b_1 = 0 \ , \tag{5.22}$$

$$\begin{aligned} a_1 a_2(1 + \alpha + 2\psi)\bar{t}_3 - d_1 d_2(1 + \delta + \phi)\bar{t}_1 + a_1 b_2(1 + 2\beta_1) \\ + b_1 d_2(1 + 2\beta_2) = 0 \ . \end{aligned} \tag{5.23}$$

By multiplying both sides of (5.22) by $d_2(1 + 2\beta_2)$ and subtracting from (5.23), we obtain

$$\begin{aligned} a_1 \Bigg\{ a_2(1 + \alpha + 2\psi)\bar{t}_3 - \left(\frac{d_1 d_2}{a_1} \right) \\ \times (\delta + \phi - \phi_1 + 2\beta_2)\bar{t}_1 + b_2(1 + 2\beta_1) - d_2(1 + 2\beta_2 + 2\psi_1)\bar{t}_2 \Bigg\} = 0 \ , \end{aligned}$$

or, since $a_1 \neq 0$,

$$a_2(1+\alpha+2\psi)\bar{t}_3 - \left(\frac{d_1 d_2}{a_1}\right)(\delta+\phi-\phi_1+2\beta_2)\bar{t}_1 + b_2(1+2\beta_1) - d_2(1+2\beta_2+2\psi_1)\bar{t}_2 = a_2\bar{t}_3 - d_2\bar{t}_2 + b_2 + \Delta = 0\,,$$

where

$$\Delta = a_2(\alpha+2\psi)\bar{t}_3 - \left(\frac{d}{a}\right)a_2(\delta+\phi-\phi_1+2\beta_2)\bar{t}_1 + b_2\beta_1 - d_2(2\beta_2+2\psi_1)\bar{t}_2\,.$$

Thus, we can rewrite (5.19) for $i = 2$ as

$$\tilde{e}_2' = -\bar{c}_2\bar{s}_3 a_2 + \bar{s}_2\bar{c}_3 d_2 - \bar{c}_2\bar{c}_3 b_2 + \tilde{c}_3\tilde{c}_2(a_2\bar{t}_3 - d_2\bar{t}_2 + b_2 + \Delta)\,. \tag{5.24}$$

Now, as we start the half-recursive method from t_1, it means that $|\bar{t}_1| \leq |\bar{t}_3|$ and $|\bar{d}| \leq |\bar{a}|$. Hence from (5.17), (5.18), and (5.24), we derive the inequality:

$$\begin{aligned}|\tilde{e}_2'| &\leq |\tilde{s}_3\tilde{c}_2 a_2(\alpha+2\psi)| + |\tilde{c}_3\tilde{c}_2 a_2(\delta+\phi-\phi_1+2\beta_2)| \\ &\quad + |\tilde{c}_3\tilde{c}_2 b_2\beta_2| + |\tilde{c}_3\tilde{s}_2 d_2(2\beta_2+2\psi_1)| \\ &\leq K_3\epsilon||A_2||\,,\end{aligned}$$

completing the proof. □

In summary, we have proved two results using backward error analysis. First, the transformed matrix $\bar{A}'$ is almost diagonal in that inequalities (5.15) and (5.16) both hold. Second, we can safely set each computed matrix $\bar{A}_i'$, $i = 1, 2$, to a triangular form because (5.20) and (5.21) are valid. As a final note, even though we have assumed that $rb < 0$, we can easily prove similar results for the case where $rb \geq 0$.

5.2. Special cases. In this subsection, we assume that inequality (5.4) is violated. To be specific, define

$$\gamma := \min\,(|\,\bar{a}\,|\,,\ |\,\bar{b}\,|\,,\ |\,\bar{d}\,|\,) \tag{5.25}$$

and

$$\Gamma := \max\,(|\,\bar{a}\,|\,,\ |\,\bar{b}\,|\,,\ |\,\bar{d}\,|\,)\,. \tag{5.26}$$

Now,

$$\gamma \leq \epsilon\,\Gamma\,; \tag{5.27}$$

i.e., one of the elements of $\bar{A}$ is numerically insignificant. This situation requires modifications to our algorithm, since the proposed formulas may break down. In particular, we do not solve a quadratic equation to determine either $\bar{t}_1$ or $\bar{t}_3$. Instead, we set one of the two tangents to zero and attempt to compute all the other tangents from the recurrences. We divide the special cases into three groups: first,

$$|\,\bar{a}\,| + |\,\bar{d}\,| \neq 0 \quad \text{and} \quad |\,\bar{b}\,| \neq 0\,; \tag{5.28}$$

second,

$$|\,\bar{a}\,| + |\,\bar{d}\,| = 0 \quad \text{and} \quad |\,\bar{b}\,| \neq 0\,; \tag{5.29}$$

and third,

$$|\bar{b}| = 0\,. \tag{5.30}$$

First, assume that (5.28) holds. Hence at least one, but not all, of the following three conditions hold:

$$\gamma = \bar{b}, \quad \gamma = \bar{a}, \quad \text{or} \quad \gamma = \bar{d}\,.$$

We set $\bar{t}_1$ to zero if

$$|\bar{a}| > |\bar{d}|\,, \tag{5.31}$$

and set $\bar{t}_3$ to zero if

$$|\bar{a}| \leq |\bar{d}|\,. \tag{5.32}$$

Thus, the sizes of the diagonal elements of $\bar{A}$ will be compared to determine which one of $\bar{t}_1$ or $\bar{t}_3$ should be zeroed. Without loss of generality, assume that (5.31) holds; hence, $\bar{t}_1$ becomes the reference angle. So, $\bar{t}_2$ and $\bar{t}_3$ are computed from recurrence (4.23) and (4.24). Furthermore, since $\bar{t}_1 = 0$, it follows that $\bar{t}_3 = -\bar{b}/\bar{a}$. Substituting these values into (5.22)–(5.25), we can verify that Theorem 5.5 holds. Similarly, Theorem 5.6 follows from (5.19). We note that it is very important to decide which reference angle to choose, even for the case when $\bar{b}$ is numerically zero. At first, the choice of the reference angle may seem arbitrary for a "small" $\bar{b}$, since either $\bar{t}_1$ or $\bar{t}_3$ can be set to zero. However, an unnecessarily large error may occur unless we pay special care.

Second, assume that (5.29) holds. Then, at least one of the a_i's equals zero and at least one of the d_j's also equals zero, for $i, j = 1, 2$. A solution is to permute either the rows or the columns to ensure that the transformed product is diagonal and that the data are reordered. Hence for this case, we may set the two extreme tangents $\{\bar{t}_1, \bar{t}_3\}$ to $\{0, \infty\}$, resulting in the transformations being rotations of negative ninety and zero degrees, respectively. To be specific, consider the case where one or more a_i's equal zero. If $a_1 = 0$, set $\bar{t}_1 = 0$ and $\bar{t}_2 = \bar{t}_3 = \infty$. If $a_1 \neq 0$ and $a_2 = 0$, set $\bar{t}_1 = 0$, compute $\bar{t}_2$ from the forward recurrence and set $\bar{t}_3 = \infty$. Note that we may also choose to determine the tangents using the values of the d_j's.

Third, assume that (5.30) holds. We need to account for the fact that we are really solving an $n \times n$ problem. Although the 2×2 subproblem is already numerically diagonal, it is not sufficient to set $\bar{t}_1 = \bar{t}_3 = \infty$, which will leave the 2×2 product unchanged. The $n \times n$ data need to be reordered, calling for $\bar{t}_1 = \bar{t}_3 = 0$, i.e., the affected rows and columns will be permuted. Unfortunately, while applying the symmetric permutation, the triangular structures of both $\bar{A}_1$ and $\bar{A}_2$ are destroyed. Therefore, $\bar{t}_2$ is determined from the recurrence.

6. Concluding remark. In this paper we have presented a simple and accurate way to calculate the PSVD or GSVD of two 2×2 upper triangular matrices. In Appendix C, we present an example that shows that our half-recursive method produces identical numerical results as the method in [1]. A significant issue in the design of PSVD algorithms is how to compute the middle transformation. The method used in our half-recursive algorithm is computationally more efficient than the method in [1] and yields identical results. The following table lists the number of floating point operations used to compute the three transformations, Q_1, Q_2, and Q_3, for the three

different algorithms in the regular case. The column labeled Simplified direct lists the operation count for the Bai and Demmel algorithm if our simplified method is substituted for their method of computing the middle transformation.

TABLE 1
Floating point operation counts.

	Direct	Simplified direct	Half-recursive
Addition	20	13	11
Multiplication	25	17	15
Division	13	11	8
Square root	4	4	4

The half-recursive method is less expensive than the direct method and similar in cost to the simplified direct algorithm. In addition, the upper-triangular structure of the 2×2 matrices is maintained by the half-recursive method. Application of the 2×2 half-recursive algorithm to $n \times n$ problems is a topic for further investigation.

Appendix A. Proof of Theorem 5.5. We first present a lemma.

LEMMA A.1. *Let $\tilde{\sigma}_1$ and $\tilde{t}_1$ be the exact values corresponding to the given data $\bar{a}$, $\bar{b}$, and $\bar{d}$, and let $\bar{t}_1$ be the computed value of $\tilde{t}_1$. Define a residual r_1 by*

$$r_1 := \frac{\bar{b}\bar{d}}{\bar{a}}(\bar{t}_1^2 + 2\tilde{\sigma}_1\bar{t}_1 - 1) . \tag{A.1}$$

Then

$$| r_1 | \leq K_4\epsilon|\bar{b}| , \tag{A.2}$$

where K_4 is a positive constant.

Proof. See the proof of Lemma 5.2 in [2]. □

We now have the necessary tools for proving Theorem 5.5.

Proof. First, from Lemma 5.2 and relation (5.9) we get

$$\tilde{e}' = \tilde{c}_1\tilde{c}_3[(-\bar{a}\bar{t}_3 + \bar{d}\bar{t}_1 - \bar{b}) + (\tilde{a}\bar{t}_3 - \tilde{d}\bar{t}_1 + \bar{b})] = (\tilde{a} - \bar{a})\tilde{c}_1\tilde{s}_3 - (\tilde{d} - \bar{d})\tilde{s}_1\tilde{c}_3 .$$

Using (5.1)–(5.2) and (5.11), we prove the inequality:

$$| \tilde{e}' | \leq K\epsilon (| a | + | d |) \leq K_1\epsilon \| \bar{A} \| . \tag{A.3}$$

Second, rewrite (A.1) as

$$r_1 = \frac{1}{\bar{a}}[\bar{d}\bar{b}\bar{t}_1^2 + \bar{t}_1(\bar{d}^2 - \bar{a}^2 - \bar{b}^2) - \bar{d}\bar{b}] = \frac{1}{\bar{a}}[(\bar{d}\bar{t}_1 - \bar{b})(\bar{b}\bar{t}_1 + \bar{d}) - \bar{t}_1\bar{a}^2] . \tag{A.4}$$

From (5.22)–(5.24), we obtain

$$\frac{1}{\bar{a}}(\bar{d}\bar{t}_1 - \bar{b}) = \bar{t}_3 + \frac{\tilde{e}'}{\tilde{c}_1\tilde{c}_3\bar{a}} . \tag{A.5}$$

Substituting (A.5) into (A.4) and rearranging terms, we get

$$-\bar{a}\bar{t}_1 + \bar{d}\bar{t}_3 + \bar{b}\bar{t}_1\bar{t}_3 = r_1 - \frac{\tilde{e}'(\bar{b}\bar{t}_1 + \bar{d})}{\tilde{c}_1\tilde{c}_3\bar{a}} ,$$

and so

$$\tilde{b}' = \tilde{c}_1\tilde{c}_3 r_1 - \frac{\tilde{e}'(\bar{b}\bar{t}_1 + \bar{d})}{\bar{a}} \,. \tag{A.6}$$

From (4.18), we derive

$$| \tilde{t}_1\tilde{\sigma}_1 | \leq \frac{1}{2} \,,$$

and from (4.16), we get

$$| \tilde{\sigma}_1 | = \left| \frac{\tilde{r} - \bar{b}}{2\bar{d}} \right| \geq \left| \frac{\bar{b}}{2\bar{d}} \right| .$$

It follows that

$$| \tilde{t}_1 | \leq \left| \frac{\bar{d}}{\bar{b}} \right| < \left| \frac{\bar{a}}{\bar{b}} \right| \,, \tag{A.7}$$

since we have assumed that $| \bar{d} | < | \bar{a} |$. Finally, recall from (5.3) that $\bar{t}_1 = \tilde{t}_1(1+10\epsilon_5)$, and use (A.6), Lemma A.1 and (A.5) to obtain

$$| \tilde{b}' | \leq \tilde{c}_1\tilde{c}_2 | r_1 | + 2 | \tilde{e}' | \leq K_2\epsilon \, \| \bar{A} \| \,, \tag{A.8}$$

thus completing the proof. □

Appendix B. How to compute the middle transformation. As pointed out by Bai and Demmel in [1], a critical issue concerns how the middle transformation should be computed. They proposed the following scheme for its computation after both end transformations have been determined. To relate the test for computing Q_2 in [1] to the test in the half-recursive method, we first translate our setting to that in [1]. Let

$$U^T \equiv \begin{pmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{pmatrix}, \quad Q^T \equiv \begin{pmatrix} c_2 & -s_2 \\ s_2 & c_2 \end{pmatrix}, \quad \text{and} \quad V^T \equiv \begin{pmatrix} c_3 & -s_3 \\ s_3 & c_3 \end{pmatrix}.$$

Note that the relation, given by

$$Q_1 A_1 = \begin{pmatrix} s_1 & c_1 \\ -c_1 & s_1 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \\ 0 & d_1 \end{pmatrix} = \begin{pmatrix} s_1 a_1 & s_1 b_1 + c_1 d_1 \\ -c_1 a_1 & -c_1 b_1 + s_1 d_1 \end{pmatrix} \tag{B.1}$$

upon permuting rows and changing the signs of the top row, is equivalent to

$$U^T A_1 = \begin{pmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \\ 0 & d_1 \end{pmatrix} = \begin{pmatrix} c_1 a_1 & c_1 b_1 - s_1 d_1 \\ s_1 a_1 & s_1 b_1 + c_1 d_1 \end{pmatrix} \equiv G \,. \tag{B.2}$$

Similarly,

$$A_2 Q_3^T = \begin{pmatrix} a_2 & b_2 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} s_3 & -c_3 \\ c_3 & s_3 \end{pmatrix} = \begin{pmatrix} s_3 a_2 + c_3 b_2 & -c_3 a_2 + s_3 b_2 \\ c_3 d_2 & s_3 d_2 \end{pmatrix} \,. \tag{B.3}$$

By changing the sign of the second columns and permuting columns, we obtain

$$V^T \mathrm{adj}(A_2) = \begin{pmatrix} c_3 & -s_3 \\ s_3 & c_3 \end{pmatrix} \begin{pmatrix} d_2 & -b_2 \\ 0 & a_2 \end{pmatrix} = \begin{pmatrix} c_3 d_2 & -c_3 b_2 - s_3 a_2 \\ s_3 d_2 & -s_3 b_2 + c_3 a_2 \end{pmatrix} \equiv H. \tag{B.4}$$

In [1], Bai and Demmel used (B.2) and (B.4) as a starting point for computing Q_2. Their argument is as follows. After postmultiplications of both (B.2) and (B.4) by Q_2, the (1,2) elements of G and H should become zeros. Now, one should compute Q_2 from the one product, either G or H, for which the computed element in the (1,2) position has a smaller error relative to the norm of the row in which it resides. The magnitude of that error can be only bounded and hence the test for the choice is based on the bounds of the errors. It is easy to see that the bound g for the relative error in the (1,2) element of the computed G is

$$g = \frac{|c_1 b_1| + |s_1 d_1|}{|c_1 a_1| + |c_1 b_1 - s_1 d_1|}, \tag{B.5}$$

while the bound h for the relative error in the (1,2) element of the computed H is

$$h = \frac{|c_3 b_2| + |s_3 a_2|}{|c_3 d_2| + |c_3 b_2 + s_3 a_2|}. \tag{B.6}$$

Now if $g \leq h$, then Bai and Demmel compute Q_2 from $U^T A$ and otherwise from $V^T B$. The next lemma shows that the conditions specifying how Q_2 is computed by Bai and Demmel and by the half-recursive method are essentially equivalent.

LEMMA B.1. *In exact arithmetic, the condition*

$$g \leq h \tag{B.7}$$

where g is defined by (B.5) *and h is defined by* (B.6) *is equivalent to the condition*

$$|a| \geq |d|. \tag{B.8}$$

Proof. First note that (B.5) and (B.6) can be simplified to

$$g = \frac{|b_1| + |t_1 d_1|}{|a_1| + |t_1 d_1 - b_1|} \tag{B.9}$$

and

$$h = \frac{|b_2| + |t_3 a_2|}{|d_2| + |t_3 a_2 + b_2|}, \tag{B.10}$$

respectively. Through (4.23) and (4.25), the relations (B.5) and (B.6) simplify further to

$$g = \frac{|b_1| + |t_1 d_1|}{|a_1|(1 + |t_2|)} \tag{B.11}$$

and

$$h = \frac{|b_2| + |t_3 a_2|}{|d_2|(1 + |t_2|)}, \tag{B.12}$$

respectively. Hence (B.7) is equivalent to

$$|b_1 d_2| + |t_1 d| \leq |a_1 b_2| + |a t_3| \,. \tag{B.13}$$

We now prove that (B.8) implies (B.7). The proof that $|a| < |d|$ implies that $g < h$ is analogous and is omitted. Our proof is elementary but tedious as it requires us to consider a large number of cases. Assume that $|a| \geq |b|$. Then Lemma 4.1 implies that $t_3 \geq t_1$. From (4.24) we see that

$$|a t_3 + b| = |d t_1| \,,$$

and as $|a t_3| \geq |a t_1|$, we conclude that

$$\operatorname{sign}(a t_3) = -\operatorname{sign}(b) = -\operatorname{sign}(a_1 b_2 + b_1 d_2) \,, \tag{B.14}$$

as from (4.21) $b = a_1 b_2 + b_1 d_2$. Substituting (4.24) into (B.13) and using (4.21) again, we get that (B.13) is equivalent to the following inequality:

$$|b_1 d_2| + |a t_3 + a_1 b_2 + b_1 d_2| \leq |a_1 b_2| + |a t_3| \,. \tag{B.15}$$

Case 1. $-|b| \geq |b_1 d_2| - |a_1 b_2|$. Then

$$|a t_3| \geq |d t_1| - |b| \geq |d t_1| + |b_1 d_2| - |a_1 b_2| \,,$$

establishing (B.13).

Case 2a. $-|b| > |b_1 d_2| - |a_1 b_2|$ and $|a t_3| > |b|$. Then $|a_1 b_2| > |b_1 d_2|$ and using (B.8) we obtain that

$$|b_1 d_2| + |d t_1| = |b_1 d_2| + |a t_3 + a_1 b_2 + b_1 d_2| = |a t_3| + 2\,|b_1 d_2| - |a_1 b_2| \,,$$

from which (B.13) follows.

Case 2b. $-|b| > |b_1 d_2| - |a_1 b_2|$ and $|a t_3| \leq |b|$. Then again $|a_1 b_2| > |b_1 d_2|$. Now from (B.14)

$$\begin{aligned} |b_1 d_2| + |d t_1| &= |b_1 d_2| + |a t_3 + a_1 b_2 + b_1 d_2| \\ &= |b_1 d_2| - |a t_3| + |a_1 b_2| - |b_1 d_2| = |a_1 b_2| - |a t_3| \,, \end{aligned}$$

from which (B.13) again follows. □

Remark. Note that there might be a slight difference in using (B.7) or (B.8) as the lemma holds only in exact arithmetic. In finite precision computation, the relations (B.7) and (B.8) may not always be equivalent. However, we have not been able to find any numerical example where these two conditions are not equivalent. Moreover, as shown in this paper, the consequences of numerical nonequivalence are numerically insignificant.

Appendix C. Numerical example. It has been proved in Appendix B that the half-recursive procedure computes essentially the same numerical results as the direct method of [1]. For both methods, the end transformations are computed explicitly from the product $A = A_1 A_2$, and the middle transformation is computed from the same direction. The greatest difference between the fully recursive method and the other two occurs when there is cancellation in forming the product $A = A_1 A_2$. In the

following PSVD example, A_1 and A_2 each has an $O(1)$ norm, but the product A_1A_2 has an $O(10^{-5})$ norm. Hence errors which are small relative to the initial matrices may be large relative to the product.

$$A_1 = \begin{pmatrix} 2.316797292247488e+00 & -1.437687878748196e-01 \\ 0 & -5.208536329107726e-06 \end{pmatrix},$$
$$A_2 = \begin{pmatrix} 2.472499811756353e-05 & 2.624474233535929e-01 \\ 0 & 4.229273187671001e+00 \end{pmatrix},$$
$$A_1A_2 = \begin{pmatrix} 5.728280868959543e-05 & -1.110223024625157e-16 \\ 0 & -2.202832304370565e-05 \end{pmatrix}.$$

The three methods all compute the left transformation from the explicit product and calculate the middle transformation from A_1. We use the subscripts dir, hr, and fr to distinguish between results computed via the direct, half-recursive, and fully recursive methods, respectively. The computed values of $A'_{1,\mathrm{dir}}$, $A'_{1,\mathrm{hr}}$, and $A'_{1,\mathrm{fr}}$ are *numerically identical* in that the corresponding entries are numerically equal:

$$\bar{A}'_{1,\mathrm{dir}} = \begin{pmatrix} 2.321253790030786e+00 & 2.775557561562891e-17 \\ 3.225930076892087e-07 & -5.198536633811768e-06 \end{pmatrix},$$
$$\bar{A}'_{1,\mathrm{hr}} = \begin{pmatrix} -5.198536633811768e-06 & -3.225930076892087e-07 \\ -2.775557561562891e-17 & 2.321253790030786e+00 \end{pmatrix},$$
$$\bar{A}'_{1,\mathrm{fr}} = \begin{pmatrix} -5.198536633811768e-06 & -3.225930076892087e-07 \\ -2.775557561562891e-17 & 2.321253790030786e+00 \end{pmatrix}.$$

The computed matrices $A'_{2,\mathrm{dir}}$, $A'_{2,\mathrm{hr}}$, and $A'_{2,\mathrm{fr}}$ are numerically triangular, but now the (1,2) element of $\bar{A}'_{2,\mathrm{fr}}$ is significantly different from the corresponding elements in $\bar{A}'_{1,\mathrm{dir}}$ and $\bar{A}'_{1,\mathrm{hr}}$:

$$\bar{A}'_{2,\mathrm{dir}} = \begin{pmatrix} 2.467752941777026e-05 & 5.551115123125783e-17 \\ 1.531353724707768e-06 & 4.237408446913959e+00 \end{pmatrix},$$
$$\bar{A}'_{2,\mathrm{hr}} = \begin{pmatrix} 4.237408446913959e+00 & -1.531353724707768e-06 \\ -5.551115123125783e-17 & 2.467752941777026e-05 \end{pmatrix},$$
$$\bar{A}'_{2,\mathrm{fr}} = \begin{pmatrix} 4.237408446913959e+00 & -1.531363626946 76e-06 \\ 0 & 2.467752941777026e-05 \end{pmatrix}.$$

To maintain triangularity, $\bar{A}'_1$ and $\bar{A}'_2$ are truncated by setting the appropriate elements to zero. Let A''_1 and A''_2 denote the truncated matrices. The product $\bar{A}'' = A''_1 \cdot A''_2$ should be diagonal:

$$\bar{A}''_{\mathrm{dir}} = \begin{pmatrix} 5.728280868959542e-05 & 0 \\ 1.615587133892632e-27 & -2.202832304370564e-05 \end{pmatrix},$$
$$\bar{A}''_{\mathrm{hr}} = \begin{pmatrix} -2.202832304370564e-05 & -1.615587133892632e-27 \\ 0 & 5.728280868959542e-05 \end{pmatrix},$$
$$\bar{A}''_{\mathrm{fr}} = \begin{pmatrix} -2.202832304370564e-05 & 5.010342801562901e-17 \\ 0 & 5.728280868959542e-05 \end{pmatrix}.$$

Clearly, $\bar{A}''_{\text{hr}}$ and $\bar{A}''_{\text{dir}}$ are numerically diagonal, but $\bar{A}''_{\text{fr}}$ fails the criterion of diagonality. Forcing $\bar{A}''_{\text{fr}}$ to be a diagonal matrix requires a truncation of $O(10^{-17})$, which is significant with respect to $\|\bar{A}''\|$. The matrices $\bar{A}''_{\text{dir}}$ and $\bar{A}''_{\text{hr}}$ require only insignificant truncations to obtain diagonality, but we have previously made $O(10^{-17})$ truncations during their computation to force $\bar{A}''_{2,\text{dir}}$ and $\bar{A}''_{2,\text{hr}}$ to triangular forms. Thus, equal amounts of absolute truncation errors have been committed by all three methods. The only difference is that the relative truncation error is largest for the fully recursive method.

It is interesting to note that if triangularity is not enforced and the factors $\bar{A}'_1$ and $\bar{A}'_2$ are multiplied, then none of the products can be considered diagonal. One may say that the numerical diagonality of $\bar{A}''_{\text{hr}}$ and $\bar{A}''_{\text{dir}}$ is a consequence of the truncation to triangular forms.

$$\bar{A}'_{1,\text{dir}} \cdot \bar{A}'_{2,\text{dir}} = \begin{pmatrix} 5.728280868959542e-05 & 2.464671807471544e-16 \\ 1.615587133892632e-27 & -2.202832304370564e-05 \end{pmatrix},$$

$$\bar{A}'_{1,\text{hr}} \cdot \bar{A}'_{2,\text{hr}} = \begin{pmatrix} -2.202832304370564e-05 & -1.615587133892632e-27 \\ -2.464671807471544e-16 & 5.728280868959542e-05 \end{pmatrix},$$

$$\bar{A}'_{1,\text{fr}} \cdot \bar{A}'_{2,\text{fr}} = \begin{pmatrix} -2.202832304370564e-05 & 5.010342801562901e-17 \\ -1.176117105626251e-16 & 5.728280868959542e-05 \end{pmatrix}.$$

In conclusion, our example shows that the half-recursive and direct methods produce numerically identical results, while the fully recursive method fails to meet the diagonality criterion.

REFERENCES

[1] Z. Bai and J.W. Demmel, *Computing the Generalized Singular Value Decomposition*, Report UCB/CSD 91/645, Computer Science Division, University of California, Berkeley, Aug. 1991.

[2] A.W. Bojanczyk, L.M. Ewerbring, F.T. Luk, and P. Van Dooren, *An accurate product SVD algorithm*, Signal Processing, 25 (1991), pp.189–201.

[3] J. P. Charlier, M. Vanbegin, and P. Van Dooren, *On efficient implementations of Kogbetliantz's algorithm for computing the singular value decomposition*, Numer. Math., 52 (1988), pp. 279–300.

[4] K. V. Fernando and S. J. Hammarling, *A product induced singular value decomposition for two matrices and balanced realisation*, in Linear Algebra in Signals, Systems and Control, B. N. Datta, C. R. Johnson, M. A. Kaashoek, R. J. Plemmons, and E. D. Sontag, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1988, pp. 128–140.

[5] M. T. Heath, A. J. Laub, C. C. Paige, and R. C. Ward, *Computing the SVD of a product of two matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1147–1159.

[6] C. C. Paige, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1126–1146.

[7] H. Zha, *A numerical algorithm for computing the restricted singular value decomposition of matrix triplets*, Linear Algebra Appl., 168 (1992), pp. 1–25.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 383–392, April 1994

003

NONLOCAL PERTURBATION ANALYSIS OF THE SCHUR SYSTEM OF A MATRIX*

M. M. KONSTANTINOV†, P. HR. PETKOV‡, AND N. D. CHRISTOV‡

Abstract. The sensitivity of the Schur system (the Schur basis and the Schur form) of a general matrix relative to perturbations, is studied. The estimates obtained are nonlocal and sharp. Asymptotic bounds (condition numbers particularly) in the form of power series are derived as a particular case.

Key words. Schur canonical form, perturbation analysis, conditioning of the Schur form

AMS subject classifications. Primary 15A21; Secondary 93B35, 65F15

1. Introduction. Basic notations. The Schur system of a matrix is a useful tool in many theoretical and computational problems (see, for example, [1]–[8]). Although much less sensitive to perturbations compared with Jordan and Frobenius systems, the Schur system nevertheless may be ill conditioned or even ill posed. Obtaining perturbation bounds for the Schur system is very important from both a theoretical and a practical point of view. In particular, the following problems, essential in numerical linear algebra and control theory, exploit the Schur form of a general matrix, and the solutions depend on the sensitivity of the corresponding Schur system.

(a) The computation of matrix exponential [6], [9], [10].

(b) The solution of Riccati equations by the Schur approach [2], [11].

(c) The pole assignment problem [12].

However, in contrast to other problems in linear algebra, the sensitivity of the Schur system of a matrix has not been studied to a sufficient extent. In particular, the sensitivity of invariant subspaces corresponding to clusters of equal or close eigenvalues have been examined [1]. At the same time, an invariant subspace is usually much less sensitive to perturbations compared with the corresponding basis. This phenomenon is especially important when studying the whole Schur system of an $n \times n$ matrix. Here the maximum invariant subspace is the basic n-dimensional space itself and is not sensitive at all, while the Schur basis may be infinitely sensitive.

In this paper we present a complete perturbation analysis of the whole Schur system without the assumption that the perturbations in the original matrix are asymptotically small. For small perturbations in the data, we give asymptotic bounds in the form of power series expansions.

We shall use the following notations:

$\mathbb{R}$ ($\mathbb{C}$), the set of real (complex) numbers;

$\mathbb{R}_+ = [0, \infty)$;

$\mathbb{C}^{m.n}$, the space of $m \times n$ complex matrices $\mathbf{A} = [a_{ij}]$ ($\mathbb{C}^{n.1} = \mathbb{C}^n$);

$\mathbf{A}^T$, the transpose of $\mathbf{A}$;

$\mathbf{A}^H = \bar{A}^T$, the complex conjugate transpose of $\mathbf{A}$;

$\mathbf{I}_n$, the unit $n \times n$ matrix;

$\mathbb{U}_n \subset \mathbb{C}^{n.n}$, the group of unitary matrices $\mathbf{U}$ ($\mathbf{U}^H\mathbf{U} = \mathbf{I}_n$);

$\mathbb{T}_n \subset \mathbb{C}^{n.n}$, the set of upper triangular matrices;

$\lambda_i(\mathbf{A})$, the eigenvalues of $\mathbf{A}$;

* Received by the editors July 22, 1991; accepted for publication (in revised form) August 6, 1992.

† University of Architecture, Civil Engineering and Geodesy, 1 Hr. Smirnenski Blv., BG-1421 Sofia, Bulgaria.

‡ Department of Automatics, Sofia Technical University, BG-1756 Sofia, Bulgaria.

$\sigma_1(\mathbf{A}) \geq \cdots \geq \sigma_r(\mathbf{A}) > 0$, the singular values of $\mathbf{A}$;
$\|\mathbf{x}\|$, the Euclidean norm of $\mathbf{x} \in \mathbb{C}^n$;
$\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$, the spectral norm of $\mathbf{A}$;
$\|\mathbf{A}\|_F = [\sigma_1^2(\mathbf{A}) + \cdots + \sigma_r^2(\mathbf{A})]^{1/2}$, the Frobenius norm of $\mathbf{A}$.
The set $\mathbb{C}^{m,n}$ is identified with $\mathbb{C}^{mn}$ as a linear space.
Each matrix $\mathbf{A} \in \mathbb{C}^{n,n}$ $(n \geq 2)$ is decomposed as the sum

$$\mathbf{A} = \text{low}\,(\mathbf{A}) + \text{diag}\,(\mathbf{A}) + \text{up}\,(\mathbf{A}) = \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3$$

of its strictly lower, diagonal, and strictly upper parts, say,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ c & 0 \end{bmatrix} + \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} + \begin{bmatrix} 0 & b \\ 0 & 0 \end{bmatrix}.$$

Low $(\cdot)$, diag $(\cdot)$, and up $(\cdot)$ are linear operators (projections) in $\mathbb{C}^{n,n}$ of ranks $n(n-1)/2$, n and $n(n-1)/2$, respectively.

2. Statement of the problem. Let $\mathbf{A} \in \mathbb{C}^{n,n}$. Then there exists $\mathbf{U} \in \mathbb{U}_n$ such that

$$\mathbf{U}^H\mathbf{A}\mathbf{U} = \mathbf{T}, \quad \text{or} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{T},\ \mathbf{T} \in \mathbb{T}_n,$$

where the matrix $\mathbf{T}$ is referred as the *Schur canonical form* of $\mathbf{A}$. The columns of $\mathbf{U}$ form the *Schur basis* of $\mathbf{A}$. If the matrix $\mathbf{A}$ is real and has real spectrum, then $\mathbf{T}$ is also real and $\mathbf{U}$ may be chosen real and orthogonal. The pair $(\mathbf{U}, \mathbf{T}) = (\mathbf{U}, \mathbf{U}^H\mathbf{A}\mathbf{U})$ is said to be the *Schur system* of $\mathbf{A}$. Note that the matrices $\mathbf{U}$ and (generally) $\mathbf{T}$ are not uniquely determined. Indeed, if $\mathbf{U}^H\mathbf{A}\mathbf{U} \in \mathbb{T}_n$, then $\mathbf{V}^H\mathbf{A}\mathbf{V} \in \mathbb{T}_n$, $\mathbf{V} = \mathbf{U}\mathbf{D}$, for each $\mathbf{D} = \text{diag}\,[d_1, d_2, \ldots, d_n]$, $|d_1| = |d_2| = \cdots = |d_n| = 1$. Moreover, we can achieve a Schur form $\mathbf{T}$ with any prescribed ordering of the eigenvalues of $\mathbf{A}$ on the diagonal of $\mathbf{T}$.

Denote by

$$\mathbb{B}(\mathbf{A}) = \{\mathbf{U} \in \mathbb{U}_n\colon \text{low}\,(\mathbf{U}^H\mathbf{A}\mathbf{U}) = \mathbf{0}\} \subset \mathbb{U}_n,$$

the set of all $\mathbf{U} \in \mathbb{U}_n$, whose columns form a Schur basis for $\mathbf{A}$, and let

$$\mathbb{S}(\mathbf{A}) = \{(\mathbf{U}, \mathbf{U}^H\mathbf{A}\mathbf{U})\colon \mathbf{U} \in \mathbb{B}(\mathbf{A})\} \subset \mathbb{U}_n \times \mathbb{T}_n$$

be the set of Schur systems of $\mathbf{A}$. It follows from the definitions that $\mathbb{B}(\mathbf{A})$ and $\mathbb{S}(\mathbf{A})$ are compact sets.

We assume for definiteness that the Schur system $(\mathbf{U}, \mathbf{T})$ of the unperturbed matrix $\mathbf{A}$ is chosen so that

$$\|\mathbf{I}_n - \mathbf{U}\|_F = \min\{\|\mathbf{I}_n - \mathbf{V}\|_F\colon \mathbf{V} \in \mathbb{B}(\mathbf{A})\}.$$

Suppose that $\mathbf{A}$ is subject to a perturbation $\Delta\mathbf{A} \in \mathbb{C}^{n,n}$. Then there exists $\Delta\mathbf{U} \in \mathbb{C}^{n,n}$ such that $\mathbf{U} + \Delta\mathbf{U} \in \mathbb{U}_n$ and

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{U} + \Delta\mathbf{U}) = (\mathbf{U} + \Delta\mathbf{U})(\mathbf{T} + \Delta\mathbf{T}), \qquad \mathbf{T} + \Delta\mathbf{T} \in \mathbb{T}_n,$$

where $\mathbf{T} + \Delta\mathbf{T}$ is the Schur form of $\mathbf{A} + \Delta\mathbf{A}$.

The perturbation analysis problem (PAP) consists in estimating the norms of the perturbations $\Delta\mathbf{U}$, $\Delta\mathbf{T}$ as functions of the norm of $\Delta\mathbf{A}$. A major difficulty here is the nonuniqueness of the Schur system.

Note that there exists an "exact" minimax bound for the perturbation in $\mathbf{U}$. Indeed, for each $\delta \geq 0$, the quantity

$$\varphi_U(\delta) := \max_{\|\mathbf{Y}\|_F = \delta} \min_{\mathbf{W} \in \mathbb{B}(\mathbf{A}+\mathbf{Y})} \|\mathbf{W} - \mathbf{U}\|_F$$

is well defined according to the Weierstrass theorem: $\varphi_U(\delta) = \|\mathbf{W}^* - \mathbf{U}\|_F$, where $\mathbf{W}^* = \mathbf{U} + \Delta\mathbf{U}^* \in \mathbb{B}(\mathbf{A} + \mathbf{Y})$ for some $\mathbf{Y}^*$ with $\|\mathbf{Y}^*\|_F = \delta$. The matrix $\Delta\mathbf{A}^* = \mathbf{Y}^*$ is the "worst" perturbation of $\mathbf{A}$, which causes a maximum norm perturbation in $\mathbf{U}$.

It follows from the definition of φ_U that:

(i) For each $\Delta\mathbf{A}$ there exists $\mathbf{U} + \Delta\mathbf{U} \in \mathbb{B}(\mathbf{A} + \Delta\mathbf{A})$ such that $\|\Delta\mathbf{U}\|_F \le \varphi_U(\|\Delta\mathbf{A}\|_F)$;

(ii) There exists $\Delta\mathbf{A}^*$ such that $\|\Delta\mathbf{U}\|_F \ge \varphi_U(\|\Delta\mathbf{A}^*\|_F)$ for each $\mathbf{U} + \Delta\mathbf{U} \in \mathbb{B}(\mathbf{A} + \Delta\mathbf{A}^*)$, but $\|\Delta\mathbf{U}^*\|_F = \varphi_U(\|\Delta\mathbf{A}^*\|_F)$ for some $\mathbf{U} + \Delta\mathbf{U}^* \in \mathbb{B}(\mathbf{A} + \Delta\mathbf{A}^*)$.

Consider briefly the properties of the function φ_U: $\mathbb{R}_+ \to \mathbb{R}_+$. Obviously, $\varphi_U(0) = 0$ and the "ideal" case would be if φ_U is continuous or even differentiable in certain interval $[0, \beta)$. Unfortunately, φ_U may be discontinuous at $\delta = 0$ if, for example, $\mathbf{A}$ has multiple eigenvalues. In this case the Schur basis is *infinitely sensitive* and the PAP is *ill posed.*

The Schur form $\mathbf{T} = \mathbf{U}^T\mathbf{A}\mathbf{U}$ is usually less sensitive compared with the Schur basis $\mathbf{U}$. In fact, consider the quantity

$$\varphi_T(\delta) := \max_{\|\mathbf{Y}\|_F = \delta} \min_{(\mathbf{W},\mathbf{S}) \in \mathbb{S}(\mathbf{A}+\mathbf{Y})} \|\mathbf{S} - \mathbf{T}\|_F.$$

It characterizes the sensitivity of the Schur form in the sense that for each $\Delta\mathbf{A}$, one has

$$\|\Delta\mathbf{T}\|_F \le \varphi_T(\|\Delta\mathbf{A}\|_F)$$

for some $(\mathbf{U} + \Delta\mathbf{U}, \mathbf{T} + \Delta\mathbf{T}) \in \mathbb{S}(\mathbf{A} + \Delta\mathbf{A})$.

It may be shown that for some matrices $\mathbf{A}$, the function φ_U is discontinuous at $\delta = 0$, while φ_T is continuous and $\varphi_T(\delta)$ is of order at most $O(\delta^{1/n})$, $\delta \to 0$.

The construction of the exact minimax bounds $\varphi_U(\delta)$, $\varphi_T(\delta)$ is practically impossible. That is the reason we look for upper bounds $e_U(\delta) \ge \varphi_U(\delta)$, $e_T(\delta) \ge \varphi_T(\delta)$.

We denote

$$\delta_A = \|\Delta\mathbf{A}\|_F, \quad \delta_U = \|\Delta\mathbf{U}\|_F, \qquad \delta_T = \|\Delta\mathbf{T}\|_F.$$

The practical PAP for the Schur system of $\mathbf{A}$ may be formulated in the following way: Determine an interval $J = [0, \alpha)$ and two continuous nondecreasing functions e_U: $J \to \mathbb{R}_+$, e_T: $J \to \mathbb{R}_+$ such that $e_U(0) = e_T(0) = 0$ and

$$\delta_U \le e_U(\delta_A), \quad \delta_T \le e_T(\delta_A); \quad \delta_A \in J \tag{1}$$

for at least one pair $(\mathbf{U} + \Delta\mathbf{U}, \mathbf{T} + \Delta\mathbf{T}) \in \mathbb{S}(\mathbf{A} + \Delta\mathbf{A})$ and all $\Delta\mathbf{A}$ with $\|\Delta\mathbf{A}\|_F = \delta_A$.

Note that (1) may be violated for some Schur systems of the perturbed matrix $\mathbf{A} + \Delta\mathbf{A}$. This property of the perturbation bounds is inevitable due to the nonuniqueness of the Schur system.

It must be pointed out that the estimates (1) are *nonlocal* since they are valid for a finite (although possibly small) interval J of perturbations δ_A.

If the functions e_U and e_T are twice differentiable, we get the asymptotic estimates

$$\delta_U \le k_U\delta_A + O(\delta_A^2), \quad \delta_T \le k_T\delta_A + O(\delta_A^2); \quad \delta_A \to 0, \tag{2}$$

where $k_U = e'_U(0)$ and $k_T = e'_T(0)$ are estimates of the *absolute condition numbers* (relative to $\mathbf{U}$ and $\mathbf{T}$) of the problem of computing the Schur system of $\mathbf{A}$. Estimates of the *relative condition numbers* are $c_U = k_U\|\mathbf{A}\|_F/\sqrt{n}$ and $c_T = k_T$, respectively.

When the bounds (2) exist, the PAP for the Schur system is *well posed*, otherwise it is *ill posed.* Note that the ill-posed problems include infinitely sensitive problems when a continuous bound for δ_U does not exist.

3. Main results. In this section we present a complete perturbation analysis of the Schur system of a general matrix $\mathbf{A}$ under certain natural assumptions. We first reformulate the problem of obtaining and estimating the Schur system $(\mathbf{U} + \Delta\mathbf{U}, \mathbf{T} + \Delta\mathbf{T})$ of $\mathbf{A} +$

$\Delta\mathbf{A}$ as an operator equation $\mathbf{X} = \mathcal{U}(\mathbf{X})$, $\mathbf{X} = \mathbf{U}^H\Delta\mathbf{U}$, in $\mathbb{C}^{n,n}$ with a continuous mapping $\mathcal{U}(\cdot)$. Then we show that $\mathcal{U}(\cdot)$ maps certain compact set $\mathbf{M} \subset \mathbb{C}^{n,n}$ into itself, where $\mathbf{M}$ is "small" of order $O(\delta_A)$. Hence according to the Schauder fixed point principle, there exists a solution $\mathbf{X} \in \mathbf{M}$ and its norm may be estimated in terms of the perturbation δ_A. Note that an attempt to apply the Banach fixed point principle seems to be inappropriate here since $\Delta\mathbf{U}$ (and hence $\mathbf{X}$) is always nonunique.

We assume that $\mathbf{A}$ has n pairwise distinct eigenvalues. This condition is not restrictive since its violation may cause not only illposedness of the Schur system, but even nonexistence of a continuous band for δ_U as shown by the following simple example.

Example 1. Let $n = 2$ and $\mathbf{A} = \mathbf{I}_2$. Here $\mathbb{S}(\mathbf{A}) = \mathbb{U}_2 \times \{\mathbf{A}\}$ and $\mathbf{U} = \mathbf{I}_2$, $\mathbf{T} = \mathbf{A}$. If $\Delta\mathbf{A}$ has a nonzero element only in position (2, 1), then $\mathbb{B}(\mathbf{A} + \Delta\mathbf{A})$ consists of the matrices $\pm\mathbf{U}_1, \pm\mathbf{U}_2$, where $u_{1,12} = -u_{1,21} = u_{2,12} = u_{2,21} = 1$. Hence the minimum norm perturbation satisfies $\|\Delta\mathbf{U}\|_F = \|\mathbf{I}_2 \pm \mathbf{U}_i\|_F = 2$ and, for this type of perturbations $\Delta\mathbf{A}$, the norm δ_U is not continuous as a function of δ_A: $\delta_U = 0$ for $\delta_A = 0$ and $\delta_U = 2$ for $\delta_A > 0$.

Setting $\mathbf{X} = \mathbf{U}^H\Delta\mathbf{U}$ and $\mathbf{E} = \mathbf{U}^H\Delta\mathbf{A}\mathbf{U}$, we get

$$(\mathbf{T} + \mathbf{E})(\mathbf{I}_n + \mathbf{X}) = (\mathbf{I}_n + \mathbf{X})(\mathbf{T} + \Delta\mathbf{T})$$

and

$$\mathbf{T}\mathbf{X} - \mathbf{X}\mathbf{T} = (\mathbf{I}_n + \mathbf{X})\Delta\mathbf{T} - \mathbf{E}(\mathbf{I}_n + \mathbf{X}). \tag{3}$$

Since $\mathbf{I}_n + \mathbf{X} \in \mathbb{U}_n$, we have

$$\mathbf{X}^H + \mathbf{X} + \mathbf{X}^H\mathbf{X} = \mathbf{0}. \tag{4}$$

Equation (3) yields

$$\Delta\mathbf{T} = \mathbf{G}(\mathbf{X}) = (\mathbf{I}_n + \mathbf{X}^H)[\mathbf{T}\mathbf{X} - \mathbf{X}\mathbf{T} + \mathbf{E}(\mathbf{I}_n + \mathbf{X})] \tag{5}$$

and

$$\begin{aligned} \delta_T &= \|\mathbf{G}(\mathbf{X})\|_F = \|\mathbf{T}\mathbf{X} - \mathbf{X}\mathbf{T} + \mathbf{E}(\mathbf{I} + \mathbf{X})\|_F \\ &\le \|\mathbf{T}\mathbf{X} - \mathbf{X}\mathbf{T}\|_F + \|\mathbf{E}\|_F \le \omega_A\|\mathbf{X}\|_F + \delta_A, \end{aligned} \tag{6}$$

where

$$\omega_A := \|\mathbf{I}_n \otimes \mathbf{T} - \mathbf{T}^T \otimes \mathbf{I}_n\|_2 = \|\mathbf{I}_n \otimes \mathbf{A} - \mathbf{A}^T \otimes \mathbf{I}_n\|_2.$$

Note that obviously $\omega_A \le 2\|\mathbf{A}\|_F$. However, this bound is probably pessimistic since, e.g., $\omega_A \le \sqrt{2}\|\mathbf{A}\|_F$ for $n = 2$.

Represent $\mathbf{X}$ as

$$\mathbf{X} = \text{low}\,(\mathbf{X}) + \text{diag}\,(\mathbf{X}) + \text{up}\,(\mathbf{X}) = \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3$$

and take operation low $(\cdot)$ from both sides of (3):

$$\text{low}\,(\mathbf{T}\mathbf{X} - \mathbf{X}\mathbf{T}) = \text{low}\,(\Delta\mathbf{T}) + \text{low}\,(\mathbf{X}\Delta\mathbf{T}) - \text{low}\,[\mathbf{E}(\mathbf{I}_n + \mathbf{X})].$$

Since low $(\Delta\mathbf{T}) = \mathbf{0}$, low $(\mathbf{T}\mathbf{X}) = \text{low}\,(\mathbf{T}\mathbf{X}_1)$, low $(\mathbf{X}\mathbf{T}) = \text{low}\,(\mathbf{X}_1\mathbf{T})$, and low $(\mathbf{X}\Delta\mathbf{T}) = \text{low}\,(\mathbf{X}_1\Delta\mathbf{T})$, then

$$\text{low}\,(\mathbf{T}\mathbf{X}_1 - \mathbf{X}_1\mathbf{T}) = \text{low}\,(\mathbf{X}_1\Delta\mathbf{T}) - \text{low}\,[\mathbf{E}(\mathbf{I}_n + \mathbf{X})]. \tag{7}$$

Consider the linear operator $\mathcal{L}(\cdot)$ in $\mathbb{C}^{n,n}$ defined by

$$\mathcal{L}(\mathbf{X}) = \text{low}\,(\mathbf{T}\mathbf{X} - \mathbf{X}\mathbf{T}).$$

Since $\mathfrak{L}(\mathbf{X}) = \mathfrak{L}(\mathbf{X}_1)$, then $\mathfrak{L}(\cdot)$ maps the subspace $\mathbf{M}_1 \subset \mathbb{C}^{n,n}$ of strictly lower triangular matrices into itself. Moreover, the eigenvalues of the restriction $\mathfrak{L}_1(\cdot)$ of $\mathfrak{L}(\cdot)$ on $\mathbb{C}^{n,n}/\mathbf{M}_1$ are $\lambda_i(\mathbf{A}) - \lambda_j(\mathbf{A})$, $i > j$, and hence $\mathfrak{L}_1(\cdot)$ is invertible. If we denote by vec $(\mathbf{X}_1) \in \mathbb{R}^{\mathfrak{v}}$, $\mathfrak{v} = n(n-1)/2$, the columnwise vector representation of $\mathbf{X}_1$, then vec $(\mathfrak{L}_1(\mathbf{X}_1)) =$ mat $(\mathfrak{L}_1)$ vec $(\mathbf{X}_1)$, where the matrix mat $(\mathfrak{L}_1) = \Omega = [\Omega_{ij}] \in \mathbb{R}^{\mathfrak{v},\mathfrak{v}}$ of the operator $\mathfrak{L}_1(\cdot)$ is block lower triangular ($\Omega_{ij} = \mathbf{0}$ for $i < j$). For $n = 5$, the matrix Ω is

$$\Omega = \begin{bmatrix} \tau_{21} & t_{23} & t_{24} & t_{25} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau_{31} & t_{34} & t_{35} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \tau_{41} & t_{45} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau_{51} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -t_{12} & 0 & 0 & \tau_{32} & t_{34} & t_{35} & 0 & 0 & 0 \\ 0 & 0 & -t_{12} & 0 & 0 & \tau_{42} & t_{45} & 0 & 0 & 0 \\ 0 & 0 & 0 & -t_{12} & 0 & 0 & \tau_{52} & 0 & 0 & 0 \\ 0 & 0 & -t_{13} & 0 & 0 & -t_{23} & 0 & \tau_{43} & t_{45} & 0 \\ 0 & 0 & 0 & -t_{13} & 0 & 0 & -t_{23} & 0 & \tau_{53} & 0 \\ 0 & 0 & 0 & -t_{14} & 0 & 0 & -t_{24} & 0 & -t_{34} & \tau_{54} \end{bmatrix},$$

where t_{ij} are the elements of $\mathbf{T}$ and $\tau_{ij} = \mathbf{t}_{ii} - \mathbf{t}_{jj} = \lambda_i(\mathbf{A}) - \lambda_j(\mathbf{A})$.

Denoting

$$\eta_A := \min \{\|\text{low } (\mathbf{TY} - \mathbf{YT})\|_F : \mathbf{Y} \in \mathbf{M}_1, \|\mathbf{Y}\|_F = 1\},$$

we have $\|\mathfrak{L}_1^{-1}\|_F = \|\Omega^{-1}\|_2 = \eta_A^{-1}$. Therefore, (7) may be rewritten as

$$\mathbf{X}_1 = \mathfrak{U}_1(\mathbf{X}) := \Lambda\{\mathbf{X}_1\Delta\mathbf{T} - \text{low } [\mathbf{E}(\mathbf{I}_n + \mathbf{X})]\}, \tag{8}$$

where $\Lambda(\cdot)$: $\mathbb{C}^{n,n} \to \mathbb{C}^{n,n}$ is a linear operator such that $\mathfrak{L}(\Lambda(\mathbf{X})) = \mathbf{X}_1$ and

$$\|\Lambda(\mathbf{X})\|_F \le \eta_A^{-1}\|\mathbf{X}\|_F. \tag{9}$$

Since we may choose $\mathbf{X}$ in (3), (4) with real diagonal elements x_{ii}, it follows from (4) that $x_{ii} = -0.5\|\mathbf{x}_i\|^2$; $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $\mathbf{x}_i \in \mathbb{C}^n$, and

$$\begin{aligned} \mathbf{X}_2 = \mathfrak{U}_2(\mathbf{X}) &:= -0.5 \text{ diag } [\|\mathbf{x}_1\|^2, \|\mathbf{x}_2\|^2, \ldots, \|\mathbf{x}_n\|^2] \\ &= -0.5 \text{ diag } (\mathbf{X}^H\mathbf{X}).. \end{aligned} \tag{10}$$

Relation (4) also gives

$$\mathbf{X}_3 = \mathfrak{U}_3(\mathbf{X}) := -\text{up } (\mathbf{X}^H) - \text{up } (\mathbf{X}^H\mathbf{X}). \tag{11}$$

Equations (8), (10), and (11) form an operator equation

$$\mathbf{X} = \mathfrak{U}(\mathbf{X}), \ \mathfrak{U} = \mathfrak{U}_1 + \mathfrak{U}_2 + \mathfrak{U}_3.$$

Consider now the set

$$\mathbf{M} = \{\mathbf{X}: \|\mathbf{X}_i\|_F \le c_i;\ i = 1, 2, 3\} \subset \mathbb{C}^{n,n}, \tag{12}$$

where c_i are positive constants that will be determined later.

For $\mathbf{X} \in \mathbf{M}$, we have

$$\|\mathbf{X}\|_F^2 = \|\mathbf{X}_1\|_F^2 + \|\mathbf{X}_2\|_F^2 + \|\mathbf{X}_3\|_F^2 \le c_1^2 + c_2^2 + c_3^2 = \|\mathbf{c}\|^2,$$

$$\mathbf{c} = [c_1, c_2, c_3]^T,$$

and in view of (8), (9), and (5), (6)

$$\begin{aligned} \|\mathfrak{U}_1(\mathbf{X})\|_F &\le \eta_A^{-1}\|\text{low}\,(\mathbf{X}_1\Delta\mathbf{T}) - \text{low}\,[\mathbf{E}(\mathbf{I}_n + \mathbf{X})]\|_F \\ &\le \eta_A^{-1}(\|\mathbf{X}_1\|_F\|\Delta\mathbf{T}\|_F + \|\mathbf{E}\|_F) \\ &\le \eta_A^{-1}[c_1(\omega_A\|\mathbf{c}\| + \delta_A) + \delta_A] = c_1(\gamma\|\mathbf{c}\| + \varepsilon) + \varepsilon, \end{aligned} \tag{13}$$

where

$$\gamma := \frac{\omega_A}{\eta_A}, \qquad \varepsilon := \frac{\delta_A}{\eta_A}.$$

Since

$$\sum_{i=1}^{n} \|\mathbf{x}_i\|^2 = \|\mathbf{X}\|_F^2,$$

we get

$$\frac{\|\mathbf{X}\|_F^4}{n} \le \sum_{i=1}^{n} \|\mathbf{x}_i\|^4 \le \|\mathbf{X}\|_F^4. \tag{14}$$

Hence (10) yields

$$\|\mathfrak{U}_2(\mathbf{X})\|_F = 0.5\left(\sum_{i=1}^{n} \|\mathbf{x}_i\|^4\right)^{1/2} \le 0.5\|\mathbf{X}\|_F^2 = 0.5\|\mathbf{c}\|^2. \tag{15}$$

On the other hand,

$$\|\mathbf{X}^H\mathbf{X}\|_F^2 = 2\|\text{up}\,(\mathbf{X}^H\mathbf{X})\|_F^2 + \sum_{i=1}^{n} \|\mathbf{x}_i\|^4$$

and (14) yields

$$\|\text{up}\,(\mathbf{X}^H\mathbf{X})\|_F^2 = 0.5\left(\|\mathbf{X}^H\mathbf{X}\|_F^2 - \sum_{i=1}^{n} \|\mathbf{x}_i\|^4\right) \le 0.5\left(1 - \frac{1}{n}\right)\|\mathbf{X}\|_F^4.$$

Thus

$$\|\text{up}\,(\mathbf{X}^H\mathbf{X})\|_F \le \mu_n\|\mathbf{X}\|_F^2, \qquad \mu_n := \left[0.5\left(1 - \frac{1}{n}\right)\right]^{1/2}. \tag{16}$$

Having in mind that $\|\text{up}\,(\mathbf{X}^H)\|_F = \|\mathbf{X}_1\|_F$, it follows from (11) and (16)

$$\begin{aligned} \|\mathfrak{U}_3(\mathbf{X})\|_F &\le \|\text{up}\,(\mathbf{X}^H)\|_F + \|\text{up}\,(\mathbf{X}^H\mathbf{X})\|_F \le \|\mathbf{X}_1\|_F + \mu_n\|\mathbf{X}\|_F^2 \\ &\le c_1 + \mu_n\|\mathbf{c}\|^2. \end{aligned} \tag{17}$$

Now we shall show that under certain conditions there exists a vector

$$\mathbf{c} = \mathbf{c}(\varepsilon), \quad \mathbf{c} = [c_1, c_2, c_3]^T; \quad c_i = c_i(\varepsilon), \tag{18}$$

whose components satisfy the nonlinear algebraic system

$$c_1 = f_1(\mathbf{c}, \varepsilon) := \gamma c_1(c_1^2 + c_2^2 + c_3^2)^{1/2} + c_1\varepsilon + \varepsilon, \tag{19}$$

$$c_2 = f_2(\mathbf{c}) := 0.5(c_1^2 + c_2^2 + c_3^2), \tag{20}$$

$$c_3 = f_3(\mathbf{c}) := c_1 + \mu_n(c_1^2 + c_2^2 + c_3^2), \tag{21}$$

(or $\mathbf{c} = \mathbf{f}(\mathbf{c}, \varepsilon) = [f_1(\mathbf{c}, \varepsilon), f_2(\mathbf{c}), f_3(\mathbf{c})]^T$) and the limits

$$\lim_{\varepsilon \to 0} c_i(\varepsilon) = 0; \qquad i = 1, 2, 3. \tag{22}$$

Denote by $\mathbf{M}_\varepsilon$ the set (12), (18). Then in view of (13), (15), (17), and (19)–(21), it follows that $\mathfrak{U}(\mathbf{M}_\varepsilon) \subset \mathbf{M}_\varepsilon$. Hence according to the Schauder fixed-point principle, there exists a solution $\mathbf{X} \in \mathbf{M}_\varepsilon$ of equation $\mathbf{X} = \mathfrak{U}(\mathbf{X})$ and $\|\mathbf{X}\|_F \le \|\mathbf{c}(\varepsilon)\| = \|\mathbf{c}(\delta_A/\eta_A)\|$.

Next we shall derive conditions for solvability of (19)–(21) and upper bounds for $c_i(\varepsilon)$ and $\|\mathbf{c}(\varepsilon)\|$.

There is a critical relation $\varepsilon = \varepsilon_n(\gamma)$, $\gamma \in \mathbb{R}_+$, such that:

(i) For $0 \le \varepsilon \le \varepsilon_n(\gamma)$ the solution (18), (22) of (19)–(21) exists. Moreover, the functions $c_i(\cdot)$ are analytic in $\varepsilon < \varepsilon_n(\gamma)$ and are continuous (but not differentiable) for $\varepsilon = \varepsilon_n(\gamma)$;

(ii) For $\varepsilon > \varepsilon_n(\gamma)$ the solution (18) does not exist.

In case (i), we can take

$$\delta_U \le e_U(\delta_A) := \|\mathbf{c}(\varepsilon)\| = [2c_2(\varepsilon)]^{1/2} = \left[2c_2\left(\frac{\delta_A}{\eta_A}\right)\right]^{1/2} \tag{23}$$

and, in view of (6),

$$\delta_T \le \omega_A \delta_U + \delta_A \le e_T(\delta_A) := \omega_A\left[2c_2\left(\frac{\delta_A}{\eta_A}\right)\right]^{1/2} + \delta_A. \tag{24}$$

The critical value $\varepsilon_n = \varepsilon_n(\gamma)$ is such that the Jacobi matrix $\mathbf{J}(\mathbf{c}, \varepsilon) := \partial\mathbf{f}(\mathbf{c}, \varepsilon)/\partial\mathbf{c}$ has an eigenvalue 1 for $\varepsilon = \varepsilon_n$, $\mathbf{f}(\mathbf{c}, \varepsilon_n) = \mathbf{c}$. This may be proved using the method of Lyapunov majorant functions [13]. However, as shown below, there is an easier way to determine the relation $\varepsilon = \varepsilon_n(\gamma)$.

Equations (20) and (21) give

$$c_2 = c_1^2 + 2\mu_n c_1 c_2 + (0.5 + 2\mu_n^2)c_2^2, \tag{25}$$

while (19) and (20) yield

$$c_1 = \varepsilon + \varepsilon c_1 + \gamma c_1 (2c_2)^{1/2}. \tag{26}$$

Rewrite (25) and (26) as

$$c_1 = g(t) := \varepsilon[1 - \varepsilon - \gamma(2t)^{1/2}]^{-1},$$

$$c_1 = h(t) := -\mu_n t + [t - (0.5 + \mu_n^2)t^2]^{1/2},$$

where $t := c_2 < \min\{t_g, t_h\}$ and

$$t_g := 0.5(1 - \varepsilon)^2\gamma^{-2}, \quad t_h := (0.5 + 2\mu_n^2)^{-1}, \quad h(t_h) = 0.$$

The relation $\varepsilon = \varepsilon_n(\gamma)$ may be determined in parametric form using the conditions $g(t) = h(t)$ and $g'(t) = h'(t)$:

$$\gamma = \gamma^0(t) := (2t)^{1/2}h'(t)[2th'(t) + h(t) + h^2(t)]^{-1},$$

$$\varepsilon = \varepsilon^0(t) := h^2(t)[2th'(t) + h(t) + h^2(t)]^{-1};$$

$$0 < t \le t_{\max} := [1 + 4\mu_n^2 + \mu_n(2 + 8\mu_n^2)^{1/2}]^{-1},$$

where the function h has maximum for $t = t_{\max}$. In addition, we must impose the restriction $\gamma \ge 1$ since $\omega_A \ge \eta_A$. Thus we obtain domains in the plane (γ, ε) for which alternative (i) holds true.

A disadvantage of this result is that the relation $c_2 = c_2(\varepsilon)$ in (23), (24) is not in explicit form. However, explicit (but slightly pessimistic) bounds for δ_U, δ_T may be found in the following way.

Since in view of (25) $c_1^2 \le c_2$, we get from (26) $c_1 \le \sqrt{2\gamma}c_2 + \varepsilon c_1 + \varepsilon$. Hence if we consider the system

$$x_1 = \sqrt{2\gamma}x_2 + \varepsilon x_1 + \varepsilon, \tag{27}$$

$$x_2 = x_1^2 + 2\mu_n x_1 x_2 + (0.5 + 2\mu_n^2)x_2^2, \tag{28}$$

instead of (25), (26), we shall have $c_1 \le x_1$, $c_2 \le x_2$. Setting the expression for x_1 obtained from (27) in (28), we get

$$x_2 = p(\varepsilon) + q(\varepsilon)x_2 + r(\varepsilon)x_2^2, \tag{29}$$

where

$$\begin{aligned} p(\varepsilon) &:= \varepsilon^2(1-\varepsilon)^{-2}, \\ q(\varepsilon) &:= 2\varepsilon(1-\varepsilon)^{-2}[\sqrt{2\gamma} + \mu_n(1-\varepsilon)], \\ r(\varepsilon) &:= (1-\varepsilon)^{-2}[\sqrt{2\gamma} + \mu_n(1-\varepsilon)]^2 + 0.5 + \mu_n^2. \end{aligned} \tag{30}$$

The condition for existence of a solution of (29) is $[1 - q(\varepsilon)]^2 \ge 4p(\varepsilon)r(\varepsilon)$ or $q(\varepsilon) + 2[p(\varepsilon)r(\varepsilon)]^{1/2} \le 1$, which is equivalent to

$$2\varepsilon\{[(\sqrt{2\gamma} + \mu_n(1-\varepsilon)]^2 + (0.5 + \mu_n^2)(1-\varepsilon)^2\}^{1/2} + 2\varepsilon[\sqrt{2\gamma} + \mu_n(1-\varepsilon)] \le (1-\varepsilon)^2. \tag{31}$$

Inequality (31) may be strengthened to

$$2\varepsilon[\sqrt{2\gamma} + \mu_n(1-\varepsilon)] + 2\varepsilon\alpha_n(\gamma) \le (1-\varepsilon)^2, \tag{32}$$

where

$$\alpha_n(\gamma) := [(\sqrt{2\gamma} + \mu_n)^2 + \mu_n^2 + 0.5]^{1/2}. \tag{33}$$

Using (32), (33), it is easy to show that

$$\begin{aligned} \varepsilon \le \varepsilon_n^*(\gamma) &= [\beta_n(\gamma) - [\beta_n^2(\gamma) - 1 - 2\mu_n]^{1/2}]/(1 + 2\mu_n), \\ \beta_n(\gamma) &:= 1 + \sqrt{2\gamma} + \mu_n + \alpha_n(\gamma). \end{aligned} \tag{34}$$

If ε satisfies (31) or (34), then

$$\begin{aligned} x_2 = x_2(\varepsilon) &= [2r(\varepsilon)]^{-1}[1 - q(\varepsilon) - d^{1/2}(\varepsilon)], \\ d(\varepsilon) &:= [1 - q(\varepsilon)]^2 - 4p(\varepsilon)r(\varepsilon), \end{aligned} \tag{35}$$

and

$$\delta_U \le e_U^*(\delta_A) := [2x_2(\varepsilon)]^{1/2} = \left[2x_2\left(\frac{\delta_A}{\eta_A}\right)\right]^{1/2}, \tag{36}$$

$$\delta_T \le e_T^*(\delta_A) := \omega_A\left[2x_2\left(\frac{\delta_A}{\eta_A}\right)\right]^{1/2} + \delta_A. \tag{37}$$

Consider finally the case when δ_A and hence ε is asymptotically small. It may be shown that the solution of (25), (26) admits the power series expansion

$$\begin{aligned} c_1(\varepsilon) &= \varepsilon + (1 + \sqrt{2\gamma})\varepsilon^2 + [1 + \sqrt{2}(3 + \mu_n)\gamma + \gamma^2]\varepsilon^3 + \cdots, \\ c_2(\varepsilon) &= \varepsilon^2 + 2(1 + \sqrt{2\gamma} + \mu_n)\varepsilon^3 \\ &\quad + [3.5 + 6\mu_n + 6\mu_n^2 + 8\sqrt{2}(1 + \mu_n)\gamma + 10\gamma^2]\varepsilon^4 + \cdots, \end{aligned}$$

and

$$\delta_U \le e_U(\delta_A) = [2c_2(\varepsilon)]^{1/2} = \sqrt{2}\{\varepsilon + (1 + \sqrt{2}\gamma + \mu_n)\varepsilon^2 + [1.25 + 2\mu_n + 2.5\mu_n^2 + 3\sqrt{2}(1 + \mu_n)\gamma + 4\gamma^2]\varepsilon^3 + \cdots\}. \tag{38}$$

In particular, the condition numbers are

$$k_U = \frac{\sqrt{2}}{\eta_A}, \quad c_U = \left(\frac{2}{n}\right)^{1/2} \frac{\|\mathbf{A}\|_F}{\eta_A}; \quad k_T = c_T = 1 + \frac{\sqrt{2}\omega_A}{\eta_A}.$$

The solution of (27), (28) has the expansion

$$x_1(\varepsilon) = c_1(\varepsilon) + \sqrt{2}\gamma\mu_n\varepsilon^3 + O(\varepsilon^4),$$
$$x_2(\varepsilon) = c_2(\varepsilon) + 2\sqrt{2}\gamma\mu_n\varepsilon^4 + O(\varepsilon^5),$$

and

$$\delta_U \le e_U^*(\delta_A) = [2x_2(\varepsilon)]^{1/2} = [2c_2(\varepsilon)]^{1/2} + 2\gamma\mu_n\varepsilon^3 + O(\varepsilon^4), \tag{39}$$

i.e., the estimates e_U, e_U^*, defined by (38), (39) coincide within terms of second order with respect to ε.

The estimates derived above are relatively sharp as shown by the following examples.

Example 2. Let $\mathbf{A} \in \mathbb{R}^{5,5}$ and $\Delta\mathbf{A} = 10^{-k}\mathbf{B}$ be determined from

$$\mathbf{A} = \begin{bmatrix} 4.5 & 0.9 & -1.9 & 4.4 & -9.7 \\ 4.3 & 1.9 & -1.9 & 5.2 & -11.3 \\ -3.1 & 0.0 & 2.0 & -3.6 & 6.7 \\ 1.7 & 0.9 & -1.9 & 4.6 & -7.1 \\ 3.0 & 0.9 & -1.9 & 4.4 & -8.2 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} -3 & 1 & 7 & -4 & 1 \\ 6 & 0 & 4 & 2 & 9 \\ -3 & -2 & 7 & 1 & -5 \\ 8 & 6 & -9 & -3 & 4 \\ 7 & 4 & -3 & 2 & 6 \end{bmatrix}$$

for k being an integer. The set of eigenvalues of $\mathbf{A}$ is $\{0.1, 0.2, 1.0, 1.5, 2.0\}$. The results obtained from (36), (37) are shown in Table 1.

Our last example shows that the estimates obtained are asymptotically exact.

Example 3. Let $\mathbf{A} = \operatorname{diag}[\lambda_1, \lambda_2] \in \mathbb{R}^{2,2}$, where $\lambda_1 > \lambda_2$. We have $\omega_A = \eta_A = \lambda_1 - \lambda_2 > 0$ and hence $\gamma = 1$. If we choose $\Delta\mathbf{A}$ in the form

$$\Delta\mathbf{A} = \begin{bmatrix} -a_1 & 0 \\ -a & a_2 \end{bmatrix}; \qquad a_1, a_2 \ge 0,$$

where

$$\delta_A = (a_1^2 + a_2^2 + a^2)^{1/2} < \frac{\eta_A}{\sqrt{2}}, \tag{40}$$

TABLE 1

k	δ_A	δ_U	e_U^*	δ_T	e_T^*
11	2.492×10^{-10}	3.413×10^{-9}	1.408×10^{-8}	2.843×10^{-8}	4.721×10^{-7}
12	2.492×10^{-11}	3.412×10^{-10}	1.408×10^{-9}	2.843×10^{-9}	4.721×10^{-8}
13	2.492×10^{-12}	3.411×10^{-11}	1.407×10^{-10}	2.842×10^{-10}	4.720×10^{-9}

then

$$\mathbf{X} = \begin{bmatrix} c-1 & s \\ -s & c-1 \end{bmatrix}; \quad s^2 + c^2 = 1, \quad \frac{s}{c} = \frac{a}{\eta_A - a_1 - a_2}.$$

Hence

$$\delta_U = \|\mathbf{X}\|_F = 2(1-c)^{1/2}, \qquad c = [1 + a_1^2(\eta_A - a_1 - a_2)^{-2}]^{-1/2}. \tag{41}$$

According to (41), the maximum of δ_A in (a_2, a_2, a) subject to the constraint (40) is achieved for

$$a_1 = a_2 = \frac{\delta_A^2}{\eta_A} = \delta_A \varepsilon, \quad a = \delta_A(1 - 2\varepsilon^2)^{1/2}; \quad \varepsilon = \frac{\delta_A}{\eta_A},$$

and is equal to

$$\begin{aligned} \delta_{U,\max}(\varepsilon) &= 2[1 - c_{\min}(\varepsilon)]^{1/2}, \\ c_{\min}(\varepsilon) &= [1 - \varepsilon^2(1-\varepsilon^2)^{-1}]^{1/2}, \\ \varepsilon &< \frac{1}{\sqrt{2}} \approx 0.7071. \end{aligned} \tag{42}$$

Thus the exact bound (42) has the expansion

$$\begin{aligned} \delta_{U,\max}(\varepsilon) &= \sqrt{2}\varepsilon\left[1 + \left(\frac{5}{8}\right)\varepsilon^2 + \left(\frac{79}{128}\right)\varepsilon^4 + \cdots\right] \\ &\approx \sqrt{2}\varepsilon(1 + 0.625\varepsilon^2 + 0.617\varepsilon^4 + \cdots) \end{aligned}$$

while the bound (36) and the inequality (34) give

$$\delta_U \le e_U(\varepsilon\eta_A) \approx \sqrt{2}\varepsilon(1 + 2.914\varepsilon + 13.239\varepsilon^2 + \cdots),$$

$$\varepsilon < \varepsilon_2^*(1) \approx 0.2035,$$

and the difference between the exact (42) and approximate (39) bound is of asymptotic order $4.121\varepsilon^2$.

REFERENCES

[1] G. W. STEWART, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.

[2] A. J. LAUB, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Autom. Control, AC-24 (1979), pp. 913–921.

[3] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[4] P. VAN DOOREN, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Autom. Control, AC-26 (1981), pp. 111–129.

[5] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, 1986.

[6] M. M. KONSTANTINOV, N. D. CHRISTOV, AND P. HR. PETKOV, *Perturbation analysis of linear control problems*, 10th IFAC Congress, Munich, 1987, Vol. 9, pp. 16–21.

[7] P. HR. PETKOV, N. D. CHRISTOV, AND M. M. KONSTANTINOV, *A computational algorithm for pole assignment of linear multiinput systems*, IEEE Trans. Autom. Control, AC-31 (1986), pp. 1044–1047.

[8] G. W. STEWART AND J.-G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[9] B. KÅGSTRÖM, *Bounds and perturbation bounds for the matrix exponential*, BIT, 17 (1977), pp. 39–57.

[10] C. F. VAN LOAN, *The sensitivity of the matrix exponential*, SIAM J. Numer. Anal., 14 (1977), pp. 971–981.

[11] V. L. MEHRMANN, *The Autonomous Linear Quadratic Control Problem. Theory and Numerical Solution*, Springer-Verlag, Berlin, 1991.

[12] P. HR. PETKOV, N. D. CHRISTOV, AND M. M. KONSTANTINOV, *A new approach to the perturbation analysis of linear control problems*, 11th IFAC World Congress, Tallinn, 1990, Vol. 2, pp. 311–316.

[13] E. A. GREBENIKOV AND YU. A. RYABOV, *Constructive Methods for Analysis of Nonlinear Systems*, Nauka, Moscow, 1979. (In Russian.)

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 393–407, April 1994

004

AN ALGORITHM FOR THE SINGLE-INPUT POLE ASSIGNMENT PROBLEM*

RAFAEL BRU†, JOSÉ MAS†, AND ANA M. URBANO†

Abstract. B. N. Datta and K. Datta have proposed an efficient parallel algorithm for the single-input pole assignment problem when the spectrum set to be assigned is pairwise distinct and disjoint from the spectrum of the dynamical matrix. This paper first presents a theoretical analysis of that algorithm and then presents the necessary modifications on it for an arbitrary spectrum. Based on the modified algorithm, the authors present two new algorithms for the problem and give some results on controllability that are of independent interests.

Key words. eigenvalue assignment, single-input, controllability, Jordan form

AMS subject classifications. 93B55, 93B60, 15A18

1. Introduction.

Given the time-invariant linear dynamical system

$$\dot{x}(t) = Ax(t) + bu(t),$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^{n\times 1}$, an important problem in control theory is the eigenvalue assignment problem commonly known as the pole-assignment problem; that is, to find a vector f such that the spectrum of the matrix, $A - bf^T$ is equal to the prespecified and conjugated complex number set $\Omega = \{\mu_1, \mu_2, \ldots, \mu_n\}$. One application of this problem is the stabilization of a controllable system; that is, if the dynamic system has some disturbance, how to choose a vector f such that the closed-loop system comes back to the stable position. It is well known that this problem has a unique solution if and only if the pair (A, b) is controllable [10].

There exist several sequential methods for computing the vector f. Some well-known and important methods are found in [1], [2], [5]–[7], and [9].

In [4], B. N. Datta and K. Datta proposed a *parallel* algorithm for solving the above problem. This algorithm implicitly assumes that the eigenvalues to be assigned are pairwise distinct and different from those of A. The main aim of this paper is to modify that method for any self-conjugated set Ω, proposing some modifications of that parallel algorithm to obtain the vector f such that $A - bf^T$ has the desired spectrum for any set Ω. We point out here that a multi-input version of the Datta–Datta single-input parallel algorithm was also proposed in [3]. This multi-input version also implicitly assumes that the set of eigenvalues to be assigned is disjoint from the spectrum of A and pairwise distinct.

The paper is organized as follows. First, in §2 we present a short description of the algorithm given in [4] that is the starting point of our analysis. Then in §3 we give some controllability results and theoretical analysis of that algorithm proving necessary and sufficient conditions on the validity of that algorithm. In §4 we study the modification when $\Omega \cap \sigma(A) \neq \emptyset$ (where $\sigma(A)$ denotes the spectrum of A), and in §5 we give a general algorithm when $\Omega \cap \sigma(A) \neq \emptyset$ and some eigenvalue has multiplicity greater than one. Finally, in §6 we illustrate our algorithms with some numerical experiments with the help of MATLAB.

* Received by the editors January 17, 1989; accepted for publication (in revised form) September 21, 1992. This work was supported by Spanish CICYT grant TIC:87-0655.

† Departament de Matemàtica Aplicada, Universitat Politècnica de València, Apt. de Correus 22012, 46071, València, Spain (rbru@mat.upv.es).

2. A single-input parallel algorithm for eigenvalue assignment by Datta and Datta. Given the controllable pair (A, b) and the set $\Omega = \{\mu_1, \mu_2, \ldots, \mu_n\}$, the algorithm proposed in [4] computes a vector f such that $\sigma(A - bf^T) = \Omega$. The algorithm is suitable for parallel implementation and is described as follows.

Stage I. Transform the pair (A, b) by orthogonal similarity to the pair (H, c), where $QAQ^T = H$ is an unreduced upper Hessenberg matrix and $Qb = c = [\alpha, 0, \ldots, 0]^T$, $\alpha \neq 0$.

Stage II. Solve the n Hessenberg systems in PARALLEL:

$$(H - \mu_i I)t_i = c, \quad i = 1, \quad 2, \ldots, n.$$

Stage III. Solve for d:

$$T^T d = r,$$

where $T = [t_1, t_2, \ldots, t_n]$ and $r = [\alpha, \alpha, \ldots, \alpha]^T$.

Stage IV. Compute $f^T = \alpha^{-1} d^T Q$.

In Stage I of the algorithm, the orthogonal similarity Q transforms the pair (A, b) to the equivalent pair (H, c), where H is an upper Hessenberg matrix and $c = [\alpha, 0, \ldots, 0]^T$. It is well known that (A, b) is controllable if and only if the matrix H is unreduced (that is, $h_{i,i-1} \neq 0$, $i = 2, \ldots, n$) and $\alpha \neq 0$ (see [6] and [7]).

Since

$$Q(A - bf^T)Q^T = H - cf^T Q^T,$$

the problem of finding the vector f such that $\sigma(A - bf^T) = \Omega$ is equivalent to finding the vector g, defined by $g^T = f^T Q^T$, such that

$$\sigma(H - cg^T) = \Omega.$$

Stage II of the algorithm solves the systems

$$(H - \mu_i I)t_i = c, \quad i = 1, \quad 2, \ldots, n. \tag{1}$$

Assuming that T is nonsingular, it computes the vector $d^T = r^T T^{-1}$ in Stage III. In Stage IV the vector f is computed as

$$f^T = \alpha^{-1} d^T Q = [1, 1, \ldots, 1]T^{-1}Q = g^T Q.$$

Theorem 2.2 of [4] proves that the above algorithm provides the vector f such that $\sigma(A - bf^T) = \Omega$, assuming that the matrix T obtained in Stage II is invertible. We prove that the nonsingularity of T is guaranteed when the elements of the set Ω are pairwise distinct and when no element of Ω is in $\sigma(A)$ (see Proposition 3). Thus the above assumptions in the algorithm given in [4] are not a restriction because it works implicitly with these conditions. As we shall see in the next section, the nonsingularity of T is related with the consistency of the systems $(H - \mu I)t = c$ of Stage II.

3. Consistency and nonsingularity. In general, the matrix T obtained by solving (1) is not invertible if some eigenvalue has multiplicity greater than one as Proposition 1 shows. Furthermore, in the case that $\Omega \cap \sigma(A) \neq \varnothing$, the matrix T cannot be constructed from the systems of Stage II, as can be seen in Proposition 2. From the above remarks, we observe that the algorithm given in [4] is valid if and only if $\Omega \cap \sigma(A) = \varnothing$ and Ω is a pairwise disjoint set, as the following theorem proves. Since $\sigma(A) = \sigma(H)$, we state our results for an unreduced upper Hessenberg matrix H.

THEOREM 1. *Let (H, c) be a controllable pair. Let Ω be a conjugated complex number set. The algorithm given in* [4] *is valid if and only if the following conditions are satisfied*: (i) $\Omega \cap \sigma(H) = \varnothing$, and (ii) *the elements of Ω are pairwise distinct.*

The proof of Theorem 1 is based on the following results.

PROPOSITION 1. *Let (H, c) be a controllable pair. Let $\Omega = \{\mu_1, \mu_2, \ldots, \mu_n\}$ be a conjugated complex number set. If $\Omega \cap \sigma(H) = \varnothing$, but $\mu_i = \mu_j$ for some $i \neq j$, then the matrix T constructed in Stage* II *is singular.*

Proof. Since $\Omega \cap \sigma(H) = \varnothing$, then rank $(H - \mu_i I) = n$. Therefore, the system $(H - \mu_i I)x = c$ has a unique solution and hence the matrix T has the ith and jth columns equal. Hence T is singular. □

The following lemma gives a basic result on the controllable systems with single input, and we will use it in the next proposition. Moreover, the remark below is the main point for constructing the general algorithm given in §5.

LEMMA 1. *Let (H, c) be a controllable pair. Then*

(i) dim Ker $[H - \mu I, c] = 1$, *for all μ.*

(ii) If $\mu \in \sigma(H)$, *then the geometric multiplicity of μ is* 1.

Proof. (i) If $\mu \notin \sigma(H)$, it is obvious. Otherwise, since (H, c) is controllable, rank $[H - \mu I, c] = n$ and hence dim Ker $[H - \mu I, c] = 1$.

(ii) Since rank $[H - \mu I, c] = n$ and considering that $\mu \in \sigma(H)$, we deduce that dim Ker $(H - \mu I) = 1$. □

Remark. Let γ_j be the jth column of the matrix $H - \mu I$, $j = 1, 2, \ldots, n$.

(a) Then by Lemma 1(ii), and since $H - \mu I$ is an unreduced upper Hessenberg matrix, we have

$$\text{span } \{\gamma_1, \gamma_2, \ldots, \gamma_{n-1}, \gamma_n\} = \text{span } \{\gamma_1, \gamma_2, \ldots, \gamma_{n-1}\}.$$

(b) By Lemma 1(i) and the last remark, we have the following direct sums:

$$\mathbb{C}^n = \text{span } \{\gamma_1, \gamma_2, \ldots, \gamma_{n-1}, \gamma_n\} \oplus \text{span } \{c\}$$

$$= \text{span } \{\gamma_1, \gamma_2, \ldots, \gamma_{n-1}\} \oplus \text{span } \{c\}.$$

PROPOSITION 2. *Consider the controllable pair (H, c). Let Ω be a conjugated complex number set. If $\mu \in \Omega \cap \sigma(H)$, then the system $(H - \mu I)x = c$ is inconsistent.*

Proof. If $\mu \in \sigma(H)$, then by Lemma 1(ii), we have that rank $(H - \mu I) = n - 1$, and since (H, c) is a controllable pair, rank $[H - \mu I, c] = n$. Thus, the system $(H - \mu I)x = c$ is inconsistent. □

For the next proposition we consider, for each $p = 1, 2, \ldots, n$, the following partition of the matrix $H - \mu_i I$ and the vectors t_i and c.

$$H - \mu_i I = \left[\begin{array}{c|c} \underset{n-p}{H_1 - \mu_i I} & \underset{p}{S} \\ \hline \underset{n-p}{R} & \underset{p}{H_2 - \mu_i I} \end{array}\right]^{n-p}_{p}$$

$$t_i = \begin{bmatrix} t_{1i} \\ \cdot \\ \cdot \\ t_{n-p,i} \\ \hline t_{n-p+1,i} \\ \cdot \\ \cdot \\ t_{n,i} \end{bmatrix} = \begin{bmatrix} t_i^{n-p} \\ \hline t_i^{p} \end{bmatrix}, \qquad c = \begin{bmatrix} \alpha \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \hline 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} = \left[\begin{array}{c} c^{n-p} \\ \hline 0 \end{array}\right]^{n-p}_{p}$$

for each $i = 1, 2, \ldots, p$.

PROPOSITION 3. *Let (H, c) be a controllable pair. Let $\Omega = \{\mu_1, \mu_2, \ldots, \mu_n\}$ be a conjugated complex number set, pairwise distinct. Consider, in addition, that $\Omega \cap \sigma(H) = \varnothing$. Let t_i be the solution of the system*

$$(H - \mu_i I)t_i = c \quad \text{for each } i = 1, 2, \ldots, n. \tag{2}$$

Then the rank of the matrix $[t_1^p, t_2^p, \ldots, t_p^p]$ is equal to p for each $p = 1, 2, \ldots, n$.

Proof. We proceed by induction over p. For $p = 1$, it is evident that $t_{1n} \neq 0$. Otherwise, since H is unreduced Hessenberg matrix and using back substitution we deduce that the vector t_1 is the zero vector, which is not a solution of the nonhomogeneous system $(H - \mu_1 I)x = c$.

Suppose by induction that the rank of the matrix $[t_1^p, t_2^p, \ldots, t_p^p]$ is p. Obviously, the rank of the matrix

$$[t_1^{p+1}, t_2^{p+1}, \ldots, t_p^{p+1}, t_{p+1}^{p+1}]$$

is p or $p + 1$. Suppose the rank of this matrix is p instead of $p + 1$. Then by the induction hypothesis we have the following linear combination:

$$t_{p+1}^{p+1} = \sum_{j=1}^{p} \beta_j t_j^{p+1}. \tag{3}$$

For each $i = 1, 2, \ldots, p + 1$, by the above partitions, we can write system (2) as

$$\left[\begin{array}{c|c} H_1 - \mu_i I & S \\ \hline R & H_2 - \mu_i I \end{array}\right] \left[\begin{array}{c} t_i^{n-p-1} \\ \hline t_i^{p+1} \end{array}\right] = \left[\begin{array}{c} c^{n-p-1} \\ \hline 0 \end{array}\right]. \tag{4}$$

Now restricting ourselves to the second subsystem of (4), for $i = 1, 2, \ldots, p$, we have

$$Rt_i^{n-p-1} + (H_2 - \mu_i I)t_i^{p+1} = 0, \tag{5}$$

and for $i = p + 1$, taking into account (3), we get

$$Rt_{p+1}^{n-p-1} + (H_2 - \mu_{p+1} I) \sum_{j=1}^{p} \beta_j t_j^{p+1} = 0. \tag{6}$$

Subtracting expression (6) from the sum of all p subsystems (5) each one multiplied by β_i, $i = 1, 2, \ldots, p$, we obtain

$$R\left[\sum_{i=1}^{p} \beta_i t_i^{n-p-1} - t_{p+1}^{n-p-1}\right] + \sum_{i=1}^{p} \beta_i(\mu_{p+1} - \mu_i)t_i^{p+1} = 0.$$

Since the matrix R has only the $(1, n - p)$th entry different from zero, we have that the last p rows satisfy

$$\sum_{i=1}^{p} \beta_i(\mu_{p+1} - \mu_i)t_i^p = 0.$$

Hence the rank of matrix $[t_1^p, t_2^p, \ldots, t_p^p]$ is less than p, which contradicts the induction hypothesis. □

We are now ready to prove Theorem 1.

Proof of Theorem 1. Sufficiency: Follows from Proposition 3. Necessity: If $\Omega \cap \sigma(H) \neq \varnothing$, by Proposition 2, the matrix T cannot exist. On the other hand if multiplicity of $\mu > 1$, for some $\mu \in \Omega$, by Proposition 1, T is singular. □

4. Proposed pole placement algorithm. If $\mu \in \Omega \cap \sigma(H)$, we then have seen in Proposition 2 that the Datta–Datta algorithm is not valid. However, it is always possible to find an eigenvector of H associated with μ, solving the system $(H - \mu I)x = 0$. In fact, we shall see in Theorem 2 that in this case (that is, when $\Omega \cap \sigma(H) \neq \varnothing$) the matrix T constructed in (7) is nonsingular. We will thus have the following modified algorithm.

ALGORITHM I

Stage I. Transform the pair (A, b) by orthogonal similarity to the pair (H, c), where H is an upper unreduced Hessenberg matrix and $c = [\alpha, 0, \ldots, 0]^T$, with $\alpha \neq 0$.

Stage II. For $i = 1, 2, \ldots, n$.

If $\mu_i \notin \sigma(H)$, then

Solve the Hessenberg systems in PARALLEL

$$(H - \mu_i I)t_i = c.$$

Else

Solve the Hessenberg systems in PARALLEL for the nonzero solution

$$(H - \mu_i I)t_i = 0.$$

Stage III. Form the vector u as follows: assign 1 to the ith entry if $\mu_i \in \sigma(H)$, otherwise assign 0 to this entry.

Stage IV. Solve for g:

$$g^T T = u^T,$$

where

$$T = [t_1, t_2, \ldots, t_n].$$

Stage V. Compute $f^T = g^T Q$.

The following result proves the validity of the above algorithm.

THEOREM 2. *Let (H, c) be a controllable pair and let $\Omega = \{\mu_1, \mu_2, \ldots, \mu_n\}$ be a conjugated complex number set pairwise distinct. If t_i, $i = 1, 2, \ldots, n$, is a solution of the systems*

$$(H - \mu_i I)t_i = \begin{cases} 0 & \text{if } \mu_i \in \sigma(H), \\ c & \text{if } \mu_i \notin \sigma(H), \end{cases} \tag{7}$$

where $c = [\alpha, 0, \ldots, 0]^T$, $\alpha \neq 0$, then the matrix

$$[t_1^p, t_2^p, \ldots, t_p^p]$$

has rank p, *for each* $p = 1, 2, \ldots, n$.

Proof. The proof is analogous to that of Proposition 3. □

The assumption that the complex numbers of Ω are pairwise distinct in Theorem 2 is necessary as the following example shows.

Example 1. Let

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

and suppose that we want to assign the spectrum $\Omega = \{1, 1, 2\}$. The solution of the systems (7) yields the singular matrix

$$T = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix}.$$

5. The general algorithm. With the hypothesis of Theorem 2, solving the systems (7) is equivalent to solving the matrix system

$$HT - cu^T = T \operatorname{diag}(\mu_i), \tag{8}$$

where u^T is a row vector constructed in the proposed algorithm: its ith entry is 0 if $\mu_i \in \sigma(H)$ and 1 otherwise. From (8) note that the matrix $H - cu^T T^{-1}$ is diagonalizable as in the case of [4]. However, in the general case where the spectrum to be assigned has some eigenvalue with multiplicity greater than one, the matrix $H - cu^T T^{-1}$ is not similar to a diagonal matrix as the following proposition shows.

PROPOSITION 4. *Let (H, c) be a controllable pair, and let $\Omega = \{\mu_1, \mu_2, \ldots, \mu_p\}$ be the spectrum to be assigned with multiplicities $m(1), m(2), \ldots, m(p)$, respectively. If g is the vector such that $\sigma(H - cg^T) = \Omega$, then the Jordan form of $H - cg^T$ is*

$$J = \begin{bmatrix} J_1 & 0 & \cdots 0 \\ 0 & J_2 & \cdots 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & \cdots J_p \end{bmatrix}, \tag{9}$$

where J_k is the $m(k) \times m(k)$ Jordan matrix

$$J_k = \begin{bmatrix} \mu_k & 0 & \cdots 0 & 0 \\ 1 & \mu_k & \cdots 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots 1 & \mu_k \end{bmatrix}.$$

Proof. Let $\mu \in \sigma(H - cg^T)$; we need to prove that dim Ker $(H - cg^T - \mu I) = 1$. Suppose that t_1 and t_2 are solutions of $(H - cg^T - \mu I)x = 0$. Then

$$(H - \mu I)t_i = cg^T t_i = \xi_i c, \qquad i = 1, 2. \tag{10}$$

Now we distinguish two cases. (i) $\mu \in \sigma(H)$. Then $\xi_i = 0$, $i = 1, 2$, otherwise the systems (10) are inconsistent by Proposition 2. Therefore the vectors t_1 and t_2 are in Ker $(H - \mu I)$ which, by Lemma 1(ii), has dimension 1. Hence the vectors t_1 and t_2 are linearly dependent.

(ii) $\mu \notin \sigma(H)$. In this case $\xi_i \neq 0$, $i = 1, 2$ (otherwise the unique solution is the trivial one). Equation (10) becomes

$$(H - \mu I)(\xi_i^{-1} t_i) = c, \qquad i = 1, 2.$$

Since the system $(H - \mu I)x = c$ has unique solution, then t_1 and t_2 are linearly dependent. We conclude that dim Ker $(H - cg^T - \mu I) = 1$. □

Remark. Proposition 4 suggests that when there are in the set Ω eigenvalues with multiplicity greater than one, the matrix diag (μ_i) in (8) may be substituted by the matrix J defined in (9) to compute the matrix T. Now we discuss how to choose the vector u so that the matrix system

$$HT - cu^T = TJ \tag{11}$$

provides n consistent linear systems. Later, we shall see in Theorem 3 that the matrix T constructed from the solutions of these consistent systems is invertible.

Remark. To see the consistency of the system (11) as a function of the vector u, we partition this vector as

$$u = [u_{(1)}^T, u_{(2)}^T, \ldots, u_{(p)}^T]^T,$$

where the ith block of u^T is

$$u_{(i)} = [\beta_1^{(i)}, \beta_2^{(i)}, \ldots, \beta_{m(i)}^{(i)}]^T$$

and $m(i)$ is the multiplicity of μ_i. Also we write the matrix T by blocks as

$$T = [t_{(1)}, t_{(2)}, \ldots, t_{(p)}],$$

where the block $t_{(i)}$ is

$$t_{(i)} = [t_1^{(i)}, t_2^{(i)}, \ldots, t_{m(i)}^{(i)}]$$

for $i = 1, 2, \ldots, p$. The superscripts represent the corresponding eigenvalues. From the matrix system (11), using this notation, we obtain the following linear systems:

$$(H - \mu_i I)t_j^{(i)} = t_{j+1}^{(i)} + \beta_j^{(i)} c, \qquad j = 1, 2, \ldots, m(i), \qquad i = 1, 2, \ldots, p, \tag{12}$$

where $t_{m(i)+1}^{(i)} = 0$.

The systems (12) can be grouped into p subsystems corresponding to each eigenvalue μ_i, $i = 1, 2, \ldots, p$. Let us study the consistency of a subsystem (the same study can be done for the other subsystems). To simplify the notation, we drop the scripts (i) and work with the linear systems

$$(H - \mu I)t_j = t_{j+1} + \beta_j c, \qquad j = m, m-1, \ldots, 1, \tag{13}$$

where $t_{m+1} = 0$.

We consider two cases: (i) $\mu \notin \sigma(H)$. In this case $H - \mu I$ is nonsingular and the m linear systems (13) have unique solutions. The last system (i.e., $j = m$)

$$(H - \mu I)t_m = \beta_m c$$

is consistent for every scalar β_m. In particular, we obtain a nontrivial solution for $\beta_m = 1$. The remaining systems are consistent for any value of β_j; in particular, we shall take $\beta_j = 0, j = m-1, m-2, \ldots, 2, 1$.

(ii) $\mu \in \sigma(H)$. Suppose that the multiplicity in $\sigma(H)$ is r. Since (H, c) is controllable, by Lemma 1(ii) the r systems

$$(H - \mu I)t_m = 0,$$
$$(H - \mu I)t_{m-1} = t_m,$$
$$\cdots\cdots\cdots\cdots$$
$$(H - \mu I)t_{m-r+1} = t_{m-r+2}$$

are consistent.

(iia) In the case $m \leq r$, then the systems (13) are consistent for $\beta_m = \beta_{m-1} = \cdots = \beta_1 = 0$.

Exploiting the structure of the unreduced Hessenberg matrix, a particular solution of the systems (13) has the structure

(14) $$[t_1, t_2, \ldots, t_m] = \begin{bmatrix} \square & \\ & \ddots \\ 0 & \end{bmatrix} \begin{matrix} n-m \\ \hline \\ m \end{matrix}.$$

(iib) In the case $m > r$, the first r systems (13) are consistent for $\beta_m = \beta_{m-1} = \cdots = \beta_{m-r+1} = 0$. Then the next system of (13) to solve is

(15) $$(H - \mu I)t_{m-r} = t_{m-r+1} + \beta_{m-r}c,$$

or equivalently

(16) $$(H - \mu I)t_{m-r} - \beta_{m-r}c = t_{m-r+1},$$

which is equivalent to

(17) $$[H - \mu I, c] \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \\ \Gamma_{m-r} \end{bmatrix} = t_{m-r+1}.$$

By the remark of Lemma 1, we can also write

$$t_{m-r+1} = \delta_1\gamma_1 + \delta_2\gamma_2 + \cdots + \delta_{n-1}\gamma_{n-1} + \Gamma_{m-r}c,$$

where γ_i represents the ith column of the matrix $H - \mu I$, $i = 1, 2, \ldots, n-1$. Hence we can find a particular solution of (17) setting $\delta_n = 0$. Then a solution of (16) is $\beta_{m-r} = -\Gamma_{m-r}$ and

$$t_{m-r} = [\delta_1\ \delta_2 \cdots \delta_{n-1}\ 0]^T.$$

The other vectors $t_{m-r-1}, t_{m-r-2}, \ldots, t_1$ can similarly be obtained.

Again, by the structure of the matrix $[H - \mu I, c]$, we can get a partial solution of (13) with the following structure:

(18) $$[t_1, \ldots, t_r, t_{r+1}, \ldots, t_m] = \begin{bmatrix} \square & \\ & \ddots \\ 0 & \end{bmatrix} \begin{matrix} n-m \\ \hline \\ m \end{matrix}.$$

Example 2. Let

$$H = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \quad \text{and} \quad c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Suppose that we want to assign the spectrum $\Omega = \{1, 1, 1\}$. Since $1 \in \sigma(H)$, with multiplicity 1, we are in the case (iib), and we must solve the systems

$$(H - I)t_3 = 0,$$

$$(H - I)t_2 = t_3 + \beta_2 c,$$

$$(H - I)t_1 = t_2 + \beta_1 c.$$

A solution of the first system is $t_3 = [0, 2, 1]^T$. By (16) we get $\beta_2 = 2$, so a solution of the second system will be $t_2 = [2, 1, 0]^T$. With this solution and (17), we obtain $\beta_1 = -1$ and then one solution of the last system will be $t_1 = [1, 0, 0]^T$, and hence $u = [-1, 2, 0]^T$.

ALGORITHM II

This algorithm finds the vector f such that given the controllable pair (A, b) and the conjugated complex number set $\Omega = \{\mu_1, \mu_2, \ldots, \mu_p\}$, the matrix $A - bf^T$ has the spectrum Ω, where μ_i has multiplicity $m(i)$, $i = 1, 2, \ldots, p$. For the description of the algorithm it is assumed that the spectrum of A (and so the spectrum of H) and the multiplicities $r(1), r(2), \ldots, r(m)$ of its eigenvalues are known.

In the algorithm we represent by $u_{(s)}$ the block of the vector u that has coordinates from $m(1) + \cdots + m(s-1) + 1$ to $m(1) + \cdots + m(s-1) + m(s)$, and by $\gamma_j^{(i)}$, $j = 1, 2, \ldots, n$ the columns of matrix $H - \mu_i I$.

Stage I. Transform the pair (A, b) by orthogonal similarity Q to the pair (H, c), where H is an upper unreduced Hessenberg matrix and $c = [\alpha, 0, \ldots, 0]^T$, with $\alpha \neq 0$.

Stage II. For $i = 1, 2, \ldots, p$.

If $\mu_i \notin \sigma(H)$.

Solve the system

$$(H - \mu_i I)t_{m(i)}^{(i)} = c.$$

For $j = m(i) - 1, m(i) - 2, \ldots, 1$ solve the systems

$$(H - \mu_i I)t_j^{(i)} = t_{j+1}^{(i)}.$$

Form the $m(i)$-dimensional vector $u_{(i)} = [0, 0, \ldots, 0, 1]^T$.

Else

If $r(i) \geq m(i)$, solve the system for the nonzero solution

$$(H - \mu_i I)t_{m(i)}^{(i)} = 0.$$

For $j = m(i) - 1, m(i) - 2, \ldots, 1$ solve the systems

$$(H - \mu_i I)t_j^{(i)} = t_{j+1}^{(i)}.$$

Form the $m(i)$-dimensional vector $u_{(i)} = [0, 0, \ldots, 0]^T$.

Else solve the system for the nontrivial solution

$$(H - \mu_i I)t_{m(i)}^{(i)} = 0.$$

For $j = m(i) - 1, m(i) - 2, \ldots, m(i) - r(i) + 1$ solve the systems

$$(H - \mu_i I)t_j^{(i)} = t_{j+1}^{(i)}.$$

For $j = m(i) - r(i), m(i) - r(i) - 1, \ldots, 2, 1$, solve the systems

$$t_{j+1}^{(i)} = \delta_1^{(i)}\gamma_1^{(i)} + \delta_2^{(i)}\gamma_2^{(i)} + \cdots + \delta_{n-1}^{(i)}\gamma_{n-1}^{(i)} + \Gamma_j^{(i)}c.$$

Form the $m(i)$-dimensional vector

$$u_{(i)} = [u_1^{(i)}, u_2^{(i)}, \ldots, u_{m(i)}^{(i)}],$$

where $u_j^{(i)} = -\Gamma_j^{(i)}$, $j = 1, 2, \ldots, m(i) - r(i)$, and 0 elsewhere.

Form $t_{(j)} = [\delta_1^{(i)}, \delta_2^{(i)}, \ldots, \delta_{n-1}^{(i)}, 0]^T$.

Stage III. Form the vector $u = [u_{(1)}, u_{(2)}, \ldots, u_{(p)}]^T$.

Stage IV. Solve for g:

$$g^T T = u^T,$$

where

$$T = [t_{(1)}, t_{(2)}, \ldots, t_{(p)}]$$

and $t_{(i)}$ is

$$t_{(i)} = [t_1^{(i)}, t_2^{(i)}, \ldots, t_{m(i)}^{(i)}].$$

Stage V. Compute $f^T = g^T Q$.

The following theorem proves the validity of Algorithm II.

THEOREM 3. *Let (H, c) be a controllable pair. Let $\Omega = \{\mu_1, \mu_2, \ldots, \mu_p\}$ be an arbitrary conjugated complex number set. Suppose that the multiplicity of μ_i is $m(i)$ and $r(i)$ in Ω and $\sigma(H)$, respectively, $i = 1, 2, \ldots, p$. Then the solutions of the systems*

$$(H - \mu_i I)t_j^{(i)} = t_{j+1}^{(i)} + \beta_j^{(i)} c,$$

$$j = 1, 2, \ldots, m(i), \qquad i = 1, 2, \ldots, p$$

are linearly independent, where

(a) $t_{m(i)+1}^{(i)} = 0$.

(b) *If $\mu_i \notin \sigma(H)$, then $\beta_{m(i)}^{(i)} = 1$ and $\beta_j^{(i)} = 0$ for $j = m(i) - 1, m(i) - 2, \ldots, 2, 1$.*

(c) *If $\mu_i \in \sigma(H)$ and*

(c1) *$r(i) \geq m(i)$, then $\beta = 0$, $j = m(i), m(i) - 1, \ldots, 2, 1$.*

(c2) *$r(i) < m(i)$, then $\beta_j^{(i)} = 0$, $j = m(i), \ldots, m(i) - r(i) + 1$ and the others $\beta_j^{(i)}$ are calculated by* (16).

Proof. We need to prove that the matrix

$$T = [t_{(1)}, t_{(2)}, \ldots, t_{(p)}],$$

where

$$t_{(i)} = [t_1^{(i)}, t_2^{(i)}, \ldots, t_{m(i)}^{(i)}], \qquad i = 1, 2, \ldots, p$$

is nonsingular. We will proceed in three steps.

Step 1. $\mu_i \notin \sigma(H)$. We can proceed by induction on $m(i)$ as in the proof of Proposition 3. We start in this case in the reverse order because the first linear system we solve is $(H - \mu_i I)t_{m(i)}^{(i)} = c$. Then we obtain that the rank of

$$t_{(i)} = [t_1^{(i)}, t_2^{(i)}, \ldots, t_{m(i)}^{(i)}]$$

corresponding to this eigenvalue is $m(i)$.

Step 2. $\mu_i \in \sigma(H)$. Taking into account the structure of the particular solutions (14) and (18), the block

$$t_{(i)} = [t_1^{(i)}, t_2^{(i)}, \ldots, t_{m(i)}^{(i)}]$$

has rank $m(i)$.

Step 3. It remains to prove the independence of the blocks $t_{(i)}$ of T corresponding to the eigenvalues μ_i for all $i = 1, 2, \ldots, p$. For that, proceeding similarly to the proof

of Proposition 3, we obtain that T is nonsingular. In this case, again, the induction is applied in the reverse order as Step 1 of this theorem. □

6. Results. In this section we present numerical results obtained by our algorithms on two test problems: the Frank and Wilkinson matrices. These matrices were used in [3] as test matrices.

The accuracy obtained compares favorably with that obtained by Datta and Datta [4] in their algorithm. We performed experiments with Algorithms I and II. MATLAB [8] was used for implementing our algorithms.

Example 1. The Wilkinson matrix

$$W = \begin{bmatrix}
20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
20 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 20 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 20 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 20 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 20 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 20 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 20 & 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 6 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 1
\end{bmatrix}$$

has the following computed eigenvalues $\sigma(W)$:

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}.$$

For Algorithm I the spectrum Ω to be assigned is

$$\Omega = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30\}$$

with all multiplicities equal 1.

Then $\Omega \cap \sigma(W) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

The entries of the computed first row of the matrix $W - cg^T$ are

120.000051723387	0
−247.500243369342	0
396.000555157266	0
−450.450800870721	0
378.378800801735	0
−236.486822771660	0
108.108291772582	0
−34.459526679174	0
6.891906919606	0
−0.654731295163	0

and its eigenvalues are

1. 00000000000000	21. 00000000090067
2. 00000000000000	22. 00000004690938
3. 00000000000000	23. 00000088284309
4. 00000000000000	24. 00001038739562
5. 00000000000000	24. 99999594467154
6. 00000000000000	25. 99993877257182
7. 00000000000000	27. 00006858100040
8. 00000000000000	28. 00015437349975
9. 00000000000000	28. 99989411123250
10. 00000000000005	29. 99998862236297

The norm of the difference vector of the eigenvalues assigned and the eigenvalues of $W - cg^T$ is $O(2 \times 10^{-4})$.

For Algorithm II the spectrum Ω to be assigned is

$$\Omega = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

with all multiplicities equal 2.

Then $\Omega \cap \sigma(W) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

The entries of the computed first row of the matrix $W - cg^T$ are

−80.000000000259	0
−202.500000001586	0
−216.000000003493	0
−132.300000003715	0
−47.628000002097	0
−9.922500000639	0
−1.134000000102	0
−0.063787500008	0
−0.001417500000	0
−0.000007087500	0

and its eigenvalues are

0. 99999999501227	6. 00000000000000
1. 00000000000000	6. 00000001350166
2. 00000000000000	6. 99999999820177
2. 00000002630856	7. 00000000000000
2. 99999995228625	8. 00000000000000
3. 00000000000000	8. 00000000031483
4. 00000000000000	9. 00000000000000
4. 00000005701010	9. 00000000000223
4. 99999995710373	9. 99999999999965
5. 00000000000000	10. 00000000000005

The norm of the difference vector of the eigenvalues assigned and the eigenvalues of $W - cg^T$ is $O(9 \times 10^{-8})$.

Example 2. The Frank matrix

$$F = \begin{bmatrix} 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 11 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 10 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 9 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 8 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 7 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 6 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

has the following computed eigenvalues $\sigma(W)$

0.03102805830617	1. 55398870913215
0.04950743419656	3. 51185594858076
0.08122765574367	6. 96153308556712
0.14364652066476	12. 31107740086857
0.28474972048519	20. 19898864587716
0.64350531900585	32. 22889150157219

For Algorithm I the spectrum Ω to be assigned written in two columns is

$$\Omega = \left\{ \begin{array}{ll} 0.03102805830617 & 7 \\ 0.04950743419656 & 8 \\ 0.08122765574367 & 9 \\ 0.14364652066476 & 10 \\ 0.28474972048519 & 11 \\ 0.64350531900585 & 12 \end{array} \right\}$$

with all multiplicities equal 1.

$$\text{Then } \Omega \cap \sigma(F) = \left\{ \begin{array}{l} 0.03102805830617 \\ 0.04950743419656 \\ 0.08122765574367 \\ 0.14364652066476 \\ 0.28474972048519 \\ 0.64350531900585 \end{array} \right\}.$$

The entries of the computed first row of the matrix $F - cg^T$ are

−7.766335249261	−162.577346221521
−35.308642532913	−161.397773124918
−62.761983253348	−146.348974404763
−95.091914484174	−119.095129320986
−126.374891086865	−82.976983907449
−150.199014439408	−42.265088670901

and its eigenvalues

0.03102815917842	6. 99999999999358
0.04950736946874	8. 00000000005549
0.08122767023319	8. 99999999986791
0.14364651222413	10. 00000000014808
0.28474972064216	10. 99999999992647
0.64350531900500	11. 99999999999584

The norm of the difference vector of the eigenvalues assigned and the eigenvalues of $F - cg^T$ is $O(1.5 \times 10^{-7})$.

For Algorithm II the spectrum Ω to be assigned is

$$\Omega = \left\{ \begin{array}{r} 1.\ 55398870913215 \\ 3.\ 51185594858076 \\ 6.\ 96153308556712 \\ 12.\ 31107740086857 \\ 20.\ 19898864587716 \\ 32.\ 22889150157219 \end{array} \right\}$$

with all multiplicities equal 2.

$$\text{Then } \Omega \cap \sigma(F) = \left\{ \begin{array}{r} 1.\ 55398870913215 \\ 3.\ 51185594858076 \\ 6.\ 96153308556712 \\ 12.\ 31107740086857 \\ 20.\ 19898864587716 \\ 32.\ 22889150157219 \end{array} \right\}.$$

The entries of the computed first row of the matrix $F - cg^T$ are

87.53267058319	2379.73451460592
−255.22818519940	−2175.29690462507
656.26133592782	1757.99622044554
−1193.96118935235	−1255.19608884964
1806.75457380974	809.50021222572
−2234.69607175733	−478.49127345807

and its eigenvalues are

1. 55398866499479
1. 55398875326951
3. 51185594858072 − 0.00000024596755 i
3. 51185594858072 + 0.00000024596755 i
6. 96153308556700 − 0.00000063665662 i
6. 96153308556700 + 0.00000063665662 i
12. 31107740086755 − 0.00000160986180 i
12. 31107740086755 + 0.00000160986180 i
20. 19898650094308
20. 19899079082090
32. 22888869444581
32. 22889430868959

The norm of the difference vector of the eigenvalues assigned and the eigenvalues of $F - cg^T$ is $O(5 \times 10^{-6})$.

Acknowledgments. The authors would like to thank B. N. Datta and the referees for several useful comments that improved the first version of this paper.

REFERENCES

[1] M. ARNOLD AND B. N. DATTA, *An algorithm for multi-input eigenvalue assignment problem*, IEEE Trans. Auto. Control, 35 (1990), pp. 1149–1152.

[2] B. N. DATTA, *An algorithm to assign eigenvalues in a Hessenberg matrix: Single input case*, IEEE Trans Auto. Control, 5 (1987), 414–417.

[3] ———, *Parallel and large-scale matrix computations in control: some ideas*, Linear Algebra Appl., 121 (1989), pp. 243–264.

[4] B. N. DATTA AND K. DATTA, *Efficient parallel algorithms for controllability and eigenvalue assignment problems*, Proc. 25th IEEE Conf. Decision and Control, Athens, Greece, 1986, pp. 1611–1616.

[5] J. KAUTSKY, N. NICHOLS, AND P. VANDOOREN, *Robust pole assignment in linear feedback*, Internat. J. Control, 41 (1985), pp. 1129–1155.

[6] M. KONSTANTINOV, P. PETKOV, AND N. CHRISTOV, *Synthesis of linear systems with desired equivalent form*, J. Assoc. Comput. Mach., 6 (1980), pp. 27–35.

[7] G. MIMINIS AND C. C. PAIGE, *An algorithm for pole-assignment of time invariant linear system.* Internat. J. Control, 35 (1981), pp. 130–138.

[8] C. MOLER, J. LITTLE , S. BANGERT, AND S. KLEIMAN, *PC-MATLAB User's Guide*, The MathWorks Inc., Sherborn, MA, 1986.

[9] R. V. PATEL AND P. MISRA, *Numerical algorithm for eigenvalue assignment by state feedback.* Proc. IEEE 72, 12 (1984), pp. 1755–1764.

[10] W. M. WONHAM, *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York, 1979.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 408–417, April 1994

005

BACKWARD ERROR ESTIMATES FOR TOEPLITZ SYSTEMS*

J. M. VARAH†

Abstract. Given a computed approximate solution $\bar{x}$ to $Ax = b$, it is interesting to find nearby systems with $\bar{x}$ as exact solution and that have the same structure as A. This paper shows that the distance to these nearby structured systems can be much larger than for the corresponding general perturbation for general and symmetric Toeplitz systems. In fact, even the correctly rounded solution $\hat{x}$ may require a structured perturbation with terms as large as $\|\hat{x}\|$ times the machine precision.

Key words. backward error, stability, Toeplitz system

AMS subject classifications. 65F05, 65F35

1. Introduction. Given the linear system $Ax = b$ and a computed solution $\bar{x}$, it is interesting to find nearby systems for which $\bar{x}$ is the exact solution. That is, to find δA and δb such that

$$(A + \delta A)\bar{x} = b + \delta b \tag{1.1}$$

with δA and δb small. If we define the associated residual vector

$$r = r(\bar{x}) = b - A\bar{x},$$

then (1.1) becomes

$$(\delta A)\bar{x} = r + \delta b. \tag{1.2}$$

If we consider general perturbations δA and δb, then conditions (1.1) or (1.2) do not specify them fully, and we must impose additional conditions (such as minimizing some measure of the size of δA and δb). If however the matrix A has some special form, and we are interested in maintaining this form in the allowable perturbations, then the solution of (1.1) or (1.2) becomes more complicated. In this paper, we consider the case of A being of Toeplitz form. This issue of restricted perturbations for structured systems has also been considered by Higham and Higham [7] who provide a framework for dealing with general structured perturbations and define the notion of structured condition numbers. They use Toeplitz matrices as one of their examples, and the work presented here should be considered as an extension of their work.

Note that the scaling of the problem is important; we assume throughout that $\|A\| \simeq 1$ and $\|b\| \simeq 1$, so that ill conditioning of A is reflected in $\|x\|$ being (possibly) large, but not small. We assume Hölder vector norms and corresponding subordinate matrix norms. Then, in fact,

$$\frac{\|b\|}{\|A\|} \leq \|x\| \leq \kappa(A)\left(\frac{\|b\|}{\|A\|}\right),$$

where $\kappa(A) = \|A\|\,\|A^{-1}\|$ is the (standard) condition number of A in any norm. We also assume throughout that A is nonsingular.

* Received by the editors September 30, 1991; accepted for publication (in revised form) September 23, 1992.

† Computer Science Department, University of British Columbia, Vancouver, British Columbia, Canada V6T 1W5 (varah@cicsr.ubc.ca).

To get some sense of the size of the residual $r(\bar{x})$, it is useful to consider what happens in the best possible case, when $\bar{x} = \hat{x}$, the correctly rounded solution. Since

$$\hat{x}_i = x_i(1 + \eta_i),$$

where $|\eta_i| \leq \eta$ = machine round-off level, we can write

$$\hat{x} = (I + D_\eta)x,$$

and then

$$r = r(\hat{x}) = b - A\hat{x} = -AD_\eta x, \tag{1.3}$$

giving

$$\frac{\|r\|}{\|x\|} \leq \eta\|A\|.$$

In particular, $\|r\|/\|x\| \simeq \eta$ independent of the solution x. We also remark that (1.3) implies

$$|r| \leq \eta|A|\,|x|, \tag{1.4}$$

where the inequality is meant to be taken componentwise.

Thus, the most we can expect for a computed solution $\bar{x}$ is that $\|r(\bar{x})\|/\|\bar{x}\| \simeq \eta$. Such behaviour occurs, for example, with solutions computed by Gaussian elimination with partial pivoting (generally) or with Cholesky factorization on positive definite symmetric systems. As a result, it is not generally appropriate to solve (1.1) or (1.2) by taking $\delta A = \mathrm{O}$. This gives $\delta b = -r$, which for $\|\bar{x}\|$ large means a large backward error $\|\delta b\| = \|r\| \simeq \eta\|\hat{x}\|$ even for the correctly rounded solution. Instead, one attempts to find solutions δA, δb with

$$\max(\|\delta A\|, \|\delta b\|) \leq c\left(\frac{\|r\|}{\|x\|}\right) \tag{1.5}$$

for c some constant close to 1.

Allowing general perturbations δA, δb, one can indeed find solutions satisfying (1.5), as has been known for some time, and we review this material in §2. See, also, the excellent survey paper by Higham [6]. Then in §3 we consider symmetric Toeplitz perturbations of symmetric Toeplitz matrices. We are motivated to do this from interest in the stability properties of special methods available for solving Toeplitz systems, such as the Levinson method (see Golub and Van Loan [5, p. 183], for example). One might hope that the computed solution obtained from such a method is the exact solution of a nearby Toeplitz system. Indeed, Bunch [3] refers to this behaviour as "strong stability." However, we find that under these restrictions, the perturbations δA and δb satisfy not (1.5), but

$$\max(\|\delta A\|, \|\delta b\|) \leq c\|r\|. \tag{1.6}$$

This means that for ill-conditioned Toeplitz systems, computed solutions (even correctly rounded solutions) satisfy Toeplitz systems that are as much as $\kappa(A)\eta$ away from the original system. We illustrate this behaviour with some numerical examples in §4. Finally in §5, we discuss the same problem for general Toeplitz systems, where the conclusion is similar.

2. General backward error. Consider the basic system (1.2). The equations decouple and we consider the first one in detail:

$$((\delta A)_{11} \cdots (\delta A)_{1n})\bar{x} = r_1 + (\delta b)_1.$$

If $r_1 = 0$, we can take all unknowns = 0. So assume that $r_1 \neq 0$ and let

$$(\delta A)_{1i} = r_1 e_i z_i, \qquad (\delta b)_1 = r_1 f y,$$

where $\{e_i\}_1^n$ and f are fixed scaling factors and $\{z_i\}_1^n$ and y are to be determined. Then the defining equation can be written

$$(e_1\bar{x}_1 \cdots e_n\bar{x}_n - f)\begin{pmatrix} z_1 \\ \vdots \\ z_n \\ y \end{pmatrix} = 1 \tag{2.1}$$

or $u^T v = 1$.

Normally, besides satisfying (2.1), we want to make the perturbations as small as possible in some sense, which amounts to minimizing $\|v\|$ for some norm. In particular, if we use a Hölder norm $\|v\|_q$, with $\frac{1}{p} + \frac{1}{q} = 1$,

$$1 = |u^T v| \leq \|u\|_p \|v\|_q$$

for any u and v satisfying (2.1), and

$$\min_v \|v\|_q = \frac{1}{\|u\|_p}.$$

We could use $p = q = 2$, but it is more natural to use $p = 1$, $q = \infty$, giving

$$\min_v \|v\|_\infty = \frac{1}{\|u\|_1} = \frac{1}{f + \sum e_i |\bar{x}_i|}, \tag{2.2}$$

which is attained by using v with $v_i = \operatorname{sgn}(u_i)/\|u\|_1$.

This max norm solution translates into

$$(\delta A)_{1i} = \frac{\pm r_1 e_i}{f + \sum e_i |\bar{x}_i|}, \qquad \delta_1 = \frac{\pm r_1 f}{f + \sum e_i |\bar{x}_i|},$$

which replicates the Oettli–Prager result [8] for general scaling factors E and f. One particular case deserves special mention: $e_i = \|A\|, f = \|b\|$. Then

$$(\delta A)_{1i} = \frac{\pm r_1 \|A\|}{\|b\| + \|A\| \, \|\bar{x}\|_1}, \qquad (\delta b)_1 = \frac{\pm r_1 \|b\|}{\|b\| + \|A\| \, \|\bar{x}\|_1},$$

and similarly for the other rows. Note that in this case, we do obtain

$$\max(\|\delta A\|, \|\delta b\|) \leq c\left(\frac{\|r\|}{\|\bar{x}\|}\right)$$

as predicted in §1, where c involves $\|A\|$ and $\|b\|$, and is close to 1.

3. The symmetric Toeplitz case. Now assume A is a symmetric Toeplitz matrix, and that we want δA to be symmetric and Toeplitz as well. That is,

$$\delta A = \begin{bmatrix} \varepsilon_0 & \varepsilon_1 & \cdots & \varepsilon_{n-1} \\ \varepsilon_1 & \varepsilon_0 & \cdots & \varepsilon_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n-1} & \cdots & \cdots & \varepsilon_0 \end{bmatrix}.$$

In the defining equation (1.2), the key observation (noted also in [7]) is to rewrite $(\delta A)\bar{x}$ as $X\varepsilon$, where ε is the vector $(\varepsilon_0, \ldots, \varepsilon_{n-1})^T$ and (for n odd):

$$X = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \bar{x}_3 & \cdots & \cdots & \cdots & \bar{x}_{n-1} & \bar{x}_n \\ \bar{x}_2 & (\bar{x}_1 + \bar{x}_3) & \bar{x}_4 & \cdots & \cdots & \cdots & \bar{x}_n & 0 \\ \bar{x}_3 & (\bar{x}_2 + \bar{x}_4) & (\bar{x}_1 + \bar{x}_5) & \cdots & \cdots & & & \vdots \\ \vdots & & & \ddots & & & & \vdots \\ \bar{x}_{(n+1)/2} & \cdots & \cdots & \cdots & (\bar{x}_1 + \bar{x}_n) & 0 & \cdots & 0 \\ \vdots & & & & & \ddots & & \vdots \\ \bar{x}_{n-1} & (\bar{x}_{n-2} + \bar{x}_n) & \bar{x}_{n-3} & \cdots & \cdots & \cdots & \bar{x}_1 & 0 \\ \bar{x}_n & \bar{x}_{n-1} & \bar{x}_{n-2} & \cdots & \cdots & \cdots & \bar{x}_2 & \bar{x}_1 \end{bmatrix}. \tag{3.1}$$

For n even, the middle row and column are not present. Note that each $\bar{x}_i$ appears once in each row, and note that $\|X\|_\infty = \|\bar{x}\|_1$. Also, note that X can be singular: if $\sum \bar{x}_i = 0$ for example, then $Xe = 0$ for $e = (1, 1, \ldots, 1)^T$. On the other hand, $\bar{x} = (1, 0, \ldots, 0)^T$ gives $X = I$, the identity matrix.

Using this matrix, the defining equation (1.2) reads

$$X\varepsilon - \delta = r,$$

where $\delta = \delta b$, which is n equations in the $2n$ unknowns ε and δ. This can also be expressed as

$$(X - I)\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix} = r \quad \text{or} \quad Gv = r. \tag{3.2}$$

Again, we would like the perturbation to be as small as possible, and thus we are led to the constrained optimization problem

$$\min_v \|v\| \quad \text{with } Gv = r.$$

Note that we can include componentwise scaling factors in ε, δ by using diagonal scaling factors D_e, D_f giving $G = (XD_e - D_f)$.

Again, the most natural norm is $\|v\|_\infty$ giving a constrained Chebyshev optimization problem, which can be rewritten using $\delta = X\varepsilon - r$ as the overdetermined discrete Chebyshev problem

$$\min_\varepsilon \left\| \begin{pmatrix} X \\ I \end{pmatrix}\varepsilon - \begin{pmatrix} r \\ 0 \end{pmatrix} \right\|_\infty.$$

This problem is difficult to solve explicitly, although algorithms have been developed to solve individual cases (see [1] or [2]). The basic question here is whether the solution $\|v\|_\infty \simeq \|r\|/\|\bar{x}\|$.

The following example shows that this is not always true, and the consequence is that even for a rounded solution $\hat{x}$, the closest perturbed symmetric Toeplitz system with $\hat{x}$ as exact solution can be $\eta\|\hat{x}\|$ away.

Example.

$$A = \begin{bmatrix} 1 & 1-\mu & 1-\alpha \\ 1-\mu & 1 & 1-\mu \\ 1-\alpha & 1-\mu & 1 \end{bmatrix}$$

with μ and α small and positive. (In fact, μ is not crucial in what follows.) One eigenvalue $\lambda_1 = \alpha$ with corresponding eigenvector $(1, 0, -1)^T$.

Hence when $b = (1, 0, -1)^T$, the solution $x = \frac{1}{\alpha}(1, 0, -1)^T$. Now take the rounded solution $\hat{x} = \frac{1}{\alpha}(1 + \eta_1, 0, -(1 + \eta_3))^T$. A short calculation gives, using $\hat{\eta} = \eta_1 - \eta_3$,

$$r(\hat{x}) = -\frac{\hat{\eta}}{\alpha}(1, 1, 1)^T + 0(\eta)$$

and

$$X = \frac{1}{\alpha}\begin{bmatrix} 1 + \eta_1 & 0 & -(1 + \eta_3) \\ 0 & \hat{\eta} & 0 \\ -(1 + \eta_3) & 0 & 1 + \eta_1 \end{bmatrix}.$$

Thus the equations $Gv = r$ are (to first order in η):

$$(1 + \eta_1)v_1 - (1 + \eta_3)v_3 - \alpha v_4 = -\hat{\eta},$$

$$\hat{\eta}v_2 - \alpha v_5 = -\hat{\eta},$$

$$-(1 + \eta_3)v_1 + (1 + \eta_1)v_3 - \alpha v_6 = -\hat{\eta}.$$

It is easy to see that the minimax solution of these equations has all $|v_i| = |\hat{\eta}|/(\alpha + \hat{\eta})$. In fact,

$$v_1 = v_2 = v_3 = -\hat{\eta}/(\alpha + \hat{\eta}), \qquad v_4 = v_5 = v_6 = \hat{\eta}/(\alpha + \hat{\eta}).$$

Hence, for this example, we have explicitly

$$\frac{\|\hat{x} - x\|}{\|x\|} \simeq \eta, \qquad \frac{\|r(\hat{x})\|}{\|x\|} \simeq \eta,$$

$$\|\delta A\| = \|\delta b\| \simeq \frac{\eta}{\alpha}.$$

Also, one can find a much closer *general* perturbation δA with $(A + \delta A)\hat{x} = b$; in fact

$$\delta A = \frac{-\hat{\eta}}{2}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} + 0(\eta^2).$$

Later on, we provide numerical evidence of the same behaviour, using rounded solutions, and solutions computed by the Cholesky and Levinson algorithms. All exhibit the same behavior as above.

Now return to the basic problem (3.2). Although an explicit form of the solution in the max norm is not available, we can find an approximate solution by solving the corresponding constrained least squares problem

$$\min \|v\|_2 \quad \text{with } Gv = r.$$

If $M_2 = \min_v \|v\|_2$ and $M_\infty = \min_v \|v\|_\infty$, then

$$\frac{1}{\sqrt{n}} M_2 \leq M_\infty \leq M_2,$$

and hence the solutions differ in value by at most a factor $\sqrt{n}$.

Moreover, the constrained least squares solution has a simple form

$$v = G^T z, \qquad Gv = r,$$

where z is the vector of Lagrange multipliers (see, for example, [4, p. 156]). That is,

$$(I + XX^T)z = r, \qquad \begin{pmatrix} \varepsilon \\ \delta \end{pmatrix} = \begin{pmatrix} X^T z \\ -z \end{pmatrix}. \tag{3.3}$$

Alternatively, one can solve the overdetermined problem

$$\min_{\varepsilon} \left\| \begin{pmatrix} X \\ I \end{pmatrix} \varepsilon - \begin{pmatrix} r \\ 0 \end{pmatrix} \right\|_2,$$

with solution $(I + X^T X)\varepsilon = X^T r$ and $\delta = X\varepsilon - r$.

Note that from this formulation, it easily follows that if $r(\bar{x})$ is such that $r^T X = 0$, then $\varepsilon = 0$ and $\delta = -r$, which gives a solution (as we mentioned earlier) that is unacceptably large when $\|\bar{x}\|$ is large. However, this is by no means the only troublesome case, as we see below.

Of course, the more acceptable computational method for finding v, at least in cases where X is ill conditioned, is to use the QR decomposition (Golub and Van Loan [5, p. 567]):

(i) $$\text{set } G^T = \begin{pmatrix} X^T \\ -I \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = (Q_1 \mid Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix},$$

(ii) $$\text{find } w = Q^T v \text{ via } \begin{Bmatrix} R^T w^{(1)} = r \\ w^{(2)} = 0 \end{Bmatrix},$$

(iii) $$\text{find } v = Qw = Q_1 w^{(1)}.$$

An even more explicit formulation of the minimum least squares solution can be derived from (3.3) using the singular value decomposition (SVD).

THEOREM 1. *Let $X = UDV^T$ be the* SVD *of the matrix X in* (3.1), *and let $r = \sum \beta_i u^{(i)}$ be the expansion of $r = r(\bar{x})$ in the singular vectors $\{u^{(i)}\}$. Then the solution z to* (3.3) *has the expansion*

$$z = \sum \left(\frac{\beta_i}{1 + \sigma_i^2} \right) u^{(i)},$$

and the minimal least squares solution v has

$$\|v\|_2^2 = z^T r = \sum \frac{\beta_i^2}{1 + \sigma_i^2}.$$

Proof. The proof follows by substitution.

Theorem 1 above gives a reformulation of the minimum restricted perturbation in the ℓ_2 sense, which makes $\bar{x}$ exact. This basic result can be applied in various ways as we now explore.

THEOREM 2. *Suppose that the computed solution $\bar{x}$ is such that the* SVD *coefficients $\{\beta_i\}$ of $r(\bar{x})$ satisfy*

$$|\beta_i| \leq c\sqrt{1 + \sigma_i^2}\,\eta$$

for c not much bigger than 1. *Then $\|v\|_2 \leq c\sqrt{n}\,\eta$, and hence the minimum perturbation is close to the round-off level.*

Proof. Again, the proof follows by direct substitution.

To clarify the situation, it helps to separate the cases where the solution x does and does not reflect the ill condition of A. When x does not, that is, when $\|x\| \simeq 1$, Theorem

2 applies as long as the computed solution $\bar{x}$ produces $\|r(\bar{x})\|$ close to the round-off level η. Since $\|X\|_\infty = \|\bar{x}\|_1$, $1 \simeq \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$, and the requirements of Theorem 2 hold since the $\{\beta_i\}$ are all at the round-off level. Note that the near singularity of X is immaterial in this case.

Now consider large $\|x\|$, and recall that $\|x\|$ can be as large as $\kappa(A)$. Assume that $\bar{x}$ produces a residual with $\|r(\bar{x})\| \simeq \eta\|\bar{x}\|$, as occurs with $\bar{x} = \hat{x}$, the correctly rounded solution. Then *some* $\{\beta_i\}$ are also this large, and *some* $\{\sigma_i\}$ are as large as $\|\bar{x}\|$. If the large $\{\beta_i\}$ occur *only* for large $\{\sigma_i\}$, then the conditions required for Theorem 2 still hold. However, experimental evidence with ill-conditioned symmetric positive definite Toeplitz systems indicates that typically *all* $\beta_i \simeq \eta\|\bar{x}\|$, whether $\bar{x}$ is the correctly rounded solution, the Cholesky solution, or the solution computed using the Levinson algorithm. We present some examples in the next section. We can quantify the situation as follows: Assume that

$$\beta_i = c_i\eta\|\bar{x}\| \text{ with } |c_i| \leq \bar{c},$$
$$\sqrt{1+\sigma_i^2} = d_i\|\bar{x}\| \quad \text{for } i = 1, \ldots, n.$$

Note that $1 \simeq d_1 \geq d_2 \geq \cdots \geq d_n \geq 1/\|\bar{x}\|$.

THEOREM 3. *With the above definitions of* $\{c_i\}$ *and* $\{d_i\}$,

$$\min \|v\|_2 = \eta\left(\sum_1^n \frac{c_i^2}{d_i^2}\right)^{1/2}.$$

Thus

$$\eta|c_n|\frac{\|\bar{x}\|}{\sqrt{1+\sigma_n^2}} \leq \min\|v\|_2 \leq \eta n\bar{c}\frac{\|\bar{x}\|}{\sqrt{1+\sigma_n^2}}. \tag{3.4}$$

Proof. Again, the proof follows by direct substitution.

Thus, the actual minimum symmetric Toeplitz perturbation is very dependent on the particular problem and can vary in size from η to $\kappa(A)\eta$. To realize a large perturbation, the data vector b must give rise to a solution x that reflects the ill condition of A and that also results in an appreciable spread in the singular values of the matrix X.

4. Some numerical results. The first example is the 3×3 matrix from §3:

$$A = \begin{bmatrix} 1 & 1-\mu & 1-\alpha \\ 1-\mu & 1 & 1-\mu \\ 1-\alpha & 1-\mu & 1 \end{bmatrix}.$$

We took $\alpha = 10^{-6}$, $\mu = \frac{\alpha}{3}$. A has then two eigenvalues near 10^{-6}. For various data vectors b, we computed solutions to $Ax = b$ as follows:

(i) $\bar{x}$ = Cholesky solution,

(ii) $\bar{\bar{x}}$ = Levinson solution,

(iii) $\hat{x}$ = Correctly rounded solution, obtained from $\bar{x}$ using double precision iterative refinement.

Working precision was long precision on an IBM mainframe, with special routines for "double long" calculations, so that $\eta \simeq 10^{-16}$. For each approximate solution, we computed the residual r, the minimal least squares solution $v = \binom{\varepsilon}{\delta}$ from (3.3), the singular values $\sigma_i(X)$, and the coefficients $\beta_i(r)$.

In Case 1 below, b reflects the ill condition of A and $\|x\|$ is large. For each approximate solution, the closest symmetric Toeplitz system with that exact solution is roughly

$\eta\|x\|$ away. In Case 2, however, $\|x\|$ is near 1 and, even though X is singular, the smallest perturbation is now close to the round-off level η.

Case 1.

$$b = (-0.72, 0.55, 0.22)^T, \qquad \|x\|_\infty = 4.8 \times 10^6,$$

$$\sigma_i(X)\text{: } 1.0 \times 10^7, \quad 2.9 \times 10^6, \quad 0.32.$$

	$\|r\|_\infty$	$\|v_{\min}\|_2$	β
$\bar{x}$	1.0×10^{-10}	4.7×10^{-11}	$-.25 \times 10^{-10}$, $-.14 \times 10^{-9}$, $.50 \times 10^{-10}$.
$\bar{\bar{x}}$	3.2×10^{-10}	3.9×10^{-11}	$-.15 \times 10^{-9}$, $-.33 \times 10^{-9}$, $-.41 \times 10^{-10}$.
$\hat{x}$	1.8×10^{-10}	8.7×10^{-11}	$-.86 \times 10^{-10}$, $.28 \times 10^{-9}$, $-.92 \times 10^{-10}$.

Case 2.

$$b = (-0.58, -.58, 0.58)^T \|x\|_\infty = 0.19,$$

$$\sigma_i(X)\text{: } .60, \quad .21, \quad .84 \times 10^{-9}.$$

	$\|r\|_\infty$	$\|v_{\min}\|_2$	β
$\bar{x}$	1.0×10^{-17}	1.4×10^{-17}	$-.16 \times 10^{-16}$, $.47 \times 10^{-18}$, $.93 \times 10^{-18}$.
$\bar{\bar{x}}$	2.4×10^{-17}	3.2×10^{-17}	$-.37 \times 10^{-16}$, $.37 \times 10^{-17}$, $.31 \times 10^{-17}$.
$\hat{x}$	2.1×10^{-17}	3.1×10^{-17}	$.36 \times 10^{-16}$, $-.40 \times 10^{-17}$, $-.88 \times 10^{-26}$.

As a second example, consider the prolate matrix (see Slepian [9]) of order 11 with $a_{ij} = \gamma_{|j-i|}$,

$$\gamma_k = \frac{\sin(\pi k/2)}{\pi k}, \qquad \gamma_0 = \frac{1}{2}.$$

This matrix is positive definite symmetric and Toeplitz and its smallest eigenvalue λ_1 is near 10^{-7}. Generally, the behaviour with various data vectors b is similar to that of the first example, and we mention only one case where b is the eigenvector corresponding to λ_1. For this case (and for b = other eigenvectors as well), the matrix X is near singular, giving more than enough spread in the singular values to result in a minimum perturbation of nearly $\kappa(A)\eta$. The β-coefficients are all near 10^{-10}.

$$b = \text{first eigenvector}, \qquad \|x\|_\infty = 5.4 \times 10^6,$$

$$\sigma_i(X)\text{: } 3.3 \times 10^7, \ldots, \qquad 7.3 \times 10^{-8}.$$

	$\|r\|_\infty$	$\|v_{\min}\|_2$
$\bar{x}$	3.1×10^{-10}	2.5×10^{-10}.
$\bar{\bar{x}}$	1.5×10^{-10}	1.1×10^{-10}.
$\hat{x}$	3.7×10^{-11}	2.1×10^{-11}.

5. The general Toeplitz case. If A is a general (unsymmetric) Toeplitz matrix, and we allow similar perturbations δA, then

$$\delta A = \begin{bmatrix} \varepsilon_0 & \varepsilon_1 & \cdots & \varepsilon_{n-1} \\ \varepsilon_{-1} & \varepsilon_0 & \cdots & \varepsilon_{n-2} \\ \vdots & \ddots & \ddots & \vdots \\ \varepsilon_{-n+1} & \cdots & \varepsilon_{-1} & \varepsilon_0 \end{bmatrix},$$

giving $(2n-1)$ parameters $\{\varepsilon_i\}$, plus n more in δb. The defining equation (1.2) can again be rewritten

$$X\varepsilon - \delta = r \qquad \text{or} \quad (X-I)\begin{pmatrix}\varepsilon\\ \delta\end{pmatrix} = r \quad \text{or} \quad Gv = r,$$

as in (3.2), but now we have n equations in $(3n-1)$ unknowns, and X is the $n \times (2n-1)$ matrix

$$\begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_n & 0 & \cdots & 0 \\ 0 & \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_n & & \vdots \\ \vdots & \ddots & \ddots & & & \ddots & 0 \\ 0 & \cdots & 0 & \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_n \end{bmatrix}. \tag{5.1}$$

(We have reversed the order of the equations to make X look triangular.)

The situation now is somewhat different from the symmetric case discussed in §3, although the theorems still apply. There are more parameters and thus fewer restrictions on the perturbations. This is manifested in X (or really the rows of X) being better conditioned than the X of §3, making it harder to effect a large minimum perturbation.

It is instructive to compare results with those of §3. Take again the prolate matrix with $n = 11$; for the correctly rounded solution $\hat{x}$, the results are as follows:

$$b = \text{first eigenvector}, \qquad \|\hat{x}\|_\infty = 5.4 \times 10^6,$$
$$\sigma_i(X)\text{: } 2.4 \times 10^7, \ldots, 3.3 \times 10^3.$$

$\|r\|_\infty$	$\|v_{\min}\|_2$
3.7×10^{-11}	1.0×10^{-14}

(In all the examples quoted here, the $\{\beta_i\}$ were all in magnitude close to $\eta\|\hat{x}\|$.) Note that the spread in X's singular values is much less here with a corresponding decrease in the size of the perturbation.

One can increase the perturbation by taking a larger n. For $n = 15$, the results are

$$b = \text{first eigenvector}, \qquad \|\hat{x}\|_\infty = 4.8 \times 10^9,$$
$$\sigma_i(X)\text{: } 2.6 \times 10^{10}, \ldots, 1.0 \times 10^5.$$

$\|r\|_\infty$	$\|v_{\min}\|_2$
2.0×10^{-7}	3.3×10^{-12}

In both examples, note that the minimum ℓ_2 perturbation lies *between* the round-off level and the size of the residual in accordance with (3.4). In particular, to effect a minimum perturbation as large as $\kappa(A)\eta$, the vector $\bar{x}$ must yield a matrix X with $\kappa(X)$ approaching $\kappa(A)$.

Acknowledgment. The author would like to thank the (anonymous) referees for suggestions that significantly improved the paper.

REFERENCES

[1] I. BARRODALE AND C. PHILLIPS, *Algorithm* 495: *Solution of an overdetermined system of linear equations in the Chebychev norm*, ACM Trans. Math. Software, 1 (1975), pp. 264–270.

[2] R. H. BARTELS, A. R. CONN, AND C. CHARALAMBOUS, *On Cline's direct method for solving overdetermined linear systems in the ℓ_∞ sense*, SIAM J. Numer. Anal., 15 (1978), pp. 255–270.

[3] J. R. BUNCH, *The weak and strong stability of algorithms in numerical linear algebra*, Linear Algebra Appl., 88/89(1987), pp. 49–66.

[4] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, New York, 1981.

[5] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[6] N. J. HIGHAM, *How accurate is Gaussian elimination?*, in Numerical Analysis 1989, Proc. 13th Dundee Conference, D. F. Griffiths and G. A. Watson, eds., Longman Scientific and Technical, Harlow, UK, 1990, pp. 137–154.

[7] D. J. HIGHAM AND N. J. HIGHAM, *Backward error and condition of structured linear systems*. SIAM J. Matrix Anal. Appl., 13 (1992), pp. 162–175.

[8] W. OETTLI AND W. PRAGER, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Numer. Math., 6 (1964), pp. 405–409.

[9] D. SLEPIAN, *Prolate spheroidal wave functions, Fourier analysis, and uncertainty* V: *the discrete case*, Bell System Tech. J., 57 (1978), pp. 1371–1430.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 418–450, April 1994

006

MOORE–PENROSE INVERSION OF SQUARE TOEPLITZ MATRICES*

GEORG HEINIG† AND FRANK HELLINGER†

Abstract. Fast algorithms for the computation of the Moore–Penrose inverse A^+ of a square Toeplitz matrix A are constructed based on Bezoutian representations of A^+. Two approaches are presented. The first approach is a recursion of the nested submatrices, the second approach uses generalized inverses of A that are Bezoutians and a global formula.

Key words. Toeplitz matrix, Moore–Penrose inverse, fast algorithm

AMS subject classifications. 15A09, 47B35, 65F20

1. Introduction.

In this paper we consider square Toeplitz matrices

$$A = \begin{bmatrix} a_0 & a_{-1} & \dots & a_{-n+1} \\ a_1 & a_0 & \dots & a_{-n+2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \dots & a_0 \end{bmatrix} \tag{1.1}$$

with complex entries $a_i \in \mathbb{C}$. The main aim is the construction of fast algorithms for the computation of the Moore–Penrose inverse (MPI) of A. Algorithms presented in this paper are based on a matrix representation for the MPI of A that was established for the Hankel analogues in the authors' paper [8]. In this sense, this paper is a continuation of [8]; however, we tried to make the presentations fairly self-contained. Recall that the MPI of a matrix C is a matrix C^+ defined by the following four relations:

$$CC^+C = C, \tag{1.2a}$$

$$C^+CC^+ = C^+, \tag{1.2b}$$

$$(CC^+)^* = CC^+, \tag{1.2c}$$

$$(C^+C)^* = C^+C. \tag{1.2d}$$

Here the asterisk denotes the conjugate complex transpose. The MPI exists always and is unique.

When speaking about fast algorithms, we have in mind algorithms that have a computational complexity of less than $O(n^3)$. Actually, the algorithms presented in this paper have complexity $O(n^2)$ or less.

Before explaining our approach for the Moore–Penrose (MP) inversion of Toeplitz matrices, let us remember some facts concerning Toeplitz matrix inversion. The history starts with the famous paper of Levinson [17], in which, in principle, an algorithm for the fast solution of linear systems with a positive definite Toeplitz coefficient matrix is presented. The algorithm was further developed by other authors. A fast algorithm for the inverse matrix was first presented by Trench [21]. Other relevant sources are, for

* Received by the editors February 10, 1992; accepted for publication (in revised form) September 30, 1992.

† Universität Leipzig, FB Mathematik/Informatik, Augustusplatz, D-04109 Leipzig, Germany (georg@math-1.sci.kuniv.edu.kw or hellinge@server1.rz.uni-leipzig.de).

instance, [5] and [23]. In [6], formulas for the inverse of a Toeplitz matrix A are established, which have the form

$$A^{-1} = L_1U_1 - L_2U_2, \tag{1.3}$$

where L_1, L_2 are certain lower and U_1, U_2 are upper triangular Toeplitz matrices. The entries of these matrices are determined by the first and last columns of A^{-1}. The importance of the formula consists of the fact that it allows the solution of a system $Ax = b$ with complexity $O(n \log n)$, since multiplication of a vector by a triangular Toeplitz matrix requires $O(n \log n)$ flops if the fast Fourier transform (FFT) is utilized.

To be more precise, let us note that (1.3) with L_1, L_2, U_1, U_2 being determined by the first and last columns of A^{-1} actually holds only under some additional conditions. Formulas of the form (1.3) without additional conditions can be established with the help of the concept of a fundamental system of A (see [9], [10], [11][1]).

If A is nonsingular then a fundamental system is, by definition, a basis of the (two-dimensional) kernel of the $(n-1) \times (n+1)$ matrix $(a_{i-j})_{i=1}^{n-1, n}_{j=0}$. The definition of the fundamental system concept in the singular case is given in §7.

From our viewpoint today, the construction of fast inversion algorithms for Toeplitz matrices, and other structured matrices as well, can be divided into two stages: (i) representation of A^{-1} with the help of a certain "fundamental system" (a system of $O(n)$ parameters), and (ii) evaluation of the fundamental system.

To construct fast algorithms for MP inversion, we also followed the two-stage strategy as for the inversion in the usual sense. In our paper [8], we discussed the first stage for Hankel matrices, even for the general rectangular case. The main result of this paper is that the MPI of a Hankel matrix is a 4-Bezoutian. For Toeplitz matrices, this means that there is a formula of the form (1.3) for A^+ with the difference that four instead of two matrix products are involved. The precise formulation of the corresponding result and the idea of the proof are presented in §2.

In our situation, a fundamental system is by definition a basis of a four-dimensional space, which is the kernel of a certain matrix related to A. Of course, there are many fundamental systems, but among them there are two "canonical" ones. These are fundamental systems from which we get the MPI of A immediately. However, it is sometimes convenient to use not the pure canonical systems, but a mixture of them.

The properties of the canonical systems and some of the relations between them are discussed in §3.

The main aim of this paper is to treat the second stage of MP inversion, i.e., the evaluation of a fundamental system. In view of the well-known algorithms for Toeplitz matrix inversion, the first idea is to consider two subsequent nested submatrices of the family $A^{(k)} = [a_{i-j}]_0^{k-1}$ $(k = 1, \ldots, n)$. It developed that the recursion formulas for the canonical fundamental systems are rather complicated, but quite a few other fundamental systems exist with simpler recursion formulas. One version is presented in §4.

The recursion formulas lead to an $O(n^2)$ complexity algorithm for the computation of the MPI of the Toeplitz matrix A, as shown in §5. This algorithm is of the Levinson type, which means that it involves inner product calculations; hence it is not very convenient for parallel computing. In §6 for one case we show that the algorithm has also a Schur-type version that avoids inner product calculations and leads to an $O(n)$ complexity algorithm for an n-processor computer.

---

[1] In [11] instead of "fundamental system" the name "characteristic polynomials" is used.

In §7 another approach for the evaluation of a fundamental system is presented. This approach is based on the fact that there are generalized inverses of A, i.e., matrices B satisfying condition (1.2a), which have the form (1.3) and can be computed with usual Toeplitz matrix inversion algorithms. If B is a generalized inverse of A, then

$$A^+ = (I - P)BQ, \tag{1.4}$$

where Q is the orthogonal projection onto Im A and P is the orthogonal projection onto ker A. The projections P and Q can also be determined by usual Toeplitz algorithms, and formula (1.4) can be transformed into a form containing only triangular Toeplitz matrices. This allows us to compute the vectors occurring in the matrix representation of A^+ with $O(n \log^2 n)$ complexity at a sequential or $O(n)$ complexity at an n-processor parallel computer.

We point out that the methods presented in this paper are only two of many thinkable possibilities for constructing fast algorithms for the MP inversion of Toeplitz matrices. Various other ideas evolve from the literature on least squares problems in adaptive filtering and related topics (see, e.g., [18]–[20] and references therein). One suggestion is to look for recursion formulas for the MPI of the rectangular Toeplitz matrices $\partial^k A$ that are introduced in §7. The recursion will start then with the MP inversion of a column vector, which is a simple task. Furthermore, it is interesting to study the behavior of the MPI after one row and one column extension, as it appears in adaptive filtering. Moreover, a QR decomposition approach seems to be convenient for MP inversion. We hope to connect these ideas with our approach and to continue the discussion on MP inversion of structured matrices in a subsequent paper.

Finally, let us define some notations. The elements of $\mathbb{C}^n$ are identified with column vectors and the components are numbered from 0 to $n - 1$. If $x \in \mathbb{C}^n$, then we denote by $(x)_k$ its $(k + 1)$th coefficient. For $x = (x_k)_0^{n-1}$, we denote by $x(\lambda)$ the polynomial $x(\lambda) = x_0 + x_1\lambda + \cdots + x_{n-1}\lambda^{n-1}$.

After this paper had been completed, Zhong [22] published an article containing formulas for the MPI of square Toeplitz matrices that are more involved (but are related to) than those we used in §2 as the starting point of our considerations.

2. Matrix representation of the MPI. For the construction of matrix representations of MPI of Toeplitz matrices, we employ the familiar extension approach described in Lemma 2.1 (cf. [3]).

LEMMA 2.1. *Let C be an arbitrary matrix, and let U and V be matrices the columns of which form a basis of the kernel of C, C^*, respectively. Then the matrix*

$$\mathscr{C} = \begin{bmatrix} C & V \\ U^* & 0 \end{bmatrix} \tag{2.1}$$

is nonsingular and the inverse has the form

$$\mathscr{C}^{-1} = \begin{bmatrix} C^+ & (U^*)^+ \\ V^+ & 0 \end{bmatrix}. \tag{2.2}$$

Now we consider the special case of a square Toeplitz matrix $C = A$. To get information about the structure of A^+, we utilize the following kernel structure property.

LEMMA 2.2. *Let A be a singular $n \times n$ Toeplitz matrix, and let $\kappa = \dim \ker A$ $(=n - \operatorname{rank} A)$. Then there exists a vector $u = (u_i)_0^r$, $r = n - \kappa$, such that the columns of the matrix*

$$M = \begin{bmatrix} u_0 & & 0 \\ \vdots & u_0 & \\ u_r & \vdots & \ddots \\ & u_r & & u_0 \\ & 0 & \ddots & \vdots \\ & & & u_r \end{bmatrix} \quad \text{(2.3)}$$

(κ columns)

form a basis of $\ker A$.

Lemma 2.2 was first stated and proved in a slightly different form in [7]. Concerning the case of rectangular Toeplitz matrices, we refer to [11] (see, also, [8]).

Let us note that the vector u is uniquely determined by A, up to a constant factor, and can be computed with the help of usual Toeplitz matrix inversion algorithms. The details are discussed in the Appendix.

Let J_n denote the matrix of the reflection operator,

$$J_n = \begin{bmatrix} 0 & & & 1 \\ & & \iddots & \\ & \iddots & & \\ 1 & & & 0 \end{bmatrix} \Bigg\} n.$$

Then we have

$$J_n A J_n = A^T;$$

hence

$$A^* = J_n \bar{A} J_n, \quad \text{(2.4)}$$

where the bar denotes the matrix with conjugate complex entries. From relation (2.4) we conclude that the columns of the matrix

$$N = \begin{bmatrix} \bar{u}_r & & 0 & \\ \vdots & \bar{u}_r & & \\ \bar{u}_0 & \vdots & \ddots & \\ & \bar{u}_0 & & \bar{u}_r \\ & 0 & \ddots & \vdots \\ & & & \bar{u}_0 \end{bmatrix} \quad \text{(2.5)}$$

form a basis of $\ker A^*$.

To compute the MPI of Toeplitz matrix $\mathscr{A}$, we must form the matrix

$$\mathscr{A} = \begin{bmatrix} A & N \\ M^* & 0 \end{bmatrix} \quad \text{(2.6)}$$

and evaluate its inverse. Matrix $\mathscr{A}$ is a matrix consisting of Toeplitz blocks. Such a matrix is called a *Toeplitz mosaic matrix*. The theory of Toeplitz mosaic matrices is quite similar

to the theory of block Toeplitz matrices and generalizes this theory, since any block Toeplitz matrix can be transformed into a Toeplitz mosaic matrix by permuting rows and columns. So it is shown (see [12]) that the inverse of a Toeplitz mosaic matrix is of a certain Bezoutian type, where the corresponding Bezoutian concept generalizes a concept of Anderson–Jury (see [2], [15], [16]).

If we apply the general result on the inverse of Toeplitz mosaic matrices in our special situation and consider the left upper corner of the inverse matrix; then we will obtain a representation of A^+ as a 4-Bezoutian, where the r-Bezoutian concept is understood in the sense of Lerer–Tismenetsky (see [15], [16]). We do not present the exact definition here, because in this paper we are dealing only with square Toeplitz matrices, and in this case all can be reduced to the classical Bezoutian concept. Concerning the rectangular case we refer to [8].

It is convenient to introduce classical Bezoutians with the help of their generating functions.

The generating function of a matrix $C = [c_{ij}]_0^{m-1,n-1}$ is defined as the polynomial in two variables

$$C(\lambda, \mu) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} c_{ij}\lambda^i\mu^j.$$

DEFINITION 2.1. *An $n \times n$ matrix B is said to be a* (Toeplitz) Bezoutian *if and only if there are polynomials $a(\lambda)$, $b(\lambda)$ such that*

$$B(\lambda, \mu) = \frac{a(\lambda)b^{\#}(\mu) - b(\lambda)a^{\#}(\mu)}{1 - \lambda\mu}, \tag{2.7}$$

where

$$a^{\#}(\mu) = \mu^n a(\mu^{-1}), \qquad b^{\#}(\mu) = \mu^n b(\mu^{-1}).$$

The matrix B will be called the Bezoutian *of a and b and denoted by* Bez (a, b).

Bezoutians permit matrix representations that are very important from a computational viewpoint. We note one of these representations as follows:

$$\begin{aligned}\text{Bez}\,(a, b) = &\begin{bmatrix} a_0 & & & 0 \\ a_1 & a_0 & & \\ \vdots & \ddots & \ddots & \\ a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix}\begin{bmatrix} b_n & b_{n-1} & \cdots & b_1 \\ & b_n & \ddots & \vdots \\ & 0 & \ddots & b_{n-1} \\ & & & b_n \end{bmatrix} \\ &- \begin{bmatrix} b_0 & & & 0 \\ b_1 & b_0 & & \\ \vdots & \ddots & \ddots & \\ b_{n-1} & \cdots & b_1 & b_0 \end{bmatrix}\begin{bmatrix} a_n & a_{n-1} & \cdots & a_1 \\ & a_n & \ddots & \vdots \\ & 0 & \ddots & a_{n-1} \\ & & & a_n \end{bmatrix}.\end{aligned} \tag{2.8}$$

There are many other representations of Bez (a, b). Among them are formulas involving circulant matrices that are, as shown in [1], more effective in numerical computation.

Furthermore, there are also recursion formulas of Trench type [21], e.g., let c_{ij} denote the entries of the matrix Bez (a, b), then

$$\begin{aligned} c_{i0} &= a_i b_n - b_i a_n, \\ c_{0j} &= a_0 b_{n-j} - b_0 a_{n-j}, \\ c_{ij} &= c_{i-1,j-1} + a_i b_{n-j} - b_i a_{n-j}, \qquad (i, j = 1, \ldots, n-1). \end{aligned} \tag{2.9}$$

As a consequence of the representation of the inverse of the Toeplitz mosaic matrix $\mathscr{A}$, we get the following theorem.

THEOREM 2.1. *Let A be a singular $n \times n$ Toeplitz matrix and let $\mathscr{A}$ be defined by* (2.6). *Furthermore, let the vectors $x_+, y_+, z_+, w_+ \in \mathbb{C}^n$, $\xi_+, \eta_+, \zeta_+, \omega_+ \in \mathbb{C}^\kappa$ be defined via equations*

$$\mathscr{A}\begin{bmatrix} x_+ \\ \xi_+ \end{bmatrix} = \begin{bmatrix} e_0 \\ 0 \end{bmatrix}, \qquad \mathscr{A}\begin{bmatrix} y_+ \\ \eta_+ \end{bmatrix} = \begin{bmatrix} g_+ \\ 0 \end{bmatrix}, \tag{2.10}$$
$$\mathscr{A}\begin{bmatrix} w_+ \\ \omega_+ \end{bmatrix} = \begin{bmatrix} 0 \\ e_0 \end{bmatrix}, \qquad \mathscr{A}\begin{bmatrix} z_+ \\ \zeta_+ \end{bmatrix} = \begin{bmatrix} h_+ \\ 0 \end{bmatrix},$$

where

$$e_0 = [1\ 0\ \cdots\ 0]^T, \qquad g_+ = [a_{-n}\ a_{1-n}\ \cdots\ a_{-1}]^T, \qquad h_+ = [0\ \cdots\ 0\ \bar{u}_r\ \cdots\ \bar{u}_1]^T,$$

0 *is a zero vector of suitable length, and a_{-n} is an arbitrary fixed number.*

Then

$$A^+ = \mathrm{Bez}\,(\tilde{y}_+, x_+) + \mathrm{Bez}\,(z_+, w_+), \tag{2.11}$$

where

$$\tilde{y}_+(\lambda) = y_+(\lambda) - \lambda^n.$$

The proof of the theorem is completely analogous to the corresponding Hankel matrix theorem that was stated and proved for the more general rectangular case in [8]. Therefore, we do not present it here.

The recursion formula (2.9) allows us to compute A^+ with an amount of $O(n^2)$ flops, provided that the vectors x_+, y_+, z_+, w_+ are known. Moreover, with the help of formula (2.8), any pseudo solution A^+y of a system $Ax = y$ can be computed with complexity $O(n \log n)$ if the FFT is utilized.

Vectors g_+ and h_+ appear as prolongation of the blocks in the matrix $\mathscr{A}$ to the right. If we take the analogous prolongation to the left, we get the formula described in the following theorem.

THEOREM 2.2. *Let A be a singular $n \times n$ Toeplitz matrix and let $\mathscr{A}$ be defined by* (2.6). *Furthermore, let vectors $x_-, y_-, z_-, w_- \in \mathbb{C}^n$, $\xi_-, \eta_-, \zeta_-, \omega_- \in \mathbb{C}^\kappa$ be defined via equations*

$$\mathscr{A}\begin{bmatrix} x_- \\ \xi_- \end{bmatrix} = \begin{bmatrix} e_{n-1} \\ 0 \end{bmatrix}, \qquad \mathscr{A}\begin{bmatrix} y_- \\ \eta_- \end{bmatrix} = \begin{bmatrix} g_- \\ 0 \end{bmatrix}, \tag{2.12}$$
$$\mathscr{A}\begin{bmatrix} w_- \\ \omega_- \end{bmatrix} = \begin{bmatrix} 0 \\ e_{\kappa-1} \end{bmatrix}, \qquad \mathscr{A}\begin{bmatrix} z_- \\ \zeta_- \end{bmatrix} = \begin{bmatrix} h_- \\ 0 \end{bmatrix},$$

where $e_{n-1}(e_{\kappa-1})$ is the last unit vector in $\mathbb{C}^n$ ($\mathbb{C}^\kappa$), $g_- = [a_1\ a_2\ \cdots\ a_n]^T$, $h_- = [\bar{u}_{r-1}\ \cdots\ \bar{u}_0\ 0\ \cdots\ 0]^T$, 0 is a zero vector of suitable length, and a_n is an arbitrary fixed number.

Then

$$A^+ = \mathrm{Bez}\,(\tilde{y}_-, \tilde{x}_-) + \mathrm{Bez}\,(\tilde{z}_-, \tilde{w}_-), \tag{2.13}$$

where $\tilde{x}_-(\lambda) = \lambda x_-(\lambda)$, $\tilde{y}_-(\lambda) = -1 + \lambda y_-(\lambda)$, $\tilde{w}_-(\lambda) = \lambda w_-(\lambda)$, and $\tilde{z}_-(\lambda) = \lambda z_-(\lambda)$.

3. Fundamental systems. In the previous section, we presented two formulas for the MPI of a Toeplitz matrix A. In this section we first show that there is actually a

variety of such formulas. This variety is connected with a four-dimensional space, and any choice of a basis corresponds with a representation of A^+. Later, it will be clear that a convenient choice of the basis has great influence on the complexity of the formulas and, therefore, of the algorithm.

Let us start with some notation. If $C = [c_{i-j}]_{i=0,j=0}^{p-1,q-1}$ is a $p \times q$ Toeplitz matrix, then ∂C will denote the $(p-1) \times (q+1)$ Toeplitz matrix

$$\partial C := [c_{i-j}]_{i=1,j=0}^{p-1,q}.$$

For $\mathscr{A}$ given by (2.6), we define

$$\partial\mathscr{A} = \begin{bmatrix} \partial A & \partial N \\ \partial M^* & 0 \end{bmatrix}. \tag{3.1}$$

Since $\mathscr{A}$ is nonsingular, $\partial\mathscr{A}$ has a kernel of dimension four.

DEFINITION 3.1. *Any basis of* $\ker \partial\mathscr{A}$ *is called a fundamental system of* $\mathscr{A}$.

We can observe immediately that the following two systems are fundamental:

$$X_+ = \begin{bmatrix} x_+ \\ 0 \\ \hline \xi_+ \\ 0 \end{bmatrix}, \quad Y_+ = \begin{bmatrix} y_+ \\ -1 \\ \hline \eta_+ \\ 0 \end{bmatrix}, \quad W_+ = \begin{bmatrix} w_+ \\ 0 \\ \hline \omega_+ \\ 0 \end{bmatrix}, \quad Z_+ = \begin{bmatrix} z_+ \\ 0 \\ \hline \zeta_+ \\ -1 \end{bmatrix} \tag{3.2}$$

and

$$X_- = \begin{bmatrix} 0 \\ x_- \\ \hline 0 \\ \xi_- \end{bmatrix}, \quad Y_- = \begin{bmatrix} -1 \\ y_- \\ \hline 0 \\ \eta_- \end{bmatrix}, \quad W_- = \begin{bmatrix} 0 \\ w_- \\ \hline 0 \\ \omega_- \end{bmatrix}, \quad Z_- = \begin{bmatrix} 0 \\ z_- \\ \hline -1 \\ \zeta_- \end{bmatrix}. \tag{3.3}$$

The quantities appearing here are defined by (2.10) and (2.12).

Fundamental systems (3.2) and (3.3) will be referred to as *canonical*; system (3.2) will be called the (+)-*system* and (3.3) will be called the (−)-*system*. Let us point out that the canonical fundamental systems depend on fixed numbers a_n and a_{-n}.

Any fundamental system can be transformed into canonical systems. To show how to do this, we introduce matrices $\Phi_\pm$:

$$\Phi_+ = \left[\begin{array}{ccccccc|ccccc} a_0 & a_{-1} & & & \cdots & a_{-n} & & \bar{u}_r & 0 & \cdots & & 0 \\ 0 & & \cdots & & 0 & -1 & & 0 & & \cdots & & 0 \\ \bar{u}_0 & \cdots & \bar{u}_r & 0 & \cdots & 0 & & 0 & & \cdots & & 0 \\ 0 & 0 & & & \cdots & 0 & & 0 & & \cdots & 0 & -1 \end{array}\right],$$

$$\Phi_- = \left[\begin{array}{ccccccc|ccccc} a_n & a_{n-1} & & & \cdots & a_0 & & 0 & & \cdots & 0 & \bar{u}_0 \\ -1 & 0 & & & \cdots & 0 & & 0 & & \cdots & & 0 \\ 0 & \cdots & 0 & \bar{u}_0 & \cdots & \bar{u}_r & & 0 & & \cdots & & 0 \\ 0 & 0 & & & \cdots & 0 & & -1 & 0 & \cdots & & 0 \end{array}\right].$$

It can be easily seen that

$$\Phi_+[X_+Y_+W_+Z_+] = \Phi_-[X_-Y_-W_-Z_-] = I_4. \tag{3.4}$$

From this relation, we get the following proposition.

PROPOSITION 3.1. *Let* X_1, X_2, X_3, X_4 *be a fundamental system of* $\mathscr{A}$ *and let* X *be the matrix with the columns* X_i $(i = 1, \ldots, 4)$. *Then* $\Lambda_\pm := \Phi_\pm X$ *is nonsingular and the columns of* $X\Lambda_\pm^{-1}$ *form the* (±)-*canonical fundamental system of* $\mathscr{A}$.

Our next aim is to get more information about the vectors of the canonical fundamental systems. For this we need the MPI of the matrices M^* and N occurring in the definition of $\mathscr{A}$. Since M and N have full rank, we have

$$(M^*)^+ = M(M^*M)^{-1}, \qquad N^+ = (N^*N)^{-1}N^*. \tag{3.5}$$

We introduce the matrix

$$T = M^*M. \tag{3.6}$$

Obviously, T is a positive definite $\kappa \times \kappa$ Toeplitz matrix. Furthermore, we have

$$N = J_n \bar{M} J_\kappa. \tag{3.7}$$

From (3.7), we obtain

$$N^*N = J_\kappa \bar{M}^* \bar{M} J_\kappa = J_\kappa \bar{T} J_\kappa = T.$$

Hence the following lemma is true.

LEMMA 3.1. *It holds that*

$$(M^*)^+ = MT^{-1}, \qquad N^+ = T^{-1}N^*. \tag{3.8}$$

Now we can start our investigation of the vectors of the canonical fundamental systems. We use all notations from above. Furthermore, we define

$$c = T^{-1}e_0. \tag{3.9}$$

In view of (2.4), we have

$$T^{-1}e_{\kappa-1} = J_\kappa \bar{c} =: \hat{c}. \tag{3.10}$$

First we consider vectors $W_\pm$ that have the simplest structure.

PROPOSITION 3.2. *It holds that*

(a) $\omega_+ = \omega_- = 0$,

(b) $w_+ = Mc, \quad w_- = M\hat{c}$.

The assertion follows immediately from Lemma 2.1 and (3.8).

In the sequel, the numbers $\sigma_\pm$ defined by

$$\sigma_+ = [a_{-1} \quad \cdots \quad a_{-n}]u_+, \qquad \sigma_- = [a_n \quad \cdots \quad a_1]u_-, \tag{3.11}$$

where

$$u_+ = [0 \quad \cdots \quad 0 \quad u_0 \quad \cdots \quad u_r]^T, \qquad u_- = [u_0 \quad \cdots \quad u_r \quad 0 \quad \cdots \quad 0]^T$$

will play an important role. They appear in the following lemma, which is easily verified.

LEMMA 3.2. *Let $u'_\pm \in \mathbb{C}^n$ be defined by*

$$u'_+ = [0 \quad \cdots \quad 0 \quad u_0 \quad \cdots \quad u_{r-1}]^T, \qquad u'_- = [u_1 \quad \cdots \quad u_r \quad 0 \quad \cdots \quad 0]^T.$$

Then

$$Au'_+ = \sigma_+ e_0 - u_r g_+ \quad \text{and} \quad Au'_- = \sigma_- e_{n-1} - u_0 g_-. \tag{3.12}$$

From the relation (3.12), we obtain the following fact that will be important in the sequel.

COROLLARY 3.1. *At least one of the numbers, σ_+ and u_r, and one of the numbers, σ_- and u_0, are different from zero.*

In fact, $\sigma_+ = u_r = 0$ would imply that $u'_+ \in \ker A$, i.e., $u'_+ \in \operatorname{Im} M$. But u'_+ is linearly independent of the columns of M. An analogous argument provides $\sigma_- \neq 0$ or $u_0 \neq 0$.

Let M_k denote the matrix of the form (2.3) with k columns. Furthermore, we define

$$T_k := M_k^* M_k = [t_{i-j}]_0^{k-1} \quad \text{and} \quad c_k = T_k^{-1} e_0, \qquad \hat{c}_k = T_k^{-1} e_{k-1}.$$

In particular, $M_\kappa = M$, $T_\kappa = T$, $c_\kappa = c$. Let us note that T_j is a principal submatrix of T_k if $j \le k$. We denote by d and $\hat{d}$ the solutions of equations

$$Td = (t_{-\kappa+k})_{k=0}^{\kappa-1}, \qquad T\hat{d} = (t_{1+k})_{k=0}^{\kappa-1}. \tag{3.13}$$

In view of $t_{-k} = \bar{t}_k$, we have $\hat{d} = \overline{J_\kappa d}$. Furthermore, d is connected with $c_{\kappa+1}$ via

$$\begin{bmatrix} d \\ -1 \end{bmatrix} = -\frac{1}{\varepsilon_{\kappa+1}} \hat{c}_{\kappa+1}, \qquad \begin{bmatrix} -1 \\ \hat{d} \end{bmatrix} = -\frac{1}{\varepsilon_{\kappa+1}} c_{\kappa+1}, \tag{3.14}$$

where

$$\varepsilon_{\kappa+1} = (c_{\kappa+1})_0 = (c_{\kappa+1})_\kappa. \tag{3.15}$$

In the next two propositions we describe the structure of vectors X_+, X_-, Y_+, Y_- of the canonical fundamental systems.

PROPOSITION 3.3. *It holds that*

(a) $\xi_+ = u_r c$, $\xi_- = u_0 \hat{c}$.

(b) *If* $u_r = 0$, *then* $\sigma_+ \neq 0$ *and*

$$\begin{bmatrix} x_+ \\ 0 \end{bmatrix} = -\frac{1}{\sigma_+} M_{\kappa+1} \begin{bmatrix} d \\ -1 \end{bmatrix}. \tag{3.16a}$$

(c) *If* $u_0 = 0$, *then* $\sigma_- \neq 0$, *and*

$$\begin{bmatrix} 0 \\ x_- \end{bmatrix} = -\frac{1}{\sigma_-} M_{\kappa+1} \begin{bmatrix} -1 \\ \hat{d} \end{bmatrix}. \tag{3.16b}$$

Proof. Two relations (a) follow from Lemma 2.1 and Lemma 3.1.

Now let $u_r = 0$. According to Lemma 3.2, we have $Au'_+ = \sigma_+ e_0$, where $\sigma_+ \neq 0$ holds due to Corollary 3.1. Since $x_+ = A^+ e_0$, we conclude that

$$x_+ = \frac{1}{\sigma_+} u'_+ - M\alpha, \tag{3.17}$$

where $\alpha \in \mathbb{C}^\kappa$ must be chosen in such a way that x_+ is orthogonal to $\ker A$, which means that $M^* x_+ = 0$. Hence $T\alpha = M^* M\alpha = \frac{1}{\sigma_+} M^* u'_+$. Since $M^* u'_+ = (t_{-\kappa+k})_0^{\kappa-1}$, we conclude that $\alpha = \frac{1}{\sigma_+} d$ and $x_+ = -\frac{1}{\sigma_+}(Md - u'_+)$. This is equivalent to (3.16a), and so (b) is proved.

Assertion (c) is proved analogously. □

PROPOSITION 3.4. *It holds that*

(a) $\eta_+ = \sigma_+ c$, $\eta_- = \sigma_- \hat{c}$.

(b) *If* $\sigma_+ = 0$, *then* $u_r \neq 0$ *and*

$$\begin{bmatrix} y_+ \\ -1 \end{bmatrix} = \frac{1}{u_r} M_{\kappa+1} \begin{bmatrix} d \\ -1 \end{bmatrix}. \tag{3.18}$$

(c) *If* $\sigma_- = 0$, *then* $u_0 \neq 0$, *and*

$$\begin{bmatrix} -1 \\ y_- \end{bmatrix} = \frac{1}{u_0} M_{\kappa+1} \begin{bmatrix} -1 \\ \hat{d} \end{bmatrix}. \tag{3.19}$$

The proof is completely analogous to the one of Proposition 3.3.

Finally we state a proposition concerning the vectors $Z_{\pm}$.

PROPOSITION 3.5. *It holds that* $\zeta_+ = d$, $\zeta_- = \hat{d}$.

Proof. By Lemmas 2.1 and 3.1 we have

$$\zeta_+ = N^+h_+ = T^{-1}N^*h_+ = T^{-1}(t_{-\kappa+k})_{k=0}^{\kappa-1} = d$$

and

$$\zeta_- = N^+h_- = T^{-1}N^*h_- = T^{-1}(t_{1+k})_{k=0}^{\kappa-1} = \hat{d}. \qquad \square$$

Now we are able to indicate a collection of fundamental systems different from the canonical ones.

THEOREM 3.1. *The following systems are fundamental.*

System (A). *Case* (I) $u_0u_r \neq 0$, X_+, X_-, W_+, W_-.
Case (II) $u_0\sigma_+ \neq 0$, Y_+, X_-, W_+, W_-.
Case (III) $u_r\sigma_- \neq 0$, X_+, Y_-, W_+, W_-.
Case (IV) $\sigma_+\sigma_- \neq 0$, Y_+, Y_-, W_+, W_-.
System (B). W_+, W_-, Z_+, Z_-.
System (C). X_+, X_-, Y_+, Y_-.

Proof. It suffices to show the linear independence of the system. The linear independence of Systems (A) and (B) follows from Propositions 3.2–3.5 and the fact that the vectors

$$\begin{bmatrix} c \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \hat{c} \end{bmatrix}$$

are linearly independent since $(c)_0 \neq 0$ and the vectors

$$\begin{bmatrix} d \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ \hat{d} \end{bmatrix}$$

are linearly independent in view of (3.14).

It remains to verify that System (C) is linearly independent. Suppose that for $\alpha_{\pm}$, $\beta_{\pm} \in \mathbb{C}$,

$$\alpha_+X_+ + \alpha_-X_- + \beta_+Y_+ + \beta_-Y_- = 0.$$

By Propositions 3.3 and 3.4, we obtain for the second parts of the considered vectors

$$a_+u_r\begin{bmatrix} c \\ 0 \end{bmatrix} + \alpha_-u_0\begin{bmatrix} 0 \\ \hat{c} \end{bmatrix} + \beta_+\sigma_+\begin{bmatrix} c \\ 0 \end{bmatrix} + \beta_-\sigma_-\begin{bmatrix} 0 \\ \hat{c} \end{bmatrix} = 0.$$

Since the vectors

$$\begin{bmatrix} c \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \hat{c} \end{bmatrix}$$

are linearly independent, we conclude that

$$\alpha_+u_r + \beta_+\sigma_+ = \alpha_-u_0 + \beta_-\sigma_- = 0.$$

Hence the second parts of the vectors

$$V_+ := \alpha_+X_+ + \beta_+Y_+ \quad \text{and} \quad V_- := \alpha_-X_- + \beta_-Y_-$$

vanish and their first parts depend linearly. Repeating the arguments of the proof of Proposition 3.3, we obtain that the first parts of $V_\pm$ are multiples of the vectors

$$M_{\kappa+1}\begin{bmatrix} d \\ -1 \end{bmatrix} \quad \text{and} \quad M_{\kappa+1}\begin{bmatrix} -1 \\ \hat{d} \end{bmatrix},$$

respectively. But the latter two vectors are linearly independent. Consequently, $V_+ = V_- = 0$. In view of the linear independence of X_+, Y_+ and X_-, Y_-, we may conclude that $\alpha_\pm = \beta_\pm = 0$, which proves the linear independence of the system. □

In view of Corollary 3.1, all possible situations are covered by Cases (I)–(IV) of System (A) of Theorem 3.1.

In the next two sections, we establish a recursive algorithm that is based on System (A) of Theorem 3.1. In this connection, we must show how to jump from one of the cases (I)–(IV) to another one. It follows from the algebraic theory of Toeplitz matrices that only the following changes are possible:

(3.20) $$\text{(I)} \to \text{(II–IV)}, \quad \text{(II)} \to \text{(IV)}, \quad \text{(III)} \to \text{(IV)}.$$

For the jumps indicated in (3.20) we need the following relation.

PROPOSITION 3.6. *It holds that*

(a) *if* $u_r \neq 0$, *then*

$$Y_+ = \frac{\sigma_+}{u_r} X_+ - \frac{1}{\varepsilon u_r}(W_- - \bar{\alpha} W_+), \tag{3.21}$$

where

$$\varepsilon = (c)_0, \tag{3.22a}$$

$$\alpha = [t_\kappa \quad \cdots \quad t_1]c. \tag{3.22b}$$

(b) *If* $u_0 \neq 0$, *then*

$$Y_- = \frac{\sigma_-}{u_0} X_- - \frac{1}{\varepsilon u_0}(W_+ - \alpha W_-), \tag{3.23}$$

where ε *and* α *are defined as above.*

Proof. We prove only (a). The proof of (b) is analogous.

Suppose that $u_r \neq 0$. In view of Corollary 3.1, we may assume that $u_0 \neq 0$ or $\sigma_- \neq 0$.

Since $Y_+ \in \ker \partial\mathscr{A}$, Y_+ is by Theorem 3.1 a linear combination of the vectors X_+, X_-, W_+, W_- if $u_0 \neq 0$ and a linear combination of X_+, Y_-, W_+, W_- if $\sigma_- \neq 0$. We recall that according to Propositions 3.2, 3.3, and 3.4, the second parts of the vectors $W_\pm$ vanish and the second parts of X_+, X_-, Y_+, Y_- are given by

$$\begin{bmatrix} u_r c \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ u_0 \hat{c} \end{bmatrix}, \begin{bmatrix} \sigma_+ c \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \sigma_- \hat{c} \end{bmatrix},$$

respectively. Since the last component of $\hat{c}$ is nonzero, we conclude that Y_+ is already a linear combination of X_+, W_+, and W_- in both cases $u_0 \neq 0$ and $\sigma_- \neq 0$. From this and (3.22), we conclude that

$$Y_+ = \frac{\sigma_+}{u_r} X_+ + \mu_+ W_+ + \mu_- W_- \tag{3.24}$$

for certain $\mu_\pm \in \mathbb{C}$.

To evaluate $\mu_{\pm}$, we apply the functionals

$$f_1^T = [0 \ \cdots \ 0 \ -1 \,|\, 0 \ \cdots \ 0], \qquad f_2^T = [\bar{u}_0 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0 \,|\, 0 \ \cdots \ 0],$$

which are the second and third rows of the matrix Φ_+, to (3.24). As the result, we get

$$1 = -\mu_-(w_-)_{n-1} \quad \text{and} \quad 0 = \mu_+ + \bar{\alpha}\mu_-,$$

where

$$\bar{\alpha} = [\bar{u}_1 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0]w_-.$$

This can be transformed into

$$\mu_- = -\frac{1}{(w_-)_{n-1}} \quad \text{and} \quad \mu_+ = -\frac{\bar{\alpha}}{(w_-)_{n-1}}. \tag{3.25}$$

From Proposition 3.2, we get $(w_-)_{n-1} = u_r(\hat{c})_{\kappa-1} = u_r c_0$ and

$$\bar{\alpha} = [\bar{u}_1 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0]M\hat{c} = [t_{-1} \ \cdots \ t_{-\kappa}]\hat{c}.$$

Inserting this and (3.25) into (3.24), we obtain (3.21), (3.22). □

Remark. We have proved that ε and α are also given by

$$\begin{aligned} \varepsilon &= (\hat{c})_{\kappa-1}, \\ \bar{\alpha} &= [\bar{u}_1 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0]w_-, \\ \alpha &= [0 \ \cdots \ 0 \ \bar{u}_0 \ \cdots \ \bar{u}_{r-1}]w_+. \end{aligned} \tag{3.26}$$

COROLLARY 3.2. *It holds that*

(a) *if* $u_r \neq 0$, *then*

$$\begin{bmatrix} y_+ \\ -1 \end{bmatrix} = \frac{\sigma_+}{u_r}\begin{bmatrix} x_+ \\ 0 \end{bmatrix} - \frac{1}{\varepsilon u_r}\left(\begin{bmatrix} 0 \\ w_- \end{bmatrix} - \bar{\alpha}\begin{bmatrix} w_+ \\ 0 \end{bmatrix}\right). \tag{3.27}$$

(b) *If* $u_0 \neq 0$, *then*

$$\begin{bmatrix} -1 \\ y_- \end{bmatrix} = \frac{\sigma_-}{u_0}\begin{bmatrix} 0 \\ x_- \end{bmatrix} - \frac{1}{\varepsilon u_0}\left(\begin{bmatrix} w_+ \\ 0 \end{bmatrix} - \alpha\begin{bmatrix} 0 \\ w_- \end{bmatrix}\right), \tag{3.28}$$

where α *and* ε *are given by* (3.22) *or* (3.26).

We discuss some specifications for the case of a Hermitian or positive semidefinite Toeplitz matrix A.

Suppose that $A^* = A$, i.e., $a_{-i} = \bar{a}_i$ $(i = 0, \ldots, n-1)$. In this case the vector $u = (u_i)_0^r$ appearing in Lemma 2.2 can be chosen in such a way that $\hat{u} = u$, i.e., $u_i = \bar{u}_{r-i}$ $(i = 0, \ldots, r)$. Hence matrix $\mathscr{A}$ is Hermitian.

In view of the relation

$$\begin{bmatrix} J_n & 0 \\ 0 & J_n \end{bmatrix}\mathscr{A}\begin{bmatrix} J_n & 0 \\ 0 & J_n \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{N} \\ \bar{M}^* & 0 \end{bmatrix},$$

we obtain the following relations.

Remark 3.1. If A is Hermitian, then $x_- = \hat{x}_+$, $y_- = \hat{y}_+$, $w_- = \hat{w}_+$, $z_- = \hat{z}_+$. Furthermore, $\sigma_- = \bar{\sigma}_+$. The latter relation implies the following remark.

Remark 3.2. If A is Hermitian, then only Cases (I) and (IV) of Theorem 3.1 appear. For semidefinite matrices still more is true.

Remark 3.3. If A is positive or negative semidefinite, then only Case (I) of Theorem 3.1 appears.

Remark 3.3 is a consequence of Iohvidov's rule for the signature of a Toeplitz matrix (see [13] and [11]).

4. Recursions of fundamental systems. In this section we consider, besides the Toeplitz matrix $A = [a_{i-j}]_0^{n-1}$, an extension A' of A of the form

$$A' = [a_{i-j}]_0^n.$$

The problem we discuss is how a fundamental system for the Toeplitz mosaic matrix $\mathscr{A}'$ associated with A' can be obtained from a fundamental system of $\mathscr{A}$. The recursion will then lead to a fast algorithm for computing a fundamental system of an arbitrary square Toeplitz matrix.

Since there are many fundamental systems, we must decide which system provides the simplest recursion formulas. The first idea is to take one of the canonical fundamental systems. However, the corresponding formulas for this approach seem to be rather complicated. All Systems (A)–(C) described in Theorem 3.1 lead to simpler formulas. So far, we do not really know which version is the simplest. From a certain viewpoint, System (A) seems to be the most natural, therefore we describe it in detail.

Let us agree upon some notational convention. All quantities corresponding to matrix A' will be denoted by the same symbols as the analogous quantities corresponding to A, only supplemented by a prime.

The first question we must answer is that of the structure of matrix $\mathscr{A}'$. For this, we must determine the kernel of A' in dependence on ker A. The following lemma is well known (see [11]).

LEMMA 4.1. *Let the numbers $\sigma_\pm$ be defined by* (3.11). *Then*

(I) $$\kappa' = \kappa + 1, \quad u' = u, \quad M' = M_{\kappa+1} \quad \text{if } \sigma_+ = \sigma_- = 0;$$

(II) $$\kappa' = \kappa, \quad u' = \begin{bmatrix} u \\ 0 \end{bmatrix}, \quad M' = \begin{bmatrix} M_\kappa \\ 0 \end{bmatrix} \quad \text{if } \sigma_+ \neq \sigma_- = 0;$$

(III) $$\kappa' = \kappa, \quad u' = \begin{bmatrix} 0 \\ u \end{bmatrix}, \quad M' = \begin{bmatrix} 0 \\ M_\kappa \end{bmatrix} \quad \text{if } \sigma_+ = \sigma_- \neq 0;$$

(IV) $$\kappa' = \kappa - 1, \quad u' = \begin{bmatrix} 0 \\ u \\ 0 \end{bmatrix}, \quad M' = \begin{bmatrix} 0 \\ M_{\kappa-1} \\ 0 \end{bmatrix} \quad \text{if } \sigma_+\sigma_- \neq 0.$$

In all cases $N' = J_{n+1}\bar{M}'J_{\kappa'}$.

The simplest recursion formulas are those for vectors $w_\pm$. They are based on the following lemma.

LEMMA 4.2. *For the first and last columns, c_κ and $\hat{c}_\kappa$ of T_κ^{-1} ($T_\kappa = M_\kappa^* M_\kappa$), the following recursions hold true*:

(4.1) (i) $$[c_{\kappa+1}(\lambda) \quad \hat{c}_{\kappa+1}(\lambda)] = \frac{1}{1 - |\alpha^2|}[c(\lambda) \quad \lambda\hat{c}(\lambda)]\begin{bmatrix} 1 & -\bar{\alpha} \\ -\alpha & 1 \end{bmatrix},$$

where

(4.2) $$\alpha = [t_\kappa \quad \cdots \quad t_1]c \quad (\bar{\alpha} = [t_{-1} \quad \cdots \quad t_{-\kappa}]\hat{c}),$$

and

(4.3) (ii) $$[c_{\kappa-1}(\lambda) \quad \lambda\hat{c}_{\kappa-1}(\lambda)] = [c(\lambda) \quad \hat{c}(\lambda)]\begin{bmatrix} 1 & \bar{\delta} \\ \delta & 1 \end{bmatrix},$$

where

$$\delta = -\frac{(c)_{\kappa-1}}{(c)_0}. \tag{4.4}$$

Formulas (4.1) and (4.2) are well known and easily checked. Formula (4.1) provides only the classical Levinson algorithm.

Remark 4.1. For $c_{\kappa+1} := T_{\kappa+1}^{-1} e_0$, the relation $\alpha_\kappa = (c_{\kappa+1})_\kappa/(c_{\kappa+1})_0$ holds.

THEOREM 4.1. *Let* $\sigma_\pm$ *be defined by* (3.11). *The vectors* $w'_\pm$ *are obtained from* w_+ *and* w_- *via the following formulas*:

(I) Case $\sigma_+ = \sigma_- = 0$:

$$[w'_+(\lambda) \quad w'_-(\lambda)] = \frac{1}{1-|\alpha^2|}[w_+(\lambda) \quad w_-(\lambda)]\begin{bmatrix}1 & 0\\ 0 & \lambda\end{bmatrix}\begin{bmatrix}1 & -\bar{\alpha}\\ -\alpha & 1\end{bmatrix}, \tag{4.5}$$

where

$$\alpha = [t_\kappa \quad \cdots \quad t_1]c = [0 \quad \cdots \quad 0 \quad \bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]w_+ \tag{4.6a}$$

and

$$\bar{\alpha} = [t_{-1} \quad \cdots \quad t_{-\kappa}]\hat{c} = [\bar{u}_1 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0]w_-. \tag{4.6b}$$

(II) Case $\sigma_+ \neq 0$, $\sigma_- = 0$:

$$[w'_+(\lambda) \quad w'_-(\lambda)] = [w_+(\lambda) \quad w_-(\lambda)]. \tag{4.7}$$

(III) Case $\sigma_+ = 0$, $\sigma_- \neq 0$:

$$[w'_+(\lambda) \quad w'_-(\lambda)] = [\lambda w_+(\lambda) \quad \lambda w_-(\lambda)]. \tag{4.8}$$

(IV) Case $\sigma_+\sigma_- \neq 0$:

$$[w'_+(\lambda) \quad w'_-(\lambda)] = [w_+(\lambda) \quad w_-(\lambda)]\begin{bmatrix}1 & \bar{\delta}\\ \delta & 1\end{bmatrix}\begin{bmatrix}\lambda & 0\\ 0 & 1\end{bmatrix}, \tag{4.9}$$

where

$$\delta = -\frac{(c)_{\kappa-1}}{(c)_0} = -\frac{(w_+)_t}{(w_-)_t} \tag{4.10a}$$

and

$$\bar{\delta} = -\frac{(\hat{c})_0}{(c)_0} = -\frac{(w_-)_s}{(w_+)_s}, \tag{4.10b}$$

where $(w_\pm)_s$ *is the first and* $(w_\pm)_t$ *the last nonzero component of* $w_\pm$.

Proof. (I). By Proposition 3.2 and Lemma 4.1, we have

$$w'_+ = M_{\kappa+1}c_{\kappa+1}, \qquad w' = M_{\kappa+1}\hat{c}_{\kappa+1}.$$

Taking (4.1) into account, we obtain (4.5) with α being defined by the first equalities of (4.6a) or (4.6b).

To get the second expressions for constant α, we remark that

$$\mathscr{A}'\begin{bmatrix}w_+\\ \hline 0\\ 0\end{bmatrix} = \begin{bmatrix}0\\ \hline e_0 + \alpha_1 e_\kappa\end{bmatrix} \quad \text{and} \quad \mathscr{A}'\begin{bmatrix}0\\ \hline w_-\\ 0\end{bmatrix} = \begin{bmatrix}0\\ \hline \alpha_2 e_0 + e_\kappa\end{bmatrix},$$

where

$$\alpha_1 = [0 \quad \cdots \quad 0 \quad \bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]w_+, \qquad \alpha_2 = [\bar{u}_1 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0]w_-.$$

From this relation, we conclude that

$$w'_+(\lambda) = \frac{1}{1 - \alpha_1\alpha_2}(w_+(\lambda) - \alpha_1\lambda w(\lambda))$$

and

$$w'_-(\lambda) = \frac{1}{1 - \alpha_1\alpha_2}(\lambda w_-(\lambda) - \alpha_2 w_+(\lambda)).$$

Comparing this with (4.5), we obtain $\alpha_1 = \alpha$ and $\alpha_2 = \bar{\alpha}$.

(II). By Lemma 4.1, we have in this case

$$M' = \begin{bmatrix} M \\ 0 \end{bmatrix}.$$

Hence $T' = (M')^*M' = M^*M = T$, which implies that $c' = c$. Thus by Proposition 3.2

$$w'_+ = M'c' = \begin{bmatrix} w_+ \\ 0 \end{bmatrix}, \qquad w'_- = M'\hat{c}' = \begin{bmatrix} w_- \\ 0 \end{bmatrix},$$

which is the same as (4.7).

(III). By Lemma 4.1, we have

$$M' = \begin{bmatrix} 0 \\ M \end{bmatrix},$$

which implies that $T' = (M')^*M' = M^*M = T$ and $c' = c$. Hence by Proposition 3.2

$$w'_+ = M'c' = \begin{bmatrix} 0 \\ w_+ \end{bmatrix}, \qquad w'_- = M'\hat{c}' = \begin{bmatrix} 0 \\ w_- \end{bmatrix},$$

which is equivalent to (4.8).

(IV). Again, applying Lemma 4.1, we get

$$T' = (M')^*M' = M^*_{\kappa-1}M_{\kappa-1} = T_{\kappa-1} \quad \text{and} \quad c' = c_{\kappa-1}.$$

According to Proposition 3.2, we have

$$w'_+ = \begin{bmatrix} 0 \\ M_{\kappa-1}c' \\ 0 \end{bmatrix}, \qquad w'_- = \begin{bmatrix} 0 \\ M_{\kappa-1}\hat{c}' \\ 0 \end{bmatrix}, \tag{4.11}$$

and by Lemma 4.2, we have

$$\begin{bmatrix} M_{\kappa-1}c' \\ 0 \end{bmatrix} = M_\kappa \begin{bmatrix} c' \\ 0 \end{bmatrix} = M_\kappa(c - \delta\hat{c}) = w_+ - \delta w_- \tag{4.12a}$$

and

$$\begin{bmatrix} 0 \\ M_{\kappa-1}c' \end{bmatrix} = M_\kappa \begin{bmatrix} 0 \\ \hat{c}' \end{bmatrix} = M_\kappa(\hat{c} - \bar{\delta}c) = w_- - \bar{\delta}w_+. \tag{4.12b}$$

Taking (4.11) and (4.12) together, we get (4.9) with δ being defined by the first equalities of (4.10). The second expressions follow from $w_+ = Mc$, $w_- = M\hat{c}$. □

Remark 4.2. The number δ appearing in Case (IV) can also be expressed in the form

$$\delta = -\frac{f_+^T w_+}{f_+^T w_-} \quad \text{and} \quad \bar{\delta} = -\frac{f_-^T w_-}{f_-^T w_+}, \tag{4.13}$$

where

$$f_+^T = [a_{-1} \quad \cdots \quad a_{-n}] \quad \text{and} \quad f_-^T = [a_n \quad \cdots \quad a_1].$$

This follows from the relations

$$\mathscr{A}'\begin{bmatrix} 0 \\ w_+ \\ \hline 0 \end{bmatrix} = \begin{bmatrix} f_+^T w_+ e_0 \\ \hline e_0 \end{bmatrix}, \qquad \mathscr{A}'\begin{bmatrix} 0 \\ w_- \\ \hline 0 \end{bmatrix} = \begin{bmatrix} f_+^T w_- e_0 \\ \hline 0 \end{bmatrix},$$

$$\mathscr{A}'\begin{bmatrix} w_+ \\ 0 \\ \hline 0 \end{bmatrix} = \begin{bmatrix} f_-^T w_+ e_n \\ \hline 0 \end{bmatrix}, \quad \text{and} \quad \mathscr{A}'\begin{bmatrix} w_- \\ 0 \\ \hline 0 \end{bmatrix} = \begin{bmatrix} f_-^T w_- e_0 \\ \hline e_{\kappa-2} \end{bmatrix}.$$

Now we are going to construct recursion formulas for vectors $x_\pm$, $y_\pm$ according to System (A) of Theorem 3.1. We begin with Case (I).

THEOREM 4.2. *Suppose that* $\sigma_+ = \sigma_- = 0$. *Then* $u_0 u_r \neq 0$ *and*

$$[x'_+(\lambda) \quad x'_-(\lambda)] = \frac{1}{1 - \alpha_+\alpha_-}[x_+^0(\lambda) \quad x_-^0(\lambda)]\begin{bmatrix} 1 & -\alpha_- \\ -\alpha_+ & 1 \end{bmatrix}, \tag{4.14}$$

where

$$x_+^0(\lambda) = x_+(\lambda) - \beta_+ w'_-(\lambda), \qquad x_-^0(\lambda) = \lambda x_-(\lambda) - \beta_- w'_+(\lambda), \tag{4.15a}$$

$$\alpha_+ = \alpha u_r/u_0, \qquad \alpha_- = \bar{\alpha} u_0/u_r, \tag{4.15b}$$

$$\beta_+ = [0 \ \cdots \ 0 \ \bar{u}_0 \ \cdots \ \bar{u}_{r-1}]x_+, \qquad \beta_- = [\bar{u}_1 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0]x_-, \tag{4.15c}$$

and α *is defined by* (4.6).

Proof. We have

$$\mathscr{A}'\begin{bmatrix} x_+ \\ 0 \\ \hline \xi_+ \\ 0 \end{bmatrix} = \begin{bmatrix} e_0 + \rho_+ e_n \\ \hline \beta_+ e_\kappa \end{bmatrix} \quad \text{and} \quad \mathscr{A}'\begin{bmatrix} 0 \\ x_- \\ \hline 0 \\ \xi_- \end{bmatrix} = \begin{bmatrix} \rho_- e_0 + e_n \\ \hline \beta_- e_0 \end{bmatrix}, \tag{4.16}$$

where

$$\rho_+ = [a_n \quad \cdots \quad a_1]x_+, \qquad \rho_- = [a_{-1} \quad \cdots \quad a_{-n}]x_-. \tag{4.17}$$

Hence

$$\mathscr{A}'\begin{bmatrix} x_+^0 \\ \xi_+ \\ \hline 0 \end{bmatrix} = \begin{bmatrix} e_0 + \rho_+ e_n \\ \hline 0 \end{bmatrix} \quad \text{and} \quad \mathscr{A}'\begin{bmatrix} x_-^0 \\ 0 \\ \hline \xi_- \end{bmatrix} = \begin{bmatrix} \rho_- e_0 + e_n \\ \hline 0 \end{bmatrix}.$$

This leads to (4.14) with $\alpha_\pm$ being replaced by $\rho_\pm$. It remains to show that $\alpha_\pm = \rho_\pm$.

Note that (4.16) implies the following recursion for $\xi_\pm$:

$$[\xi'_+(\lambda) \quad \xi'_-(\lambda)] = \frac{1}{1 - \rho_+\rho_-}[\xi_+(\lambda) \quad \lambda\xi_-(\lambda)]\begin{bmatrix} 1 & -\rho_- \\ -\rho_+ & 1 \end{bmatrix}. \tag{4.18}$$

On the other hand according to Proposition 3.3 we have

$$\xi_+ = u_r c, \quad \xi_- = u_0 \hat{c}, \quad \xi'_+ = u_r c_{\kappa+1}, \quad \xi'_- = u_0 \hat{c}_{\kappa+1}.$$

Taking (4.1) into account, we obtain

$$[\xi'_+(\lambda) \quad \xi'_-(\lambda)] = \frac{1}{1-|\alpha^2|}[\xi_+(\lambda) \quad \lambda\xi_-(\lambda)]\begin{bmatrix} 1 & -\bar{\alpha}u_0/u_r \\ -\alpha u_r/u_0 & 1 \end{bmatrix}.$$

A comparison with (4.18) yields

$$\rho_+ = \alpha u_r/u_0 = \alpha_+ \quad \text{and} \quad \rho_- = \bar{\alpha}u_0/u_r = \alpha_-. \qquad \square$$

Remark 4.3. For constants $\beta_\pm$ defined by (4.15c), the following is true:

$$\beta_+ = (z_-)_{n-1}, \qquad \beta_- = (z_+)_0.$$

In fact, since $x_+ = A^+ e_0$ and $J_n A^+ J_n = (A^+)^T$, we have

$$\begin{aligned} \beta_+ &= [0 \quad \cdots \quad 0 \quad \bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]A^+ e_0 \\ &= e_0^T J_n A^+ J_n [0 \quad \cdots \quad 0 \quad \bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]^T \\ &= e_{n-1}^T A^+ [\bar{u}_{r-1} \quad \cdots \quad \bar{u}_0 \quad 0 \quad \cdots \quad 0]^T \\ &= e_{n-1}^T z_-. \end{aligned}$$

The second relation is proved analogously.

The treatment of Case (II) will be started with some preliminary consideration.

LEMMA 4.3. *Suppose that* $\sigma_+ \neq 0$ *and* $\sigma_- = 0$. *Then* $\xi'_+ = 0$ *and*

$$x'_+(\lambda) = \frac{1}{\varepsilon\sigma_+}(\lambda w_-(\lambda) - \bar{\alpha} w_+(\lambda)), \tag{4.19}$$

where

$$\varepsilon = (\hat{c})_{\kappa-1} = \frac{1}{\sigma_+}[a_{-1} \quad \cdots \quad a_{-n}]w_- \tag{4.20a}$$

and

$$\bar{\alpha} = [t_{-1} \quad \cdots \quad t_{-\kappa}]\hat{c} = [\bar{u}_1 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0]w_-. \tag{4.20b}$$

Proof. We have

$$\mathscr{A}'\begin{bmatrix} w_+ \\ 0 \\ \hline 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \hline e_0 \end{bmatrix} \quad \text{and} \quad \mathscr{A}'\begin{bmatrix} 0 \\ w_- \\ \hline 0 \end{bmatrix} = \begin{bmatrix} \rho_1 e_0 \\ \hline \rho_2 e_0 \end{bmatrix},$$

where according to Proposition 3.2,

$$\rho_1 = [a_{-1} \quad \cdots \quad a_{-n}]w_- = [a_{-1} \quad \cdots \quad a_{-n}]M\hat{c} = \sigma_+ e_{\kappa-1}^T \hat{c} = \varepsilon\sigma_+$$

and

$$\rho_2 = [\bar{u}_1 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0]w_- = [\bar{u}_1 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0]M\hat{c} = [t_{-1} \quad \cdots \quad t_{-\kappa}]\hat{c} = \bar{\alpha}.$$

This implies that

$$\mathscr{A}'\begin{bmatrix} x'_+ \\ \hline 0 \end{bmatrix} = \begin{bmatrix} e_0 \\ \hline 0 \end{bmatrix},$$

where x'_+ is given by (4.19), which is just our assertion. $\square$

THEOREM 4.3. *Assume that* $\sigma_+ \neq 0$ *and* $\sigma_- = 0$. *Then vectors* x'_- *and* y'_+ *can be computed via the formulas*

$$x'_-(\lambda) = \lambda x_-(\lambda) + \left(\frac{\bar{\alpha}\tau_-}{\varepsilon\sigma_+} - \beta_-\right)w_+(\lambda) - \frac{\tau_-}{\varepsilon\sigma_+}\lambda w_-(\lambda) \tag{4.21}$$

and

$$y'_+(\lambda) = \lambda y_+(\lambda) + \left(\frac{\bar{\alpha}(\phi_+ - a_{-n-1})}{\varepsilon\sigma_+} - \gamma_+\right)w_+(\lambda) - \frac{\phi_+ - a_{-n-1}}{\varepsilon\sigma_+}\lambda w_-(\lambda), \tag{4.22}$$

where

$$\begin{aligned} \tau_- &= [a_{-1} \ \cdots \ a_{-n}]x_- = (y_+)_0, \qquad \phi_+ = [a_{-1} \ \cdots \ a_{-n}]y_+, \\ \beta_- &= [\bar{u}_1 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0]x_-, \qquad \gamma_+ = [\bar{u}_1 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0]y_+, \end{aligned} \tag{4.23}$$

and $\bar{\alpha}$, ε *are defined by* (4.20).

Proof. We have

$$\mathscr{A}'\begin{bmatrix} 0 \\ \hline x_- \\ \hline \xi_- \end{bmatrix} = \begin{bmatrix} \tau_- e_0 + e_n \\ \hline \beta_- e_0 \end{bmatrix} \quad \text{and} \quad \mathscr{A}'\begin{bmatrix} 0 \\ \hline y_+ \\ \hline \eta_+ \end{bmatrix} = \begin{bmatrix} \phi_+ e_0 + g' \\ \hline \gamma_+ e_0 \end{bmatrix},$$

where $g' = [0 \ \ a_{-n} \ \cdots \ a_{-1}]^T$ and β_-, γ_+, ϕ_+, τ_- are given by (4.23). From these relations, we conclude that

$$x'_-(\lambda) = \lambda x(\lambda) - \tau_- x'_+(\lambda) - \beta_- w_+(\lambda)$$

and

$$y'_+(\lambda) = \lambda y_+(\lambda) - (\phi_+ - a_{-n-1})x'_+(\lambda) - \gamma_+ w_+(\lambda).$$

Taking (4.19) into account, we obtain (4.21) and (4.22).

It remains to prove that $\tau_- = (y_+)_0$. However, we have $x_- = A^+ e_{n-1}$. Hence

$$\tau_- = [a_{-1} \ \cdots \ a_{-n}]A^+ e_{n-1}.$$

In view of

$$J_n A^+ J_n = (A^+)^T,$$

we conclude that

$$\tau_- = e_{n-1}^T J_n A^+ J_n [a_{-1} \ \cdots \ a_{-n}]^T = e_0^T A^+ [a_{-n} \ \cdots \ a_{-1}]^T = e_0^T y_+,$$

which is the assertion. □

Remark 4.4. If there is a change from Case (I) to Case (II), vector y_+ appearing in (4.22) is not immediately available, but it can be computed with the help of (3.27) since in this case we still have $u_r = 0$.

Case (III) can be treated analogously to Case (II). We formulate only the corresponding results.

LEMMA 4.4. *Suppose that* $\sigma_+ = 0$ *and* $\sigma_- \neq 0$. *Then* $\xi'_- = 0$ *and*

$$x'_-(\lambda) = \frac{1}{\varepsilon\sigma_-}(w_+(\lambda) - \alpha\lambda w_-(\lambda)), \tag{4.24}$$

where

$$\varepsilon = (c_+)_0 = \frac{1}{\sigma_-}[a_n \quad \cdots \quad a_1]w_+ \tag{4.25a}$$

and

$$\alpha = [t_\kappa \quad \cdots \quad t_1]c = [0 \quad \cdots \quad 0 \quad \bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]w_+. \tag{4.25b}$$

Moreover, α and ε are the same as in Lemma 4.3.

THEOREM 4.4. *Assume that $\sigma_+ = 0$ and $\sigma_- \neq 0$. Then the vectors x'_+ and y'_- can be computed via the formulas*

$$x'_+(\lambda) = x_+(\lambda) - \frac{\tau_+}{\varepsilon\sigma_-} w_+(\lambda) + \left(\frac{\alpha\tau_+}{\varepsilon\sigma_-} - \beta_+\right)\lambda w_-(\lambda) \tag{4.26}$$

and

$$y'_-(\lambda) = y_-(\lambda) - \frac{\phi_- - a_{n+1}}{\varepsilon\sigma_-} w_+(\lambda) + \left(\frac{\alpha(\phi_- - a_{n+1})}{\varepsilon\sigma_-} - \gamma_-\right)\lambda w_-(\lambda), \tag{4.27}$$

where

$$\begin{aligned} \tau_+ &= [a_n \quad \cdots \quad a_1]x_+ = (y_-)_{n-1}, \qquad \phi_- = [a_n \quad \cdots \quad a_1]y_-, \\ \beta_+ &= [0 \quad \cdots \quad 0 \quad \bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]x_+, \qquad \gamma_+ = [0 \quad \cdots \quad 0 \quad \bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]y_-, \end{aligned} \tag{4.28}$$

and α, ε are defined by (4.25).

It remains to consider Case (IV).

LEMMA 4.5. *Assume that $\sigma_+\sigma_- \neq 0$. Then $\xi'_+ = \xi'_- = 0$ and*

$$x'_+ = \frac{1}{\varepsilon\sigma_+}\begin{bmatrix} 0 \\ w \end{bmatrix}, \qquad x'_- = \frac{1}{\varepsilon\sigma_-}\begin{bmatrix} w_+ \\ 0 \end{bmatrix}, \tag{4.29}$$

where

$$\varepsilon = (c)_0 = \frac{1}{\sigma_-}[a_n \quad \cdots \quad a_1]w_+ = \frac{1}{\sigma_+}[a_{-1} \quad \cdots \quad a_{-n}]w_-. \tag{4.30}$$

Proof. We have

$$\mathscr{A}'\begin{bmatrix} w_+ \\ 0 \\ \hline 0 \end{bmatrix} = \begin{bmatrix} \rho_1 e_n \\ \hline 0 \end{bmatrix} \quad \text{and} \quad \mathscr{A}'\begin{bmatrix} 0 \\ w_- \\ \hline 0 \end{bmatrix} = \begin{bmatrix} \rho_2 e_0 \\ \hline 0 \end{bmatrix},$$

where

$$\rho_1 = [a_n \quad \cdots \quad a_1]w_+ = [a_n \quad \cdots \quad a_1]Mc = \sigma_- e_0^T c = \varepsilon\sigma_-$$

and

$$\rho_2 = [a_{-1} \quad \cdots \quad a_{-n}]w_- = [a_{-1} \quad \cdots \quad a_{-n}]M\hat{c} = \sigma_+ e_{n-1}^T \hat{c} = \varepsilon\sigma_+.$$

From these relations we get (4.29) immediately. □

Now we can establish a recursion from $w_\pm$, $y_\pm$ to $y'_\pm$. However, the formula will be a little nicer for $y_\pm$, $y'_\pm$ being replaced by $\tilde{y}_\pm$, $\tilde{y}'_\pm$, which are the first parts of the corresponding vectors of the canonical fundamental systems and which are defined by

$$\begin{aligned} \tilde{y}_+(\lambda) &= y_+(\lambda) - \lambda^n, \qquad \tilde{y}'_+(\lambda) = y'_+(\lambda) - \lambda^{n-1}, \\ \tilde{y}_-(\lambda) &= -1 + \lambda y_-(\lambda), \qquad \tilde{y}'_-(\lambda) = -1 + \lambda y'_-(\lambda). \end{aligned} \tag{4.31}$$

THEOREM 4.5. *Assume that* $\sigma_+\sigma_- \neq 0$. *Then the vectors* $\tilde{y}'_\pm$ *can be computed from* $\tilde{y}_\pm$ *and* $w_\pm$ *with the help of the formulas*

$$\tilde{y}'_+(\lambda) = y^0_+(\lambda) + \frac{a_{-n-1} - \nu_+}{\varepsilon\sigma_+}\lambda w_-(\lambda),$$

$$\tilde{y}'_-(\lambda) = y^0_-(\lambda) + \frac{a_{n+1} - \nu_-}{\varepsilon\sigma_-} w_+(\lambda), \tag{4.32}$$

where

$$[y^0_+(\lambda) \quad y^0_-(\lambda)] = [\tilde{y}_+(\lambda) \quad \tilde{y}_-(\lambda)]\begin{bmatrix}\lambda & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & -\mu_-\\ -\mu_+ & 1\end{bmatrix} \tag{4.33}$$

and

$$\mu_+ = \frac{\delta\sigma_+}{\sigma_-}, \quad \mu_- = \frac{\bar{\delta}\sigma_-}{\sigma_+}, \quad \delta = \frac{(c)_{\kappa-1}}{(c)_0}, \tag{4.34}$$

$$\begin{aligned} \nu_+ &= \phi_+ - \mu_+(\psi_- - a_0), & \nu_- &= \phi_- - \mu_-(\psi_+ - a_0),\\ \phi_+ &= [a_{-1} \quad \cdots \quad a_{-n}]y_+, & \psi_- &= [a_{-1} \quad \cdots \quad a_{-n}]y_-,\\ \psi_+ &= [a_n \quad \cdots \quad a_1]y_+, & \phi_- &= [a_n \quad \cdots \quad a_1]y_-. \end{aligned} \tag{4.35}$$

Moreover,

$$\psi_- = \psi_+. \tag{4.36}$$

Proof. Let $V\eta_\pm$, $\Lambda\eta_\pm$ denote the vectors obtained from $\eta_\pm$ after cancelling the first or last component, respectively. Then we observe that

$$\mathscr{A}'\begin{bmatrix}0\\ y_+\\ \hline \Lambda\eta_+\end{bmatrix} = \begin{bmatrix}\phi_+e_0 + g^0_+\\ \hline 0\end{bmatrix} - (\eta_+)_{\kappa-1}\begin{bmatrix}h'_+\\ \hline 0\end{bmatrix},$$

$$\mathscr{A}'\begin{bmatrix}-1\\ y_-\\ \hline \Lambda\eta_-\end{bmatrix} = \begin{bmatrix}(\psi_- - a_0)e_0\\ \hline 0\end{bmatrix} - (\eta_-)_{\kappa-1}\begin{bmatrix}h'_+\\ \hline 0\end{bmatrix},$$

and

$$\mathscr{A}'\begin{bmatrix}y_+\\ -1\\ \hline V\eta_+\end{bmatrix} = \begin{bmatrix}(\psi_+ - a_0)e_{n-1}\\ \hline 0\end{bmatrix} - (\eta_+)_0\begin{bmatrix}h'_-\\ \hline 0\end{bmatrix},$$

$$\mathscr{A}'\begin{bmatrix}y_-\\ 0\\ \hline V\eta_-\end{bmatrix} = \begin{bmatrix}\phi_-e_{n-1} + g^0_-\\ \hline 0\end{bmatrix} - (\eta_-)_0\begin{bmatrix}h'_-\\ \hline 0\end{bmatrix},$$

where

$$\begin{aligned} h'_+ &= [0 \quad \cdots \quad 0 \quad \bar{u}_r \quad \cdots \quad \bar{u}_0]^T, & h'_- &= [\bar{u}_r \quad \cdots \quad \bar{u}_0 \quad 0 \quad \cdots \quad 0]^T,\\ g^0_+ &= [0 \quad a_{-n} \quad \cdots \quad a_{-1}]^T, & g^0_- &= [a_1 \quad \cdots \quad a_n \quad 0]^T \end{aligned}$$

and $\phi_\pm$, $\psi_\pm$ are defined by (4.35).

We introduce auxiliary vectors

$$\left[\begin{array}{c} y_+^0 \\ \hline \eta_+^0 \end{array}\right] := \left[\begin{array}{c} 0 \\ \hline y_+ \\ \hline \Lambda\eta_+ \end{array}\right] - \mu_+ \left[\begin{array}{c} -1 \\ \hline y_- \\ \hline \Lambda\eta_- \end{array}\right], \qquad \left[\begin{array}{c} y_-^0 \\ \hline \eta_-^0 \end{array}\right] := \left[\begin{array}{c} y_- \\ \hline 0 \\ \hline V\eta_- \end{array}\right] - \mu_- \left[\begin{array}{c} y_+ \\ \hline -1 \\ \hline V\eta_+ \end{array}\right], \tag{4.37}$$

where

$$\mu_+ = (\eta_+)_{\kappa-1}/(\eta_-)_{\kappa-1}, \qquad \mu_- = (\eta_-)_0/(\eta_+)_0. \tag{4.38}$$

Then we have

$$\begin{aligned} \mathscr{A}' \left[\begin{array}{c} y_+^0 \\ \hline \eta_+^0 \end{array}\right] &= \left[\begin{array}{c} (\phi_+ - \mu_+(\psi_- - a_0))e_0 + g_+^0 \\ \hline 0 \end{array}\right], \\ \mathscr{A}' \left[\begin{array}{c} y_-^0 \\ \hline \eta_-^0 \end{array}\right] &= \left[\begin{array}{c} (\phi_- - \mu_-(\psi_+ - a_0))e_{n-1} + g_-^0 \\ \hline 0 \end{array}\right]. \end{aligned} \tag{4.39}$$

Hence

$$y'_+ = y_+^0 + (a_{-n-1} - \phi_+ + \mu_+(\psi_- - a_0))x'_+$$

and

$$y'_- = y_-^0 + (a_{n+1} - \phi_- + \mu_-(\psi_+ - a_0))x'_-.$$

Taking Lemma 4.5 into account, we obtain (4.32).

Relation (4.33) follows from (4.37) for $\mu_\pm$ being defined by (4.38). It remains to show that $\mu_\pm$ can also be expressed by (4.34).

By Proposition 3.4, we have $\eta_+ = \sigma_+ c$, $\eta_- = \sigma_- \hat{c}$. Hence

$$(\eta_+)_{\kappa-1} = \sigma_+(c)_{\kappa-1} = \delta\varepsilon\sigma_+, \qquad (\eta_-)_{\kappa-1} = \sigma_-(\hat{c})_{\kappa-1} = \varepsilon\sigma_-,$$
$$(\eta_+)_0 = \sigma_+(c)_0 = \varepsilon\sigma_+, \qquad (\eta_-)_0 = \sigma_-(\hat{c})_0 = \bar{\delta}\varepsilon\sigma_-,$$

where

$$\varepsilon = (c)_0.$$

This implies the equivalence of (4.38) and (4.34).

It remains to prove (4.36). We have

$$\psi_- = [a_{-1} \quad \cdots \quad a_{-n}]A^+[a_1 \quad \cdots \quad a_n]^T.$$

In view of $J_n A^+ J_n = (A^+)^T$, we conclude that

$$\psi_- = [a_n \quad \cdots \quad a_1]A^+[a_{-n} \quad \cdots \quad a_{-1}]^T = [a_n \quad \cdots \quad a_1]y_+ = \psi_+. \qquad \square$$

Remark 4.5. Constants $\nu_\pm$ can also be expressed as

$$\nu_+ = [a_0 \quad \cdots \quad a_{-n}]y_+^0, \qquad \nu_- = [a_n \quad \cdots \quad a_0]y_-^0.$$

That means, for the computation of $y'_\pm$, only two inner product calculations are needed.

If A is Hermitian, then all recursions can be simplified. First we recall that, in this case, we need only the recursion of Theorems 4.1, 4.2, and 4.5. In the case of a semidefinite A, only Theorems 4.1 and 4.2 must be applied. Furthermore,

$$x_- = \hat{x}_+, \quad y_- = \hat{y}_+, \quad w_- = \hat{w}_+, \quad z_- = \hat{z}_+,$$

and the same is true, of course, for the corresponding vectors for $\mathscr{A}'$. Moreover,

$$\beta_- = \bar{\beta}_+ \quad \text{and} \quad \phi_- = \bar{\phi}_+.$$

5. Complete nested recursion.

5.1. Classification. In this section, we consider the family of all principal submatrices

$$A^{(k)} = [a_{i-j}]_0^{k-1} \ (k = 1, \ldots, n)$$

of Toeplitz matrix A given by (1.1). All quantities introduced in previous sections for matrix A are denoted by the same symbol in the case of matrix $A^{(k)}$ supplemented by superscript (k). For example, $\mathscr{A}^{(k)}$ is the regular extension of $A^{(k)}$ of the form (2.6), $x_\pm^{(k)}$, $y_\pm^{(k)}$, $w_\pm^{(k)}$, $z_\pm^{(k)}$ denote the first parts of the solutions of fundamental equations (2.10) and (2.12) for $\mathscr{A}$ replaced by $\mathscr{A}^{(k)}$. Furthermore, $u^{(k)}$ denotes the vector spanning ker $A^{(k)}$ according to Lemma 2.2, $\kappa^{(k)} = \dim \ker A^{(k)}$, and

$$\sigma_+^{(k)} := [a_{-1} \quad \cdots \quad a_{-k}]\begin{bmatrix} 0 \\ u^{(k)} \end{bmatrix}, \qquad \sigma_-^{(k)} := [a_k \quad \cdots \quad a_1]\begin{bmatrix} u^{(k)} \\ 0 \end{bmatrix}. \tag{5.1}$$

Thereby, 0 denotes a zero vector of length $\kappa^{(k)} - 1$; a_n and a_{-n} are arbitrary but fixed numbers.

The aim of this section is to establish a procedure for the computation of the vectors involved in the matrix representation of A^+ (cf. Theorem 2.1), which is based on a recursion of fundamental systems of the sequence of Toeplitz mosaic matrices $\mathscr{A}^{(k)}$.

Let r denote the largest index k for which matrix $A^{(k)}$ is nonsingular. If all $A^{(k)}$ are singular, we put $r = 0$. Then $\dim \ker A^{(r+1)} = 1$. Suppose that

$$\ker A^{(r+1)} = \operatorname{lin}\{u\}, \qquad u = (u_i)_0^r. \tag{5.2}$$

Then $u_0 u_r \neq 0$.

It is a remarkable fact that, with the help of the vector u, the kernels of all matrices $A^{(k)}$ for $k > r$ can be described. For this we still introduce the numbers

$$\sigma_{k,-} := [a_k \quad \cdots \quad a_{k-r}]u, \qquad \sigma_{k,+} := [a_{r-k} \quad \cdots \quad a_{-k}]u. \tag{5.3}$$

Clearly, $\sigma_{k,-} = \sigma_{k,+}$ for $k = 0, \ldots, r$. Let s denote the smallest integer for which $\sigma_{k,-} \neq 0$ and t the smallest integer for which $\sigma_{k,+} \neq 0$. In case that all $\sigma_{k,-}$ or $\sigma_{k,+}$ vanish, we set $s = n$ or $t = n$, respectively.

LEMMA 5.1. *Let $u^{(k)}$ denote the vector spanning* ker $A^{(k)}$ *according to Lemma* 2.2. *Then*:

(I) *For* $r < k < \min\{s, t\}$

$$u^{(k+1)} = u^{(k)}, \qquad \kappa^{(k+1)} = \kappa^{(k)} + 1.$$

(II) *For* $t \leq k < s$

$$u^{(k+1)} = \begin{bmatrix} u^{(k)} \\ 0 \end{bmatrix}, \qquad \kappa^{(k+1)} = \kappa^{(k)}.$$

(III) *For* $s \leq k < t$

$$u^{(k+1)} = \begin{bmatrix} 0 \\ u^{(k)} \end{bmatrix}, \qquad \kappa^{(k+1)} = \kappa^{(k)}.$$

(IV) *For* $\max\{s, t\} \leq k$

$$u^{(k+1)} = \begin{bmatrix} 0 \\ u^{(k)} \\ 0 \end{bmatrix}, \qquad \kappa^{(k+1)} = \kappa^{(k)} - 1.$$

The assertion of the lemma follows from the considerations in [11, I.5.9].

As a consequence we get the following corollary.

COROLLARY 5.1. *The following relations hold true.*

$$\sigma_+^{(k)} = \begin{cases} \sigma_{k,+} = 0 & r \le k < t, \\ \sigma_{t,+} \ne 0 & t \le k, \end{cases} \qquad \sigma_-^{(k)} = \begin{cases} \sigma_{k,-} = 0 & r \le k < s, \\ \sigma_{s,-} \ne 0 & s \le k. \end{cases}$$

That means the recursion part of the algorithm involves merely numbers $\sigma_+ := \sigma_{t,+}$ and $\sigma_- := \sigma_{s,-}$.

Furthermore, we may conclude the following corollary.

COROLLARY 5.2. *It holds that*

(a)
$$(u^{(k)})_0 = u_0 \ne 0 \quad \text{for } r < k \le s,$$
$$(u^{(k)})_0 = 0 \qquad \text{for } s < k.$$

(b)
$$(u^{(k)})_{r(k)-1} = u_r \ne 0 \quad \text{for } r < k \le t,$$
$$(u^{(k)})_{r(k)-1} = 0 \qquad \text{for } t < k, \text{ where } r(k) = k - \kappa^{(k)}.$$

Now we can classify matrices $A^{(k)}$ according to Cases (I)–(IV) of System (A) in Theorem 3.1.

$$\text{Case (I) } ((u^{(k)})_0 (u^{(k)})_{r(k)-1} \ne 0)\text{: } r < k \le \min\{s, t\}$$
$$\text{Case (II) } ((u^{(k)})_0 \sigma_+^{(k)} \ne 0)\text{: } t \le k \le s$$
$$\text{Case (III) } (\sigma_-^{(k)} (u^{(k)})_{r(k)-1} \ne 0)\text{: } s \le k \le t$$
$$\text{Case (IV) } (\sigma_-^{(k)} \sigma_+^{(k)} \ne 0)\text{: } \max\{s, t\} \le k.$$

Let us point out that the cases have some nonempty intersections that are levels t and s. This allows us to change the fundamental system with the help of Proposition 3.6 as it is required for the continuation of the recursive procedure.

Now we are able to describe an algorithm for MPI computation.

5.2. Initialization. The first step is the computation of the integer r, the vector u satisfying the conditions of Lemma 2.2, and the quantities $\sigma_{k,-}$ and $\sigma_{k,+}$ for $k = r+1$, $r+2, \ldots$ up to the first nonzero number. This can be done by standard algorithms of the Levinson type for Toeplitz matrix inversion. In the Appendix we present a version that is convenient for our purposes and is a slight modification of the algorithm proposed in [10] (for Hankel matrices see [9]).

The second step consists in the computation of first parts of the fundamental system of the matrix

$$\mathscr{A}^{(r+1)} = \begin{bmatrix} A^{(r+1)} & \hat{u} \\ u^* & 0 \end{bmatrix}.$$

Since we have for $k = r+1$ Case (I) of Theorem 3.1, we must evaluate $x_\pm^{(r+1)}$ and $w_\pm^{(r+1)}$. This can be done by the following relations.

PROPOSITION 5.1. *It holds that*

(5.4)
$$\text{(a)} \quad w_+^{(r+1)} = w_-^{(r+1)} = \frac{1}{u^* u}\, u.$$

(b) *If $r > 0$ and $p_\pm$ are the solutions of the equations*

(5.5)
$$A^{(r)} p_+ = (\bar{u}_{r-i})_{i=1}^{r}, \qquad A^{(r)} p_- = (\bar{u}_{r-i})_{i=0}^{r-1},$$

then

$$\rho_+ := [a_{-1} \quad \cdots \quad a_{-r}]p_+ - \bar{u}_r \neq 0, \qquad \rho_- := [a_r \quad \cdots \quad a_1]p_- - \bar{u}_0 \neq 0 \tag{5.6}$$

and

$$x_+^{(r+1)} = \frac{1}{\rho_+}\left(\begin{bmatrix} 0 \\ p_+ \end{bmatrix} - \theta_+ u\right), \qquad x_-^{(r+1)} = \frac{1}{\rho_-}\left(\begin{bmatrix} p_- \\ 0 \end{bmatrix} - \theta_- u\right), \tag{5.7}$$

where

$$\theta_+ = \frac{1}{u^*u}[\bar{u}_1 \quad \cdots \quad \bar{u}_r]p_+, \qquad \theta_- = \frac{1}{u^*u}[\bar{u}_0 \quad \cdots \quad \bar{u}_{r-1}]p_-. \tag{5.8}$$

(c) *If* $r = 0$ *then* $x_\pm^{(r+1)} = 0$.

Proof. Obviously,

$$\mathscr{A}^{(r+1)}\begin{bmatrix} u \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ u^*u \end{bmatrix}, \tag{5.9}$$

which implies (5.4).

Furthermore, we have

$$\mathscr{A}^{(r+1)}\begin{bmatrix} 0 \\ p_+ \\ -1 \end{bmatrix} = \begin{bmatrix} \rho_+ \\ 0 \\ \theta'_+ \end{bmatrix} \quad \text{and} \quad \mathscr{A}^{(r+1)}\begin{bmatrix} p_- \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ \rho_- \\ \theta'_+ \end{bmatrix},$$

where $\theta'_\pm = u^*u\theta_\pm$ and $\rho_\pm$, $\theta_\pm$ are defined by (5.6) and (5.8). Taking (5.9) into account, we obtain (5.7).

Assertion (c) is obvious. □

Vectors $p_\pm$ can be computed with the help of a standard recursive algorithm for the solution of Toeplitz systems or with the help of a matrix representation (e.g., a Gohberg–Semencul formula) for $(A^{(r)})^{-1}$ (see the Appendix).

5.3. Recursion. We arrive at the main part of the algorithm, which is the recursion of the first parts of fundamental systems of $\mathscr{A}^{(k)}$.

For $k \leq \min\{s, t\}$ we successively compute vectors $w_\pm^{(k)}$ and $x_\pm^{(k)}$ according to formulas (4.5) and (4.14), (4.15) for $w_\pm$, $x_\pm$ being replaced by $w_\pm^{(k)}$ and $x_\pm^{(k)}$, $w'_\pm$, $x'_\pm$ being replaced by $w_\pm^{(k+1)}$, $x_\pm^{(k+1)}$ and $\kappa = k - r$.

If $k = \min\{s, t\}$, then we use Corollary 3.2 to change to one of the Cases (II), (III), or (IV) depending on whether $t < s$, $s < t$, or $s = t$.

For $t \leq k \leq s$, the $w_\pm^{(k)}(\lambda)$ remain unchanged, and the $x_-^{(k)}$, $y_+^{(k)}$ are computed with the help of (4.21), (4.22). Analogously, case $s \leq k \leq t$ is treated using recursions (4.8), (4.26), and (4.27).

If $k = \max\{s, t\}$, then we must employ Corollary 3.2 to compute y_+ or y_- of the corresponding level.

As a result of the described recursive procedure, we get the first parts of the fundamental systems of $\mathscr{A}^{(n)} = \mathscr{A}$ according to Cases (I)–(IV) of Theorem 3.1. Let us note that these cases can equivalently be characterized by

$$\text{(I) } s = t = n, \quad \text{(II) } t < s < n, \quad \text{(III) } s < t < n, \quad \text{(IV) } s, t < n. \tag{5.10}$$

This follows from considerations at the end of §5.1.

In the subsequent section, it will be seen that the presented recursion formulas will be useful further on.

5.4. Canonical fundamental system. The final step will be the computation of one of the canonical fundamental systems. Since the formulas for A^+ involve only the first parts of the corresponding vectors, it suffices to evaluate them. We show how to determine the first parts of the (+)-system.

Of course, we must distinguish the four Cases (I)–(IV) listed in (5.10) (or Theorem 3.1). In these cases the following vectors must be computed:

$$\begin{aligned}
&\text{(I)} && y_+,\ z_+ && \text{from } x_\pm,\ w_\pm,\\
&\text{(II)} && x_+,\ z_+ && \text{from } x_-,\ y_+,\ w_\pm,\\
&\text{(III)} && y_+,\ z_+ && \text{from } x_+,\ y_-,\ w_\pm,\\
&\text{(IV)} && x_+,\ z_+ && \text{from } y_\pm,\ w_\pm.
\end{aligned}$$

The computation of the corresponding linear combinations can be carried out by applying matrix Φ_+ introduced in §3 to the corresponding vectors. This leads to two 4×4 systems of simple form that can be solved explicitly. However, the explicit formulas are rather complicated. We get simpler formulas if we use vectors of levels $n-1$ and $n+1$. Vectors $x_\pm^{(n+1)}$ and $w_\pm^{(n+1)}$ are obtained by adding another recursion of the same type as from $n-1$ to n. In the case where a_{n+1} or a_{-n-1} occurs in the corresponding recursion, it can be chosen arbitrary. We present the formulas without proofs. The proofs are a straightforward checking.

PROPOSITION 5.2. *It holds that*

(I) *If* $s = t = n$, *then*

$$\tilde{y}_+(\lambda) = \frac{-1}{u_r \varepsilon_{\kappa+1}} w_-^{(n+1)}(\lambda), \tag{5.11}$$

$$z_+(\lambda) = \frac{-1}{u_0 \varepsilon_{\kappa+1}} \left(x_-^{(n+1)}(\lambda) - \frac{(x_-^{(n+1)})_n}{u_r \varepsilon_{\kappa+1}} w_-^{(n+1)}(\lambda) \right). \tag{5.12}$$

(II) *If* $t < s = n$, *then*

$$x_+(\lambda) = \frac{1}{\sigma_+ \varepsilon_{\kappa+1}(1 - |\alpha_\kappa|^2)} (\lambda w_-(\lambda) - \bar{\alpha}_\kappa w_+(\lambda)), \tag{5.13}$$

$$z_+(\lambda) = \frac{-1}{u_0 \varepsilon_\kappa} (x_-^{(n+1)}(\lambda) + (x_-^{(n+1)})_n \tilde{y}_+(\lambda)). \tag{5.14}$$

(III) *If* $s < t = n$, *then*

$$\tilde{y}_+(\lambda) = \frac{-1}{(x_-^{(n+1)})_n} (x_-^{(n+1)}(\lambda) - \psi w_+(\lambda)), \tag{5.15}$$

$$z_+(\lambda) = \frac{-1}{\sigma_- \varepsilon_-} (\tilde{y}_-^{(n-1)}(\lambda) - \phi x_+(\lambda)), \tag{5.16}$$

where

$$\phi = [a_0 \quad \cdots \quad a_{-1+n}] \tilde{y}_-^{(n-1)},$$

$$\psi = [\bar{u}_0 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0] x_-^{(n+1)}.$$

(IV) *If* $s, t < n$, *then*

$$x_+(\lambda) = \frac{1}{\sigma_+ \varepsilon_{\kappa+1}} \lambda w_-^{(n-1)}(\lambda), \tag{5.17}$$

$$z_+(\lambda) = \frac{-1}{\sigma_- \varepsilon_{\kappa+1}} (\tilde{y}_-^{(n-1)}(\lambda) - \phi x_+(\lambda)), \tag{5.18}$$

where

$$\phi = [a_0 \quad \cdots \quad a_{-1+n}]\tilde{y}_-^{(n-1)}.$$

Recall that

$$\tilde{y}_+(\lambda) = y_+(\lambda) - \lambda^n,$$

$$\tilde{y}_-^{(n-1)}(\lambda) = \lambda y_-^{(n-1)}(\lambda) - 1,$$

$$\varepsilon_k = (T_k^{-1} e_0)_0 \quad (k = \kappa, \kappa + 1).$$

6. Schur-type algorithm. The disadvantage of the algorithm presented in the previous section is that it involves inner product calculations at each level. Therefore it requires $O(n \log n)$ steps in parallel computation, but an amount of $O(n)$ is desirable. In this section we show that, similar to the Toeplitz matrix inversion algorithms, there are modifications without this disadvantage. The corresponding algorithms are related to the famous Schur algorithm and will be referred to as Schur type (cf. [14], [4], [10]).

For simplicity we present here only formulas of Case (I) of Theorem 3.1. Let us recall that we only have Case (I) in the event that matrix A is semidefinite. Formulas for other cases are quite similar, but a little more complicated. We do not present them here because we think that the approach offered in the subsequent section provides more effective algorithms for parallel processing.

We use all the notations of §5.

First let us note that vector u generating the kernel of A can be computed with a Schur-type version of the algorithm, described in the Appendix, which requires $O(n)$ steps at an n-processor computer (see [9], [10]).

Since we only consider Case (I), i.e., we assume $s = t = n$, we only have to apply recursions (4.5) and (4.14) for $x_\pm$, $w_\pm$ being replaced by $x_\pm^{(k)}$, $w_\pm^{(k)}$, and $x'_\pm$; $w'_\pm$ being replaced by $x_\pm^{(k+1)}$, $w_\pm^{(k+1)}$, respectively, to obtain vectors involved in the formula for the MPI of A.

The main task is now to evaluate constants $\alpha_\pm^{(k)}$ and $\beta_\pm^{(k)}$ appearing in these formulas recursively. Constants $\alpha_\pm^{(k)}$ are, by (4.15b), up to constant factors, the same as they appear in the Schur algorithm for the inversion of the positive definite Toeplitz matrix $T_{n-r} = M_{n-r}^* M_{n-r} = [t_{i-j}]_0^{n-r-1}$. We present the corresponding formulas.

We introduce parameters $\alpha_j^{(k)}$ $(j = 1 + r - n, \ldots, 2n - 2r - 2)$ by

$$\alpha_j^{(k)} := [t_j \quad \cdots \quad t_{j-k+r+1}]c_{k-r}, \tag{6.1}$$

where we set $t_j = 0$ for $j \notin \{1 + r - n, \ldots, n - r - 1\}$ and

$$c_{k-r} = T_{k-r}^{-1} e_0.$$

Then formula (4.1) leads to the following proposition.

PROPOSITION 6.1. *Numbers* $\alpha_j^{(k)}$ *fulfill the recursion*

$$\alpha_j^{(k+1)} = \frac{1}{1 - |\alpha^{(k)}|^2} (\alpha_j^{(k)} - \alpha^{(k)} \alpha_{k-j-r}^{(k)}), \tag{6.2}$$

where

$$\alpha^{(k)} := \alpha_{k-r}^{(k)}.$$

Initial values of $\alpha_j^{(k)}$ are $\alpha_j^{(r+1)} = \frac{1}{t_0} t_j$. Note that, by definition, $\alpha_j^{(k)} = 0$ for $j = 1, \ldots, k-1-r$ and $\alpha_0^{(k)} = 1$.

Formula (6.2), together with Proposition 3.2, allows us to compute the vectors $w_\pm$ in $O(n)$ parallel steps.

It remains to find a recursion for constants $\beta_\pm^{(k)}$ defined by

$$\beta_+^{(k)} := [0 \ \cdots \ 0 \ \bar{u}_0 \ \cdots \ \bar{u}_{r-1}]x_+^{(k)}, \qquad \beta_-^{(k)} := [\bar{u}_1 \ \cdots \ \bar{u}_r \ 0 \ \cdots \ 0]x_-^{(k)} \tag{6.3}$$

appearing in the recursion (4.14) for A being replaced by $A^{(k)}$.

Let N_k denote the matrix of the form (2.5) with k columns. We need the following formula.

LEMMA 6.1. *It holds that*

$$N_k w_\pm^{(k)} = a_\pm^{(k)},$$

where

$$a_+^{(k)} = (\alpha_j^{(k)})_{j=-r}^{k-1} \quad \textit{and} \quad a_-^{(k)} = \hat{a}_+^{(k)}. \tag{6.4}$$

Proof. By Proposition 3.2, we have

$$w_+^{(k)} = M_{k-r} c_{k-r} \quad \text{and} \quad w_-^{(k)} = M_{k-r} \hat{c}_{k-r}.$$

Hence

$$N_k w_+^{(k)} = N_k M_{k-r} c_{k-r} = \tilde{T}_{k-r} c_{k-r}, \tag{6.5}$$

where

$$\tilde{T}_{k-r} = \begin{bmatrix} t_{1-\kappa} & & 0 \\ \vdots & \ddots & \\ t_{k-n} & & t_{1-\kappa} \\ \vdots & & \vdots \\ t_{\kappa-1} & & t_{n-k} \\ & \ddots & \vdots \\ 0 & & t_{\kappa-1} \end{bmatrix}, \qquad (\kappa = n-r),$$

and we get

$$N_k w_-^{(k)} = \tilde{T}_{k-r} \hat{c}_{k-r}. \tag{6.6}$$

Formula (6.5) implies the first relation of (6.4) immediately. The second relation follows after taking into account that

$$\tilde{T}_{k-r} J_{k-r} = J_m \bar{\tilde{T}}_{k-r},$$

where $m = 2\kappa - r + k - 2$. □

We now introduce vectors $b_\pm^{(k)}$ by

$$b_\pm^{(k)} := N_k x_\pm^{(k)}.$$

Suppose that

$$b_\pm^{(k)} = (\beta_{\pm,j}^{(k)})_{j=-r}^{k-1}.$$

Then we have, in particular,

$$\beta_{+}^{(k)} = \beta_{+,k-r}^{(k)} \quad \text{and} \quad \beta_{-}^{(k)} = \beta_{-,-1}^{(k)}. \tag{6.7}$$

PROPOSITION 6.2. *Parameters $\beta_{\pm,j}^{(k)}$ fulfill recursions*

$$\beta_{+,j}^{(k+1)} = \frac{1}{1 - |\alpha^{(k)}|^2}\left(\beta_{+,j}^{0} - \frac{u_r}{u_0}\,\alpha^{(k)}\beta_{-,j}^{0}\right), \tag{6.8a}$$

$$\beta_{-,j}^{(k+1)} = \frac{1}{1 - |\alpha^{(k)}|^2}\left(-\frac{u_0}{u_r}\,\bar{\alpha}^{(k)}\beta_{+,j}^{0} + \beta_{-,j}^{0}\right), \tag{6.8b}$$

where

$$\beta_{+,j}^{0} := \beta_{+,j}^{(k)} - \beta_{+}^{(k)}\bar{\alpha}_{k-r-j}^{(k+1)}, \qquad \beta_{-,j}^{0} := \beta_{-,j-1}^{(k)} - \beta_{-}^{(k)}\alpha_{j}^{(k+1)}. \tag{6.9}$$

These formulas follow from (4.14) after applying N_{k+1} to both sides and taking Lemma 6.1 into account.

Now recursions (6.2), (6.8), (6.9), together with (4.5) and (4.14), provide an $O(n)$ step parallel algorithm to compute vectors $w_{\pm}$, $x_{\pm}$. Vectors $y_{\pm}$, $z_{\pm}$ are obtained according to Proposition 5.2.

7. Nonrecursive approach. In this section we present another approach to obtain vectors x_+, y_+, w_+, z_+ appearing in formula (2.11), which is based on formula (1.4) for the MPI. The complexity of the corresponding algorithm equals $O(n\ (\log n)^2)$.

The first step is the construction of a generalized inverse of A that possesses a simple structure. In [11], two approaches are offered for Hankel matrices. We choose the second approach and adopt it to the case of Toeplitz matrices. The approach is based on the following well-known fact (see [11]).

LEMMA 7.1. *Suppose that C is an $n \times n$ singular matrix with defect κ and let C_+ be a $\kappa \times n$ and C_- a $n \times \kappa$ matrix such that $[C \quad C_-]$ and $\begin{bmatrix} C \\ C_+ \end{bmatrix}$ have full rank. Then*

$$\tilde{C} = \begin{bmatrix} C & C_- \\ C_+ & C_0 \end{bmatrix}$$

is nonsingular for any $\kappa \times \kappa$ matrix C_0. Furthermore, if

$$\tilde{C}^{-1} = \begin{bmatrix} B & B_+ \\ B_- & B_0 \end{bmatrix},$$

where B is $n \times n$, then B is generalized inverse to C, i.e., $CBC = C$.

In our case of a Toeplitz matrix A, we choose $C_{\pm}$ and C_0 in such a way that $\tilde{C} = A^{(n+\kappa)} = [a_{i-j}]_0^{n+\kappa-1}$, i.e., $\tilde{C}$ is a Toeplitz extension of A. To find such an extension, we introduce the concept of a fundamental system of a singular Toeplitz matrix A.

We introduce the family of Toeplitz matrices

$$\partial^k A = [a_{i-j}]_{i=k}^{n-1}{}_{j=0}^{n-1+k} \quad (k = 0, \pm 1, \ldots, \pm(n-1)). \tag{7.1}$$

Then the following lemma is true (cf. [11]).

LEMMA 7.2. *Suppose that κ denotes the defect of A, and A is not the zero matrix. Then there are two vectors $u \in \mathbb{C}^{n-\kappa+1}$ and $v \in \mathbb{C}^{n+\kappa+1}$ such that $\mathscr{C}_k := \{x(\lambda)\colon x \in \ker \partial^k A\}$, $\mathscr{C}_n := \{x(\lambda)\colon x \in \mathbb{C}^{2n}\}$ is given by*

$$\mathscr{C}_k = \begin{cases} \operatorname{lin}\{u(\lambda), \lambda u(\lambda), \ldots, \lambda^{k-1+\kappa}u(\lambda)\} & k = 1-\kappa, \ldots, \kappa, \\ \operatorname{lin}\{u(\lambda), \ldots, \lambda^{k-1+\kappa}u(\lambda), v(\lambda), \ldots, \lambda^{k-1-\kappa}v(\lambda)\} & k = \kappa+1, \ldots, n. \end{cases} \tag{7.2}$$

From Lemma 7.2 we can conclude, in particular, that the first (last) component of one of the vectors u or v is nonzero and that the polynomials $u(\lambda)$ and $v(\lambda)$ are coprime.

Any pair of vectors u, v is said to be a *fundamental system* of A. Clearly, u is unique up to a constant factor, but v is not unique. It can be changed by linear combinations of $\lambda^i u(\lambda)$. This leads to the following remark.

Remark 7.1. Vector $v = (v_i)_0^{n+\kappa}$ can be chosen in such a way that $v_0 v_{n+\kappa} \neq 0$.

With the help of vector v, we extend A to a nonsingular matrix. We define successively

$$a_k := -\frac{1}{v_0} \sum_{i=1}^{n+\kappa} a_{k-i} v_i \quad (k = n, \ldots, n+\kappa-1) \tag{7.3a}$$

and

$$a_{-k} := -\frac{1}{v_{n+\kappa}} \sum_{i=0}^{n+\kappa-1} a_{n-k+\kappa-i} v_i \quad (k = n, \ldots, n+\kappa-1). \tag{7.3b}$$

LEMMA 7.3. *Matrix $A^{(n+\kappa)}$ is nonsingular and $\{\lambda^\kappa u(\lambda), v(\lambda)\}$ is a fundamental system of $A^{(n+\kappa)}$.*

Proof. By construction $\tilde{u}, v \in \ker \partial A^{(n+\kappa)}$, where $\tilde{u}(\lambda) = \lambda^\kappa u(\lambda)$. We still must show that $A^{(n+\kappa)}$ is nonsingular. If $\tilde{\kappa} = \dim \ker A^{(n+\kappa)} > 0$, then, by Lemma 7.2, $\ker \partial A^{(n+\kappa)}$ is the linear hull of vectors corresponding to polynomials $\lambda^i w(\lambda)$ for a certain vector w. That means that all polynomials corresponding to vectors in $\ker \partial A^{(n+\kappa)}$ would have a common zero. But $\tilde{u}(\lambda)$ and $v(\lambda)$ are coprime. Consequently, $\tilde{\kappa} = 0$. □

With the arguments of the proof of Lemma 7.3, we may check that the assumptions of Lemma 7.1 are fulfilled. Hence the following lemma is true.

LEMMA 7.4. *If $(A^{(n+\kappa)})^{-1}$ is decomposed in the form*

$$(A^{(n+\kappa)})^{-1} = \begin{bmatrix} B & B_- \\ B_+ & B_0 \end{bmatrix},$$

where B is $n \times n$, then B is a generalized inverse of A.

Now we employ the well-known results on the inverse of a nonsingular Toeplitz matrix (see [10], [11]) and obtain the following theorem.

THEOREM 7.1. *Suppose that A is a Toeplitz matrix with defect $\kappa > 0$ and $u = (u_i)_0^{n-\kappa}$, $v = (v_i)_0^{n+\kappa}$ is a fundamental system of A with $v_0 v_{n+\kappa} \neq 0$. Let a_{-n} be defined according to* (7.3b). *Then the $n \times n$ principal submatrix $B^{(n)}$ of the matrix*

$$B = \frac{1}{\sigma} \operatorname{Bez}(\lambda^\kappa u(\lambda), v(\lambda)), \tag{7.4}$$

where

$$\sigma = v_{n+\kappa}[a_{-\kappa} \quad \cdots \quad a_{-n}]u \tag{7.5}$$

is a generalized inverse of A.

The fundamental system of A can be computed by standard Toeplitz matrix algorithms (see the Appendix).

To make the application of formula (1.4) possible, we must determine orthogonal projections P and Q onto $\ker A$ and $\operatorname{Im} A$, respectively.

Since by Lemma 2.2 we have $\ker A = \operatorname{Im} M$ and M has full rank, we get the following expression for P:

$$P = MT^{-1}M^*, \tag{7.6}$$

where $T = M^*M$ is a positive definite Toeplitz matrix.

Since $I - Q$ is the orthogonal projection onto $\ker A^*$, taking (3.8) into account, we obtain

$$Q = I - NT^{-1}N^*. \tag{7.7}$$

From (1.4) and Theorem 7.1 we conclude the following corollary.

COROLLARY 7.1. *If $B^{(n)}$ is the matrix from Theorem* 7.1, *then the* MPI *of A is given by*

$$A^+ = (I - MT^{-1}M^*)B^{(n)}(I - NT^{-1}N^*). \tag{7.8}$$

Since T^{-1} can be represented as the product sum of triangular Toeplitz matrices, and other matrices involved in (7.8) are also of this type, the application of A^+ to a vector with the help of formula (7.8) requires only $O(n \log n)$ flops.

To make it possible to use the somehow simpler formula (2.11), we must evaluate vectors x_+, y_+, w_+, z_+. Vector w_+ is, according to Proposition 3.2, given by $w_+ = Mc$, where $c = T^{-1}e_0$. The other three vectors are pseudo solutions of certain equations, i.e., they can be computed with the help of (7.8).

However, it is more convenient to replace x_+, y_+, w_+, z_+ by other systems that also determine A^+. A convenient choice is $z_\pm$, $w_\pm$, which corresponds to System (B) of Theorem 3.1.

Vector w_- is given by $w_- = M\hat{c}$ and vectors $z_\pm$ by

$$z_+ = A^+h_+, \qquad z_- = A^+h_-, \tag{7.9}$$

where

$$h_+ = [0 \ \cdots \ 0 \ \bar{u}_{n-\kappa} \ \cdots \ \bar{u}_1]^T, \qquad h_- = [\bar{u}_{n-\kappa-1} \ \cdots \ \bar{u}_0 \ 0 \ \cdots \ 0]^T.$$

PROPOSITION 7.1. *Vectors $z_\pm$ defined by* (7.9) *are given by*

$$\begin{aligned} z_+ &= (I_n - MT^{-1}M^*)B^{(n)}(h_+ - Nd), \\ z_- &= (I_n - MT^{-1}M^*)B^{(n)}(h_- - N\hat{d}), \end{aligned} \tag{7.10}$$

where d is given by (3.14) *and $B^{(n)}$ by Theorem* 7.1.

Proof. We have

$$N^*h_- = (t_{1+i})_0^{\kappa-1} \quad \text{and} \quad N^*h_+ = (t_{i-\kappa})_0^{\kappa-1}.$$

Thus

$$T^{-1}N^*h_- = \hat{d} \quad \text{and} \quad T^{-1}N^*h_+ = d.$$

The rest follows from (7.8). □

It remains to compute x_+ and y_+.

PROPOSITION 7.2. *It holds that*

(a) *if $u_r = 0$, then*

$$x_+(\lambda) = \frac{1}{\sigma_+\varepsilon_{\kappa+1}(1 - |\alpha|^2)}(\bar{\alpha}w_+(\lambda) - \lambda w_-(\lambda)), \tag{7.11}$$

$$\tilde{y}_+(\lambda) = -\sigma_+\varepsilon_{\kappa+1}(\lambda z_-(\lambda) + \alpha z_+(\lambda)) - \rho_+ w_+(\lambda) - \rho_-\lambda w_-(\lambda), \tag{7.12}$$

where

$$\rho_- = -\frac{\psi + |\alpha|^2 \bar{u}_r}{1 - |\alpha|^2},$$

$$\psi = [a_{-1} \quad \cdots \quad a_{-n}]z_-,$$

$$\rho_+ = \bar{\alpha}\rho_- - \sigma_+ \varepsilon_{\kappa+1} \chi,$$

$$\chi = [\bar{u}_1 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0]z_-.$$

(b) *If* $u_r \neq 0$, *then*

$$x_+(\lambda) = -u_r \varepsilon_{\kappa+1}(\lambda z_-(\lambda) + \alpha z_+(\lambda)) + \left(\rho - \frac{\bar{\alpha}}{1 - |\alpha|^2}\right) w_+(\lambda) + \frac{(z_-)_{n-1}}{1 - |\alpha|^2} \lambda w_-(\lambda), \tag{7.13}$$

$$\tilde{y}_+(\lambda) = \frac{\sigma_+}{u_r} x_+(\lambda) - \frac{1}{\varepsilon u_r}(\lambda w_-(\lambda) - \bar{\alpha} w_+(\lambda)), \tag{7.14}$$

where

$$\rho = u_r \varepsilon_{\kappa+1}[\bar{u}_1 \quad \cdots \quad \bar{u}_r \quad 0 \quad \cdots \quad 0]z_-.$$

Recall that

$$\varepsilon_{\kappa+1} = (c_{\kappa+1})_0, \qquad \varepsilon = \varepsilon_\kappa = (c)_0,$$

$$\alpha = \alpha_\kappa = -(c_{\kappa+1})_\kappa / (c_{\kappa+1})_0,$$

$$\tilde{y}_+(\lambda) = y_+(\lambda) - \lambda^n.$$

Relation (7.4) coincides with (3.27), and (7.11) is a consequence of Proposition 3.3. The other two equalities can be verified by applying matrices $\Phi_\pm$ introduced in §3. Corresponding proofs are straightforward but cumbersome, therefore we omit the details.

Appendix. Computation of a fundamental system of a square Toeplitz matrix. For the convenience of the reader, we present here an algorithm for the computation of a fundamental system of a square Toeplitz matrix A. This algorithm can be used to get vector u, which is necessary for the initialization of the algorithm described in §5 or to establish a generalized inverse of A, which is needed to apply the approach of §7.[2]

The algorithm presented below is a slight modification of the algorithm described in [10] and the two-step Hankel matrix algorithm discussed in [9]. Different from many classical procedures, our algorithm works without additional assumption.

For brevity, we describe only the Levinson-type version of the algorithm. For the Schur-type version, which is convenient for parallelisation, a divide-and-conquer procedure, and some historical comments, we refer to [9] and [10].

As above, let $A^{(k)}$ denote the principal submatrices of A, $A^{(k)} = [a_{i-j}]_0^{k-1}$. Suppose that $\kappa_k := \dim \ker A^{(k)}$. Let $\{u^{(k)}, v^{(k)}\}$ be a fundamental system of $A^{(k)}$, where $u^{(k)}$, $v^{(k)}$ are vectors of length $k - \kappa_k + 1$, $k + \kappa_k + 1$, respectively. We show how to evaluate a fundamental system for $A^{(k+1)}$. We must distinguish different cases. Indicators for these cases are the numbers

$$\sigma_{11}^{(k)} = [a_{-\kappa_k} \quad \cdots \quad a_{-k}]u^{(k)}, \qquad \sigma_{12}^{(k)} = [a_k \quad \cdots \quad a_{\kappa_k}]u^{(k)},$$

$$\sigma_{21}^{(k)} = [a_{\kappa_k} \quad \cdots \quad a_{-k}]v^{(k)}, \qquad \sigma_{22}^{(k)} = [a_k \quad \cdots \quad a_{-\kappa_k}]v^{(k)}.$$

For brevity we omit superscript (k).

[2] The definition of the fundamental system concept is given following Lemma 7.2.

THEOREM A.1. *A fundamental system of* $A^{(k+1)}$ *is given by*

$$[u^{(k+1)}(\lambda) \quad v^{(k+1)}(\lambda)] = [u^{(k)}(\lambda) \quad v^{(k)}(\lambda)]E_k(\lambda),$$

where $E_k(\lambda)$ *and* κ_{k+1} *are defined as follows*:

(1) *Case* $\kappa_k > 0$.

(1a) $\sigma_{11} = \sigma_{12} = 0$,

$$E_k(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix}, \qquad \kappa_{k+1} = \kappa_k + 1.$$

(1b) $\sigma_{11} \neq 0 = \sigma_{12}$,

$$E_k(\lambda) = \begin{bmatrix} 1 & -\lambda^s \sigma_{21}/\sigma_{11} \\ 0 & \lambda \end{bmatrix}, \qquad \kappa_{k+1} = \kappa_k$$

with $s = 2\kappa_k + 1$.

(1c) $\sigma_{11} = 0 \neq \sigma_{12}$,

$$E_k(\lambda) = \begin{bmatrix} \lambda & -\sigma_{22}/\sigma_{12} \\ 0 & 1 \end{bmatrix}, \qquad \kappa_{k+1} = \kappa_k.$$

(1d) $\sigma_{11}\sigma_{22} \neq 0$,

$$E_k(\lambda) = \begin{bmatrix} \lambda & -\sigma_{22}/\sigma_{12} - \lambda^s \sigma_{21}/\sigma_{11} \\ 0 & 1 \end{bmatrix}, \qquad \kappa_{k+1} = \kappa_k - 1$$

with $s = 2\kappa_k$.

(2) *Case* $\kappa_k = 0$.

(2a) $\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21} \neq 0$,

$$E_k(\lambda) = \begin{bmatrix} \sigma_{22} & -\sigma_{21} \\ -\sigma_{12} & \sigma_{11} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix}, \qquad \kappa_{k+1} = 0.$$

(2b) $\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21} = 0$,

$$E_k(\lambda) = \begin{bmatrix} 1 & 0 \\ -\sigma_{11}/\sigma_{21} & \lambda \end{bmatrix} \quad \text{if } \sigma_{21} \neq 0, \qquad \kappa_{k+1} = 1$$

or

$$E_k(\lambda) = \begin{bmatrix} -\sigma_{21}/\sigma_{11} & \lambda \\ 1 & 0 \end{bmatrix} \quad \text{if } \sigma_{11} \neq 0, \qquad \kappa_{k+1} = 1.$$

In the case of a strongly nonsingular matrix A, we only have case (2a). The formula corresponding to this case is the classical Levinson recursion except for a scaling factor. Furthermore, it is worth mentioning that the algorithm essentially simplifies if matrix A is symmetric or Hermitian.

Let us note that, in numerical computation, it is useful to make some scaling during the computation to improve the numerical stability of the algorithm.

Finally, let us describe how to initialize the recursion.

If $a_0 \neq 0$, then we put

$$u^{(1)}(\lambda) = 1, \qquad v^{(1)}(\lambda) = \lambda.$$

In case that $a_{1-l} \neq 0 = a_{2-l} = \cdots = a_{l-1}$, we have $\kappa_l = l - 1$ and we may choose

$$u^{(l)}(\lambda) = 1, \qquad v^{(l)}(\lambda) = \lambda^{2l-1}.$$

In case that $a_{l-1} \neq 0 = a_{l-2} = \cdots = a_{1-l}$, we have $\kappa_l = l - 1$ and we may choose

$$u^{(l)}(\lambda) = \lambda, \qquad v^{(l)}(\lambda) = 1.$$

If $a_{1-l} \neq 0 = a_{2-l} = \cdots = a_{l-2} \neq a_{l-1}$, then $\kappa_l = l - 2$ and

$$u^{(l)}(\lambda) = \lambda, \qquad v^{(l)}(\lambda) = a_{1-l} - a_{l-1}\lambda^{2l-2}$$

is a fundamental system of $A^{(l)}$.

REFERENCES

[1] G. AMMAR AND P. GADER, *New decompositions of the inverse of a Toeplitz matrix*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods, Proc. Internat. Symp. MTNS-89, M. A. Kaashoek, J. H. van Schuppen, A. C. M. Ran, eds., Vol. III, Birkhäuser, Boston, Basel, Berlin, 1990, pp. 421–428.

[2] B. ANDERSON AND E. JURY, *Generalized Bezoutian and Sylvester matrices in multivariable linear control*, IEEE Trans. Automat. Control, AC-21, 4 (1976), pp. 551–556.

[3] A. BEN-ISRAEL AND T. N. E. GREVILLE, *Generalized Inverses: Theory and Applications*, Wiley, New York, 1974.

[4] A. BULTHEEL, *Recursive relations for block Hankel and Toeplitz systems*, J. Comput. Appl. Math., 10 (1984), pp. 301–354.

[5] P. DELSARTE, Y. V. GENIN, AND Y. G. KAMP, *A generalization of the Levinson algorithm for Hermitian Toeplitz matrices with any rank profile*, IEEE Trans. Acoust. Speech Signal Process, 33 (1985), pp. 964–971.

[6] I. GOHBERG AND A. A. SEMENCUL, *On the inversion of finite-section Toeplitz matrices and their continuous analogues*, Mat. Issled., 7 (1972), pp. 201–224. (In Russian.)

[7] G. HEINIG, *Endliche Toeplitzmatrizen und zweidimensionale Wiener–Hopf-Operatoren mit homogenem Symbol*, Math. Nachr., 82 (1978), pp. 29–68.

[8] G. HEINIG AND F. HELLINGER, *On the Bezoutian structure of the Moore–Penrose inverses of Hankel matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 629–645.

[9] G. HEINIG AND P. JANKOWSKI, *Parallel and superfast algorithms for Hankel systems of equations*, Numer. Math., 58 (1990), pp. 109–127.

[10] ———, *Fast algorithms for the solution of general Toeplitz systems*, Wiss. Z. Tech. Univ. Karl-Marx-Stadt, 32 (1990), H.1, pp. 12–17.

[11] G. HEINIG AND K. ROST, *Algebraic methods for Toeplitz-like matrices and operators*, Akademie-Verlag, Berlin, Birkhäuser, Basel, Boston, Stuttgart, 1984.

[12] G. HEINIG AND A. TEWODROS, *On the inverses of Hankel and Toeplitz mosaic matrices*, Seminar Analysis: Operator Equations and Numerical Analysis, Karl-Weierstrass-Institut, Berlin, 1988, pp. 53–65.

[13] I. S. IOHVIDOV, *Hankel and Toeplitz matrices and forms*, Nauka, Moscow, 1974. (In Russian.)

[14] T. KAILATH, *A theorem of I. Schur and its impact on modern signal processing*, in I. Schur Methods in Operator Theory and Signal Processing, I. Gohberg, ed., Birkhäuser, Basel, Boston, Stuttgart, 1986.

[15] L. LERER AND M. TISMENETSKY, *Generalized Bezoutians and the inversion problem for block Toeplitz matrices*, Integral Equations Operator Theory, 9 (1986), pp. 790–819.

[16] ———, *Generalized Bezoutians and matrix equation*, Linear Algebra Appl., 96 (1988), pp. 123–160.

[17] N. LEVINSON, *The Wiener rms error criterion in filter design and prediction*, J. Math. Phys., 25 (1947), pp. 261–278.

[18] M. MORF, B. DICKINSON, T. KAILATH, AND A. C. G. VIEIRA, *Efficient solution of covariance equations for linear prediction*, IEEE Trans. Acoust. Speech Signal Process., Vol. ASSP-25, 1977, pp. 429–433.

[19] B. PORAT, B. FRIEDLANDER, AND M. MORF, *Square root covariance ladder algorithms*, IEEE Trans. Automat. Control, Vol. AC-27, No. 4, 1982, pp. 813–829.

[20] I. K. PROUDLER, J. K. MCWHIRTER, AND T. J. SHEPHERD, *QRD-based lattice-ladder algorithm for adaptive filtering*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods, Proc. Internat. Symp. MTNS-89, M. A. Kaashoek, J. H. van Schuppen, A. C. M. Ran, eds., Vol. III, Birkhäuser, Boston, Basel, Berlin, 1990, pp. 263–274.

[21] W. F. TRENCH, *An algorithm for the inversion of finite Toeplitz matrices*, SIAM J. Appl. Math., 12 (1964), pp. 515–522.

[22] X. ZHONG, *On the Moore–Penrose inverses of Toeplitz matrices*, Linear Algebra Appl., 169 (1992), pp. 9–15.

[23] S. ZOHAR, *Toeplitz matrix inversion: The theorem of W. F. Trench*, J. Assoc. Comput. Mach., 16 (1967), pp. 592–601.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 451–478, April 1994

007

FAST TRIANGULAR FACTORIZATION AND INVERSION OF HANKEL AND RELATED MATRICES WITH ARBITRARY RANK PROFILE*

DEBAJYOTI PAL† AND THOMAS KAILATH‡

Abstract. The authors present a fast procedure for computing a "modified" triangular factorization of Hankel, quasi-Hankel (matrices congruent in a certain sense to Hankel matrices) and sign-modified quasi-Hankel (products of quasi-Hankel and signature matrices) matrices. A fast procedure for computing inverse of Hankel and quasi-Hankel matrices is also presented. A modified triangular factorization is an LDL^* factorization, where L is lower triangular with unit diagonal entries and D is a block diagonal matrix with possibly varying block sizes. Only matrices with all leading minors nonzero, often called strongly regular, will always have a purely diagonal and nonsingular D matrix. The matrices studied in this paper have diagonal blocks with a particular Hankel-(like) structure.

The algorithms presented here are obtained by extending a generating function approach of Lev-Ari and Kailath for matrices with a generalized displacement structure. A particular application of the results is a fast method of computing the rank profile and inertia of the matrices involved.

Key words. triangular factorization, Schur complement, Hankel, quasi-Hankel, Toeplitz, inversion, zero minors, rank profile, inertia, Iohvidov, displacement structure

AMS subject classifications. 15A06, 15A09, 15A23, 15A57, 65F05

1. Introduction.

Triangular factorization and inversion of a general $n \times n$ matrix requires $O(n^3)$ elementary operations. The special structure of Hankel matrices,

$$H_{n-1} = [h_{i+j}], \qquad 0 \le i, j \le n-1$$

allows one to invert them or to solve a Hankel system of linear equations by fast algorithms that require only $O(n^2)$ operations (see, e.g., Trench [18], Berlekamp [1], Kung [9], Heinig and Rost [6], Citron [3], Chun [2], Labahn, Choi, and Cabay [10], and the references therein).

However, the particular problem of explicitly computing their triangular factors in $O(n^2)$ operations did not receive as much attention. Among early investigators are Phillips [14], Rissanen [16], [17], and Gragg [4]. Later, Kung [9] and Gragg and Lindquist [5] considered this problem in the context of partial realization problems.

Recently Lev-Ari and Kailath [12] and Chun [2] considered the factorization of Hankel and quasi-Hankel (QH) matrices as a special case of their work on generalized displacement structure. A QH matrix has the form $L_t H L_t^T$, where L_t and H are lower triangular Toeplitz and Hankel, respectively. QH structure arises naturally in the process of factoring a Hankel matrix, because the Schur complement of the top left corner entry of a Hankel matrix is QH. Such matrices arise in other contexts also, e.g., in the computation of the greatest common divisor (GCD) of two polynomials and in the problem of root-distribution of polynomials with respect to (w.r.t.) the real axis. In studying the closely related imaginary axis problem we encounter products

* Received by the editors October 23, 1989; accepted for publication (in revised form) October 16, 1992. This work was supported in part by the U. S. Army Research Office contract DAAL03-86-K-0045 and the Strategic Defense Initiative Organization/Innovative Science and Technology, managed by the Army Research Office contract DAAL03-87-K-0033.

† Communications Sciences Research Division, AT&T Bell Laboratories, 101 Crawfords Corner Road, Holmdel, New Jersey 07733 (debu@research.att.com).

‡ Information Systems Laboratory, Stanford University, Stanford, California 94305 (tk@rascals.stanford.edu).

of QH matrices and certain signature matrices, which we shall call sign-modified QH (SMQH) matrices.

However, the results of [12] and [2] or those of [14], [16], and [4] are only for matrices that are *strongly regular*, i.e., with all its leading principal minors nonzero. We should mention here that a strict triangular factorization is not possible unless all the leading minors are nonzero. What can be achieved otherwise is a modified triangular factorization (MTF) of the form

$$M = LD_BL^T,$$

where L is lower triangular with unit diagonal, while D_B is a block diagonal matrix with possibly varying block sizes.

Gragg [4] mentions the possibility of block triangularization via generalized Lanczos polynomials, the so-called lower triangular Hankel structures of the block diagonal entries and their relationship to nontrivial blocks in the Padé table. However Gragg [4] does not provide an algorithm for computing the sought block triangular factorization. These issues have been studied further in Gragg and Lindquist [5], where the algorithm of Kung [9] (which is mentioned in the next paragraph) was used to obtain the results. Rissanen [15] studied the factorization of Hankel matrices that are not strongly regular; however, while his method is recursive it is not fast (i.e., not an $O(n^2)$ procedure). Then, in 1974, Rissanen [17] derived a fast (i.e., $O(n^2)$) procedure for factoring nonstrongly regular Hankel matrices. Although this procedure is recursive, the need for computation of an inner product at each step it makes does not parallelize well (i.e., each recursive step requires a constant number of parallel steps so that the whole algorithm takes $O(n)$ parallel steps) and requires $O(n \log n)$ parallel steps.

Kung [9] presented a method that is fast as well as recursive; however, his procedure is not completely recursive in the sense that it utilizes the block pivots in the presence of vanishing principal minors. Moreover, Kung's procedure requires computation of an inner product at each step, so that this algorithm also does not parallelize well (as explained earlier). The determination of the number of consecutive zero principal minors for a block step is not straightforward in this procedure, and it requires computation of certain "predicted Markov parameters" until a mismatch between the "predicted" and the "given" Markov parameters is observed (see [9] for details).

Recently Citron [3] refined Kung's method to avoid block pivots and the computation of inner products; the determination of the number of consecutive zero minors for a block step was also greatly simplified.

In this paper we present a recursive $O(n^2)$ algorithm for such a factorization. Possibly this is the first paper to give a fast and completely recursive procedure for computing an MTF of Hankel, QH, and SMQH matrices. Furthermore, following the ideas of Chun [2], we develop fast $O(n^2)$ algorithms for factorizing certain block matrices that allow us to obtain alternative $O(n^2)$ algorithms to those mentioned above for solving linear equations and obtaining the inverse of the coefficient matrix.

A characteristic of all these new algorithms is that the coefficients in the recursions are all computed without invoking inner products, a fact that sets them apart from all Berlekamp–Massey-related algorithms. The absence of inner products makes parallel computation more feasible in the sense that with $O(n)$ processors, the computation time for the direct factorization methods can be reduced to $O(n)$, assuming unit time for each elementary computation; because of the inner products, the corresponding number for the Barlekamp–Massey-type algorithms is $O(n \log n)$.

TABLE 1.1
Classification of contributions.

Strong regularity required	Computes factors of inverse	Computes direct factors	Algorithm is for	Inner product required	Matrices
Yes	Trench (1965)		Inversion	Yes	H
Yes		Phillips (1971)	Factorization	Yes	H
Yes	Rissanen (1973)	Rissanen (1973)	Factorization	Yes	H
Yes		Lev-Ari and Kailath (1986)	Factorization	No	H, QH
Yes		Chun (1989)	Factorization and inversion	No	H, QH
No	Berlekamp (1968)		Linear equations	Yes	H
No		Rissanen (1971)	$O(n^3)$ factorization		H
No	Rissanen (1974)		Linear equations	Yes	H
No		Gragg (1974)	Factorization	Yes	H
No	Kung (1977)	Kung (1977)	Linear equations	Yes	H
No	Gragg and Lindquist (1983)			Yes	H
No	Heinig (1984)		Inversion and linear equations	Yes	H
No	Citron (1986)	Citron (1986)	Linear equations	No	H
No		Heinig and Jankowski (1989)	Linear equations	No	H, QH
No		Labahn et al. (1990)	Inversion and linear equations	No	H
No		Pal and Kailath (1994)	Factorization, inversion, and linear equations	No	H, QH, and SMQH

To close this introduction, we make the following observation: all the fast algorithms mentioned above (including those in this paper) either compute the direct triangular factors of the given matrix or the triangular factors of its inverse. Some of these algorithms require that the matrices be strongly regular, i.e., with all its leading principal minors nonzero, while the other algorithms work without this requirement. Although all these procedures are recursive, the need for the computation of an inner product at each step for some of these algorithms makes it difficult to parallelize them well (i.e., each recursive step requires a constant number of parallel steps so that the whole algorithm takes $O(n)$ parallel steps) and requires $O(n \log n)$ parallel steps.

Table 1.1 summarizes these classifications.

In §2, we explain the general (Jacobi) triangular factorization procedure for an arbitrary matrix. Then in §3 we show how the structure of Hankel and QH matrices can be exploited to obtain a fast recursive procedure for computing a modified triangular factorization of Hankel and QH matrices. It is shown that the diagonal blocks are lower triangular Hankel. Following Lev-Ari and Kailath [12], we introduce a "generating function" representation of Hermitian matrices and derive all our results in the form of certain polynomial recursions. In §4 we extend the results of §3 to include

certain SMQH matrices. In §5 we introduce the notions of *block generating functions* and *block quasi-Hankel* (BQH) matrices. It is shown that inverse of a QH matrix can be obtained by computing a block Schur complement of a certain associated BQH matrix. Next we discuss the so-called *admissibility conditions* and derive polynomial recursions for Schur complementation of BQH matrices with regular and arbitrary rank profiles. At this point we discuss the transmission line interpretation of these polynomial recursions and indicate that this procedure works under all circumstances including those where a Gohberg–Heinig-type formula does not hold to be good. We end this section with an example. In §6 we show how to adopt the procedure of §5 towards obtaining solutions to linear equations with a nonsingular QH coefficient matrix and an arbitrary right-hand side. Section 7 contains some concluding remarks. Appendix A provides a proof of (17) and Appendix B describes certain inertia rules of Iohvidov [7].

The related problem of factoring Toeplitz and quasi-Toeplitz (QT) matrices with arbitrary rank profile has been considered in a separate publication [13]. This is because of the fact that although an elegant unified derivation of fast triangularization procedures for structured matrices with a so-called generalized displacement structure has been obtained in the strongly regular case by Lev-Ari and Kailath [12], a unified treatment in the nonstrongly regular cases largely remains an open problem.

2. Triangular factorization of real symmetric matrices. Strongly regular real symmetric matrices $M \epsilon \mathbf{R}^{n \times n}$ can be factored as

$$M = LDL^T, \tag{1}$$

where L is a lower triangular matrix with unit diagonal elements and D is a nonsingular diagonal matrix and the superscript T denotes matrix transpose.

Let us denote the columns of matrix L by $\{\underline{l}_i, \quad i = 1, 2, \ldots, n\}$ and the diagonal elements of the matrix D by $\{d_i\}$. Then we can write

$$M = LDL^T = \sum_{i=1}^{n} d_i \underline{l}_i \underline{l}_i^T, \tag{2}$$

which suggests the following recursive computational procedure:

$$M_{i+1} = M_i - d_i \underline{l}_i \underline{l}_i^T; \qquad M_1 = M, \tag{3}$$

$$d_i = \underline{e}_i^T M_i \underline{e}_i \quad \text{and} \quad \underline{l}_i = M_i \underline{e}_i d_i^{-1}, \tag{4}$$

where $\underline{e}_i$ is the unit vector with nonzero entry at the ith position.

This is the celebrated Jacobi procedure (see [12]). Equations (2)–(4) show that the first $i-1$ rows and columns of M_i contain only zero entries, d_i is the (i,i)th element of M_i, and $d_i \underline{l}_i$ is the ith column of M_i, so that M_i has the form

$$M_i = \left[\begin{array}{c|c} O & O \\ - & - \\ O & P_i \end{array}\right].$$

P_i is called the Schur complement of the matrix M with regard to the leading $(i-1) \times (i-1)$ principal submatrix; note that P_{i+1} is the Schur complement of the $(1,1)$ entry of P_i, i.e., we compute one Schur complement at every step of the triangular factorization procedure (3)–(4).

Singularities. Given a real symmetric matrix M, it will be impossible to compute the Schur complement of the $(1,1)$ element if and only if the $(1,1)$ element is zero. The following are the possibilities:

(i) $M = 0$;

(ii) $M \neq 0$, but the first row and column of M are identically zero;

(iii) $M \neq 0$, the first row and column of M are also not identically zero, but $m_{11} = 0$.

In case (i) finding a triangular factorization is trivial. In case (ii) there is no need to find the Schur complement of m_{11}. We can simply choose the first column of L to be $\underline{e}_1$, and $d_1 = 0$; then resume the triangular factorization procedure. In case (iii) we must find the block Schur complement of the smallest nonsingular block available at the top-left corner of the matrix M. This will lead to a *modified* triangular factorization $M = LDL^T$, where $L \epsilon \mathbf{R}^{n\times n}$ is a lower triangular matrix with unit diagonal elements and $D \epsilon \mathbf{R}^{n\times n}$ is a matrix with scalar as well as block entries along the diagonal, while all the other entries are zero. Such a factorization always exists; however, it is nonunique.

3. Modified triangular factorization of Hankel and QH matrices. Since triangular factorization is nested, factorization of the leading minors does not depend on the actual size of the matrix. Hence we can consider without loss of generality that all matrices under consideration are in fact semi-infinite. Following [12], it is useful to associate M with a "generating" function

$$M(z,w) = [1 \; z \; z^2 \; \ldots \; \ldots] M [1 \; w \; w^2 \ldots \; \ldots]^*, \tag{5}$$

where * denotes the Hermitian transpose. A Hankel matrix of order n has the form $H_{n-1} = [h_{i+j}]_{i,j=0}^{n-1}$, where h_k are arbitrary $(k = 0, 1, 2, \ldots, 2n-2)$. We shall define the semi-infinite extension $H_{n-1,\infty}$ of a finite Hankel matrix H_{n-1} as a Hankel matrix with first row

$$[\, h_0 \; h_1 \; \ldots \; \ldots \; h_{2n-2} \; 0 \; 0 \; \ldots \; \ldots \,].$$

The generating function of this semi-infinite extension can be seen to be (see, e.g., [12])

$$H_{n-1,\infty}(z,w) = \frac{zh(z) - w^* h^*(w)}{z - w^*}, \tag{6}$$

where

$$h(z) = \sum_{k=0}^{2n-2} h_k z^k. \tag{7}$$

Next we shall define a matrix Q to be QH if its generating function $Q(z,w)$ has the form

$$Q(z,w) = \frac{a(z)b^*(w) - b(z)a^*(w)}{z - w^*}, \tag{8}$$

where $a(\cdot)$ and $b(\cdot)$ are real power series. The reason for this definition and for the name will appear soon (see (14) below). First, however, it is convenient to add a "normalizing" assumption that, unless otherwise specified,

$$a(0) = 0. \tag{9}$$

This is justified, except when $a(0) \neq 0$ and $b(0) = 0$ because *given any two real power serieses* $a(z)$ *and* $b(z)$, *we can construct two other real power serieses* $\alpha(z)$ *and* $\beta(z)$ *such that* $a(z)b^*(w) - b(z)a^*(w) = \alpha(z)\beta^*(w) - \beta(z)\alpha^*(w)$ *and* $\alpha(z) = 0$. The exception can be handled by simply interchanging $a(.)$ and $b(.)$. We will postpone the discussion of the so-called singular cases that include case $a(0) = 0 = b(0)$ until §3.1.

THEOREM 3.1. *The Schur complement of the* $(0,0)$ *element* h_0, *of* $H_{n-1,\infty}$ *is* QH.

Proof. If $h_0 = 0$, the Schur complement does not exist. When $h_0 \neq 0$, the Schur complement of h_0 in $H_{n-1,\infty}$ is obtained by subtracting from M the outer product of the first column and the first row, normalized by the inverse of h_0. The generating function $H^c_{n-1,\infty}(z,w)$ of this Schur complement is

$$(zw^*)H^c_{n-1,\infty}(z,w) = H_{n-1,\infty}(z,w) - H_{n-1,\infty}(z,0)[H_{n-1,\infty}(0,0)]^{-1}H_{n-1,\infty}(0,w).$$

Therefore using (7),

$$(zw^*)H^c_{n-1,\infty}(z,w) = H_{n-1,\infty}(z,w) - h(z)h^*(w)/h_0 \ .$$

After some algebra this reduces to

$$H^c_{n-1,\infty}(z,w) = \left(\frac{1}{h_0}\right)\left[\frac{h_1(z)h^*(w) - h(z)h_1^*(w)}{z - w^*}\right]; \qquad h_1(z) = \sum_{k=0}^{2n-3} h_{k+1}z^k, \tag{10}$$

which is QH. □

Given $Q(z,w)$ as in (8), where $a(\cdot)$ and $b(\cdot)$ are real polynomials, say

$$a(z) = \sum_{k=0}^{m} a_k z^k \quad \text{and} \quad b(z) = \sum_{k=0}^{q} b_k z^k,$$

let us define

$$g(z) = \frac{a(z)}{b(z)} = \sum_{k=0}^{\infty} g_k z^k \tag{11}$$

in some disc σ, centered at the origin. Also let

$$f(z) = z^{-1}(g(z) - g(0)) = \sum_{k=0}^{\infty} g_{k+1}z^k. \tag{12}$$

Then in the disc σ,

$$Q(z,w) = b(z)\frac{[zf(z) - w^*f^*(w)]}{(z - w^*)}b^*(w) \ . \tag{13}$$

The central term is the generating function of a Hankel matrix. We shall say that (13) is a congruous relationship because of the following matrix interpretation of (13):

$$Q = B_L F_H B_L^T, \tag{14}$$

where F_H is a semi-infinite Hankel matrix with first row $[\ g_1\ g_2\ g_3\ \ldots\ \ldots\ ]$ and B_L is a semi-infinite lower triangular Toeplitz matrix with first column

$$[\ b_0\ b_1\ \ldots\ \ldots\ b_q\ 0\ 0\ \ldots\ \ldots\ ]^T.$$

Note that since $b(0) \neq 0$, B_L in (14) is always full rank, so that the rank profile of Q is the same as that of F_H. The representation (14) is the reason for the name quasi-Hankel; the QH structure is preserved under Schur complementation.

THEOREM 3.2. *The Schur complement of the* $(0,0)$ *element,* g_1, *of* F_H *is* QH.

Proof. If $g_1 = 0$, then the Schur complement does not exist. So we assume $g_1 \neq 0$. By Theorem 3.1 (see (10)) the generating function $F_H^c(z,w)$ of the Schur complement of the $(0,0)$ element g_1 of F_H is

$$F_H^c(z,w) = \left(\frac{1}{g_1}\right)\left[\frac{f_1(z)f^*(w) - f(z)f_1^*(w)}{z - w^*}\right]; \qquad f_1(z) = \sum_{k=0}^{\infty} g_{k+2}z^k \quad \text{and} \quad g_1 \neq 0.$$

The congruence (13) then shows that the generating function $Q^c(z,w)$ of the Schur complement of the $(0,0)$ element of Q is

$$\begin{aligned} Q^c(z,w) &= \left(\frac{1}{g_1}\right) b(z)\left[\frac{f_1(z)f^*(w) - f(z)f_1^*(w)}{z - w^*}\right] b^*(w) \\ &= \left(\frac{1}{g_1}\right)\left[\frac{f_2(z)f^*(w) - f(z)f_2^*(w)}{z - w^*}\right]; \qquad f_2(z) = b(z)f_1(z), \end{aligned}$$

which is QH.

3.1. Singularities of the QH family of matrices. Given a QH matrix Q with generating function $Q(z,w)$ as in (8), it will be impossible to compute the Schur complement of the $(0,0)$ element if and only if the $(0,0)$ element is zero or, equivalently, $Q(0,0) = 0$. The following are the possibilities (see §2).

(i) $Q = 0$, i.e., $Q(z,w) = 0$, which will happen if and only if $a(z)/b(z) = c$, a constant.

(ii) $Q \neq 0$, but the first row and column of Q are identically zero or, in other words, $Q(z,w) \neq 0$, but $Q(z,0) = Q(0,w) = 0$. Using our assumption (9), this is *equivalent* to having $a(0) = 0 = b(0)$.

(iii) $Q \neq 0$, the first row and column of Q are also not identically zero, but $q_{00} = 0$ or, in other words, $Q(z,w) \neq 0$, $Q(z,0) \neq 0$, $Q(0,w) \neq 0$, but $Q(0,0) = 0$. Now recalling our assumption (9) that $a(0) = 0$, we must have $b(0) \neq 0$ (if not we would be in case (ii)). Then $Q(0,0) = a'(0)b^*(0) = 0 \Rightarrow a'(0) = 0$ and $Q(z,0) = a(z)b^*(0)/z \neq 0 \Rightarrow a(z) \neq 0$. Thus $a(0) = a'(0) = 0$, but $a(z) \neq 0$. This implies there exists a positive integer $t > 1$, such that $\lim_{z\to 0} z^{-t}a(z) \neq 0$. Now it is clear from (11) that *if* $b(0) \neq 0$, *then* $z^{-j}g(z)|_{z=0} = 0$ *if and only if* $z^{-j}a(z)|_{z=0} = 0$, $0 \leq j \leq t-1$ *for* $t \leq m$. *This leads to the observations that* $\{a_i = 0\}_{i=1}^{t-1} \Leftrightarrow \{g_i = 0\}_{i=1}^{t-1}$ *and that* the first $t-1$ leading principal minors of F_H are zero if and only if $g_j = 0$, $1 \leq j \leq t-1$.

We can summarize the above discussion as follows.

LEMMA 3.1. *The first nonsingular leading principal submatrix of* F_H *is of dimension* t *if and only if* t *is the smallest positive integer such that* $\lim_{z\to 0} z^{-t}a(z) \neq 0$, *provided* $b(0) \neq 0$.

Because $g_j = 0, \quad 0 \le j \le t-1$, this submatrix, say $\tilde{F}_{H(t)}$, must have the form

$$\tilde{F}_{H(t)}\tilde{I}_t = \begin{bmatrix} g_t & 0 & 0 & \dots & 0 \\ g_{t+1} & g_t & 0 & & \vdots \\ \vdots & & & & 0 \\ \vdots & & & & 0 \\ g_{2t-1} & \dots & \dots & g_{t+1} & g_t \end{bmatrix}, \tag{15}$$

where $\tilde{I}_t$ is a $t \times t$ antidiagonal unit matrix. Then $\tilde{I}_t[\tilde{F}_{H(t)}]^{-1}$ will also be lower triangular Toeplitz, say

$$\tilde{I}_t[\tilde{F}_{H(t)}]^{-1} = \begin{bmatrix} p_1 & 0 & \dots & \dots & 0 \\ p_2 & p_1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ p_{t-1} & \ddots & \ddots & & 0 \\ p_t & p_{t-1} & \dots & p_2 & p_1 \end{bmatrix} = \mathcal{P}\tilde{I}_t, \text{ say.} \tag{16}$$

Furthermore, some calculation shows (see Appendix A) that

$$\begin{bmatrix} a_t & 0 & \dots & 0 \\ a_{t+1} & a_t & & \vdots \\ \vdots & a_{t+1} & & 0 \\ a_{2t-1} & \dots & a_{t+1} & a_t \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_t \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{t-1} \end{bmatrix}. \tag{17}$$

Now we can analyze the block Schur complement of $F_{H(t)}$ and derive a recursive procedure for computing the block factorization in t scalar steps.

The Schur complement $\tilde{F}^{c}_{H(t)}$ with respect to $\tilde{F}_{H(t)}$ as in (15) is

$$\tilde{F}^{c}_{H(t)} = F_H - \mathcal{G}_t\mathcal{P}\mathcal{G}_t^T, \tag{18}$$

where

$$\mathcal{G}_t = \begin{bmatrix} g_t & 0 & 0 & \dots & 0 \\ g_{t+1} & g_t & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & g_{t+1} & g_t & 0 \\ \vdots & \ddots & \ddots & \ddots & g_t \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \quad \text{and} \quad \mathcal{P} = \begin{bmatrix} 0 & \dots & 0 & p_1 \\ \vdots & & \ddots & p_2 \\ 0 & \ddots & \ddots & \vdots \\ p_1 & p_2 & \dots & p_t \end{bmatrix}_{t\times t}.$$

Equation (18) corresponds to a block factorization step. The block diagonal entry of dimension t is the Hankel matrix $\mathcal{P}$ and the corresponding t columns of the lower triangular factor are down-shifted versions of the same infinite vector $[g_t \;\; g_{t+1} \;\; \dots \;\; \dots]^T$.

The generating function of this block factor is

$$\begin{aligned}
&[1\ z\ z^2\ \ldots]\mathcal{G}_t\mathcal{P}\mathcal{G}_t^T[1\ w\ w^2\ \ldots]^* \\
&\quad = g_t(z)[1\ z\ z^2\ \ldots\ z^{t-1}]\mathcal{P}[1\ w\ w^2\ \ldots\ w^{t-1}]^* g_t^*(w) \\
&\quad = g_t(z)\left[\sum_{k=0}^{t-1} p_{t-k}\left(\frac{z^{k+1}-(w^*)^{k+1}}{z-w^*}\right)(zw^*)^{t-1-k}\right] g_t^*(w),
\end{aligned}$$

where

$$g_t(z) = \sum_{j=0}^{\infty} g_{j+t}z^j = z^{-t}g(z).$$

Then since $Q^c(z,w) = b(z)\tilde{F}^c_{H(t)}(z,w)b^*(w)$, we have the following result.

LEMMA 3.2. Generating function of the Schur complement. *If $b(0) \neq 0$ and the first nonsingular principal submatrix of Q is of dimension t, then the block Schur complement Q_t^c of this first $t\times t$ nonsingular leading principal submatrix has generating function*

(19)

$$(zw^*)^t Q_t^c(z,w) = Q(z,w) - a_t(z)a_t^*(w)\left[\sum_{k=0}^{t-1} p_{t-k}\left(\frac{z^{k+1}-(w^*)^{k+1}}{z-w^*}\right)(zw^*)^{t-1-k}\right],$$

where $a_t(z) = z^{-t}a(z)$ and $\{p_j\}_{j=1}^t$ are given by (17).

Although computation of $\{p_j\}_{j=1}^t$ via (17) is sufficient to find the required block factors in (18), it is not yet clear how to express the generating function $Q_t^c(z,w)$ of the QH block Schur complement in the standard form (8). Some further exploration will show how to do this.

We can try to rewrite (19) to show one step at a time:

$$\begin{aligned}
(zw^*)^t Q_t^c(z,w) &= Q(z,w) - a_t(z)a_t^*(w)\left[\sum_{k=0}^{t-1} p_{t-k}\left(\frac{z^{k+1}-(w^*)^{k+1}}{z-w^*}\right)(zw^*)^{t-1-k}\right] \\
&= \frac{z^t a_t(z)[b^*(w) - p_1 a_t^*(w)] - (w^*)^t a_t^*(w)[b(z) - p_1 a_t(z)]}{z-w^*} \\
&\quad -a_t(z)a_t^*(z)\left[\sum_{k=0}^{t-2} p_{t-k}\left(\frac{z^{k+1}-(w^*)^{k+1}}{z-w^*}\right)(zw^*)^{t-1-k}\right].
\end{aligned}$$

Now by (17), $b(z) - p_1 a_t(z)$ is divisible by z. So if we define

$$z^{t-1}a_t(z) = a_1(z), \qquad zb_1(z) = b(z) - p_1 a_t(z),$$

and

$$Q_1^p(z,w) = \frac{a_1(z)b_1^*(w) - b_1(z)a_1^*(w)}{z-w^*},$$

then

$$(zw^*)^{t-1}Q_t^c(z,w) = Q_1^p(z,w) - a_t(z)a_t^*(w)\left[\sum_{k=0}^{t-2} p_{t-k}\left(\frac{z^{k+1}-(w^*)^{k+1}}{z-w^*}\right)(zw^*)^{t-2-k}\right].$$

This suggests the following result.

LEMMA 3.3. Recursive form for $Q_t^c(z,w)$. *It holds that*

$$Q_t^c(z,w) = \frac{a_t(z)b_t^*(w) - b_t(z)a_t^*(w)}{z - w^*}, \tag{20}$$

where $b_t(z)$ *is defined by the recursion,*

$$zb_{j+1}(z) = b_j(z) - p_{j+1}a_t(z); \quad p_{j+1} = \frac{b_j(0)}{a_t(0)}, \quad b_0(z) = b(z). \tag{21}$$

Proof. The lemma is proved by induction.

Define

$$Q_j^p(z,w) = \frac{a_j(z)b_j^*(w) - a_j^*(w)b_j(z)}{z - w^*},$$

where, $a_j(z) = z^{t-j}a_t(z)$ and $\{b_j(z)\}_{j=1}^t$ is as defined above. Then,

$$\begin{aligned} &Q_j^p(z,w) - a_t(z)a_t^*(w)p_{j+1}\left(\frac{z^{t-j} - (w^*)^{t-j}}{z - w^*}\right) \\ &= \frac{a_j(z)[b_j^*(w) - p_{j+1}a_t^*(w)] - a_j^*(w)[b_j(z) - p_{j+1}a_t(z)]}{z - w^*} \\ &= (zw^*)\left[\frac{a_{j+1}(z)b_{j+1}^*(w) - a_{j+1}^*(w)b_{j+1}(z)}{z - w^*}\right] = (zw^*)Q_{j+1}^p(z,w). \end{aligned}$$

Therefore

$$\begin{aligned} &Q(z,w) - a_t(z)a_t^*(w)\left[\sum_{k=0}^{t-1} p_{t-k}\left(\frac{z^{k+1} - (w^*)^{k+1}}{z - w^*}\right)(zw^*)^{t-1-k}\right] \\ &= (zw^*)^t Q_t^p(z,w) = (zw^*)^t Q_t^c(z,w). \end{aligned}$$

Hence,

$$Q_t^c(z,w) = Q_t^p(z,w) = \frac{a_t(z)b_t^*(w) - a_t^*(w)b_t(z)}{z - w^*}.$$

The recursions (20) and (21) provide an efficient means of computing the block factorization step in (19) via a sequence of t scalar recursions. It is clear that the $\{Q_j^p(z,w)\}_{j=0}^{t-1}$ do not represent any Schur complement; only the final member of this sequence, viz., $Q_t^p(z,w)$, represents the desired block Schur complement and allows us to continue with the factorization process.

Now we are ready to describe a general triangular factorization algorithm by putting together the steps described above.

3.2. Algorithm for triangular factorization of QH matrices. Let us first express the general factorization formulas (3) and (4) in generating function terms,

$$\bar{M}_{j+1}(z,w) = \bar{M}_j(z,w) - \bar{M}_j(z,0)\bar{M}_j^{-1}(0,0)\bar{M}_j(0,w), \tag{22}$$

$$d_j = \bar{M}_j(0,0), \bar{l}_j(z) = \bar{M}_j(z,0)d_j^{-1} \quad \text{when} \quad d_j \neq 0, \tag{23}$$

where

$$(zw^*)^{-1}\bar{M}_j(z,w) = M_j(z,w) = [1\ z\ z^2\ \dots\ \dots\ ]\ M_j\ [1\ w\ w^2\ \dots\ \dots\ ]^*$$

and

$$z^j\bar{l}_j(z) = l_j(z) = [1\ z\ z^2\ \dots\ \dots\ ]\ \underline{l}_j.$$

The following cases represent all the possibilities that may be encountered at any step j.

Case 1. (a) $a_j(0) \neq 0$, $b_j(0) \neq 0$. In this case, we generate two other polynomials $a_{j+1}(0)$ and $b_{j+1}(0)$, such that $a_{j+1}(0) = 0$ and $b_{j+1}(0) \neq 0$, so that $Q_j(z,w) = Q_{j+1}(z,w)$:

$$\begin{aligned} a_{j+1}(z) &= a_j(z) - \zeta_j b_j(z)\ ;\ \zeta_j = a_j(0)/b_j(0), \\ b_{j+1}(z) &= b_j(z). \end{aligned}$$

(b) $a_j(0) \neq 0$, $b_j(0) = 0$. This is just an exchange step. We simply assign

$$a_{j+1}(z) = b_j(z), \qquad b_{j+1}(z) = a_j(z)$$

so that $Q_j(z,w) = -Q_{j+1}(z,w)$.

Case 2. (a) $a_j(0) = 0$, $b_j(0) \neq 0$, $\lim_{z\to 0} z^{-1}a_j(z) \neq 0$. This is a strongly regular step (the Schur complement of the (1,1) entry can be computed) for which the recursion is

$$\begin{aligned} za_{j+1}(z) &= a_j(z), \\ zb_{j+1}(z) &= b_j(z) - \zeta_j z^{-1}a_j(z);\ \zeta_j = \lim_{z\to 0} \frac{b_j(z)}{z^{-1}a_j(z)}. \end{aligned}$$

This implies that

$$(zw^*)Q_{j+1}(z,w) = Q_j(z,w) - \zeta_j a_{j+1}(z)a_{j+1}^*(w),$$

from which we can identify

$$d_{(j-\delta_j-\gamma_j)} = (-1)^{\gamma_j}Q_j(0,0) = (-1)^{\gamma_j}\zeta_j|a_{j+1}(0)|^2,$$

and

$$l_{(j-\delta_j-\gamma_j)}(z) = z^{(j-\delta_j-\gamma_j)}a_{j+1}(z)/a_{j+1}(0),$$

where δ_j and γ_j denote the number of times Cases 1(a) and 1(b) have been encountered before step j.

Case 3. $a_j(0) = 0$, $b_j(0) \neq 0$ and there exists a positive integer $t > 1$, such that $z^{-t}a_j(z)$ is a polynomial in z and $\lim_{z\to 0} z^{-t}a_j(z) \neq 0$. The Schur complement of the (1,1) entry cannot be computed in this case; however, a block Schur complement can be computed via repeated (t times) application of (21). So the recursion is

$$\begin{aligned} za_{j+1}(z) &= a_j(z), \\ zb_{j+1}(z) &= b_j(z) - \zeta_j z^{-t}a_j(z);\ \zeta_j = \lim_{z\to 0} \frac{b_j(z)}{z^{-t}a_j(z)}. \end{aligned}$$

This implies that

$$(zw^*)^t Q^p_{j+t}(z,w) = Q^p_j(z,w) - a_{j+t}(z)a^*_{j+t}(w)\left[\sum_{k=0}^{t-1}\zeta_{j+t+k}\left(\frac{z^{k+1}-(w^*)^{k+1}}{z-w^*}\right)(zw^*)^{t-1-k}\right].$$

So, in Case 3, the block diagonal is the $t \times t$ matrix $(-1)^{\gamma_j} D_{(j-\delta_j-\gamma_j)}$, where

$$D_{(j-\delta_j-\gamma_j)} = \begin{bmatrix} 0 & 0 & \dots & 0 & \zeta_j \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \zeta_j & \dots & \dots & \dots & \zeta_{j+t-1} \end{bmatrix}. \tag{24}$$

The t columns contributing to the lower triangular factor L form the $n \times t$ Toeplitz matrix whose first column is $[\underbrace{0,\dots,0}_{j-\delta_j-\gamma_j}, a_{j+t,0}, a_{j+t,1}, \dots, \dots, a_{j+t,m}, 0, 0, \dots, 0]^T_{n\times 1}$ where

$$a_{j+t}(z) = \sum_{k=0}^{m} a_{j+t,k} z^k.$$

Case 4. $a_j(0) = 0$ and $b_j(0) = 0$. In this case both the first column and row of Q_j must be zeros. So the recursion for the polynomials is

$$za_{j+1}(z) = a_j(z),$$
$$zb_{j+1}(z) = b_j(z).$$

This implies that

$$(zw^*)Q_{j+1}(z,w) = Q_j(z,w).$$

Therefore

$$d_{(j-\delta_j-\gamma_j)} = 0 \quad \text{and} \quad l_{(j-\delta_j-\gamma_j)}(z) = z^{(j-\delta_j-\gamma_j)}.$$

We have just discussed the relationship of the polynomial recursions and the factors of triangular factorization in all of the four cases. It is important to note that at any step one only needs to compute a linear update of the polynomials $a_j(\cdot)$ and $b_j(\cdot)$. This is true irrespective of the presence of a singularity. Hence the *modified triangular factorization* of QH matrices can be computed in $O(n^2)$ computations using the recursions of this subsection.

Example. Let

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}.$$

TABLE 3.1
Recursions for triangular factorization.

Step j	$a_{j+1}(z)$ $b_{j+1}(z)$	ζ_j	t_j	δ_j	γ_j	Comment
0	$1+z+z^2+2z^3+2z^4+2z^5+2z^6+2z^7+2z^8$ $-[1+z+2z^2+2z^3+2z^4+2z^5+2z^6+2z^7]$	1	1	0	0	Strong regularity
1	$-z^2+2z^8$ $-[1+z+2z^2+2z^3+2z^4+2z^5+2z^6+2z^7]$	-1	0	0	0	Transformation step
2	$-z+2z^7$ $-[1+z+2z^2+2z^3+2z^4+4z^5+2z^6]$	1	2	1	0	Singularity Case 3
3	$-1+2z^6$ $-2[1+z+z^2+z^3+2z^4+2z^5]$	1	1	1	0	Strong regularity
4	$z+z^2+z^3+2z^4+2z^5+2z^6$ $-2[1+z+z^2+z^3+2z^4+2z^5]$	$\frac{1}{2}$	0	1	0	Transformation step
5	$1+z+z^2+2z^3+2z^4+2z^5$ $2z^2$	-2	1	2	0	Strong regularity
6	$2z^2$ $1+z+z^2+2z^3+2z^4+2z^5$	0	0	2	0	Exchange step
7	$2z$ $1+z+2z^2+2z^3+2z^4$	$\frac{1}{2}$	2	2	1	Singularity case 3
8	2 $1+2z+2z^2+2z^3$	$\frac{1}{2}$	1	2	1	Strong regularity
9	$-[4z+4z^2+4z^3]$ $1+2z+2z^2+2z^3$	2	0	2	1	Transformation step
10	$-[4+4z+4z^2]$ $1+z+2z^2$	$-\frac{1}{4}$	1	3	1	Strong regularity
11	$4z^2$ $1+z+2z^2$	-4	0	3	1	Transformation step
12	$4z$ $1+2z$	$\frac{1}{4}$	2	4	1	Singularity case 3
13	4 2	$\frac{1}{4}$	1	4	1	Strong regularity
14	0 0	2	0	4	1	Transformation step

Now defining a semi-infinite extension $H_{4,\infty}$ as in (10), we get, $H_{4,\infty}(z,w) = [zh(z)-w^*h^*(w)]/[z-w^*]$, where $h(z) = 1+z+z^2+2z^3+2z^4+2z^5+2z^6+2z^7+2z^8$. Using the procedure described earlier, we get the polynomials $\{a_j(z),\ b_j(z)\}_{j=0}^{15}$, starting with $a_0(z) = zh(z)$ and $b_0(z) = 1$. The results are summarized in Table 3.1.

This yields the following factorization,

$$H_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

4. SMQH matrices. A matrix N is said to be an SMQH matrix if its generating function admits the representation

$$N(z,w) = \frac{p(z)q^*(w) - q(-z)p^*(-w)}{z+w^*}, \tag{25}$$

where $p(z)$ and $q(z)$ are real polynomials. An important special case is when $p(z) = q(z)$, which arises in studying the problem of root distribution of the polynomial $p(z)$ with respect to the imaginary axis (see Krein and Naimark [8] and Lev-Ari, Bistritz, and Kailath [11]). Then

$$N(z,w) = \frac{p(z)p^*(w) - p^*(-w)p(-z)}{z + w^*} \tag{26}$$

and we shall call it the imaginary axis Bezoutian. Note that this is not QH, as defined in (8). However we can obtain a related QH Bezoutian as follows.

Let $m(z)$ and $n(z)$, be the even and the odd parts of $p(z)$; in terms of which we can write

$$N(z,w) = \frac{2[m(z)n^*(w) + n(z)m^*(w)]}{z + w^*} = 2I(z,w), \text{ say.} \tag{27}$$

Then note that

$$I(-z,w) = I(z,-w) = \frac{2[n(z)m^*(w) - m(z)n^*(w)]}{z - w^*} \tag{28}$$

represents a QH matrix.

Now we can apply the algorithm of the last section; certain simplifications ensue because of the special structure of $I(-z,w)$ (alternate entries of the corresponding matrix are zero). Assign $a(z) = n(z)$ and $b(z) = m(z)$. Then $a_0(z) = n(z)$ and $b_0(z) = m(z)$. Since $a_0(0) = 0$, the transformation step is not necessary. Now if $b_0(0) \neq 0$ and $\lim_{z\to 0} z^{-1}a_0(z) \neq 0$, then Case 2 yields

$$za_1(z) = a_0(z) = n(z)$$

and

$$zb_1(z) = b_0(z) - \zeta_0 z^{-1}a_0(z) = m(z) - \zeta_0 z^{-1}n(z); \qquad \zeta_0 = \lim_{z\to 0} zm(z)/n(z).$$

In this case, $a_1(z)$ is an even polynomial and $b_1(z)$ becomes an odd polynomial. However, if $b_0(0) \neq 0$ but $\lim_{z\to 0} z^{-1}a_0(z) = 0$, then Case 3 applies provided $a_0(z) \neq 0$. If then t is the smallest positive integer such that $\lim_{z\to 0} z^{-t}a_0(z) \neq 0$, then

$$za_1(z) = a_0(z) = n(z)$$

and

$$zb_1(z) = b_0 - \zeta_0 z^{-t}a_0(z) = m(z) - \zeta_0 z^{-t}n(z); \qquad \zeta_0 = \lim_{z\to 0} z^t m(z)/n(z).$$

Once again, $a_1(z)$ is an even polynomial whereas $b_1(z)$ is an odd one. But if $b_0(0) = 0$, then Case 4 applies and we get

$$za_1(z) = a_0(z) = n(z)$$

and

$$zb_1(z) = b_0(z) = m(z).$$

Even in this case, $a_1(z)$ is an even polynomial and $b_1(z)$ is an odd one. Thus,

$$I_1(z,-w) = \frac{a_1(z)b_1^*(w) - a_1^*(w)b_1(z)}{z - w^*},$$

where $a_1(z)$ is an even polynomial and $b_1(z)$ is an odd polynomial. The structure of $I(z,-w)$ makes it clear that not only is it QH, but QH with a certain structure, since

at every step of recursion one of the polynomials must be even while the other one must be odd. This leads to the elimination of any possible encounter with the Case 1(a). Since we are dealing with even and odd polynomials, we shall use $\{m_j(z)\}$ and $\{n_j(z)\}$ as variables of recursion. Define

$$I_j(z,-w) = \frac{n_j(z)m_j^*(w) - m_j(z)n_j^*(w)}{z - w^*}. \tag{29}$$

Then the recursions for $\{m_j(\cdot)\}$ and $\{n_j(\cdot)\}$ are as follows: Assign $m_0(z) = m(z)$ and $n_0(z) = n(z)$.

Case (i). If $m_j(z) \neq 0$ and $\lim_{z\to 0} z^{-1}n_j(z) \neq 0$, then

$$zn_{j+1}(z) = n_j(z),$$

$$zn_{j+1}(z) = m_j(z) - \zeta_j z^{-1}n_j(z); \qquad \zeta_j = \lim_{z\to 0}\frac{zm_j(z)}{n_j(z)}.$$

Also,

$$I_j(z,-w) = \zeta_j M_{j+1}(z)m_{j+1}^*(w) - (zw^*)I_{j+1}(z,-w)$$

or

$$I_j(z,w) = \zeta_j m_{j+1}(z)m_{j+1}^* + zw^* I_{j+1}(z,w).$$

Then the diagonal factor and the generating function of the jth column of the triangular factor are

$$d_j = I_j(0,0) = \zeta_j|m_{j+1}(0)|^2$$

and

$$l_j = m_{j+1}(z)/m_{j+1}(0).$$

Case (ii). If $m_j(0) \neq 0, \lim_{z\to 0} z^{-1}n_j(z) = 0$ while there exists a positive integer t such that $z^{-t}n_j(z)$ is a polynomial and $\lim_{z\to 0} z^{-t}n_j(z) \neq 0$, then t must be odd. Let $t = 2\psi + 1$, then

$$zm_{j+1}(z) = n_j(z)$$

and

$$zn_{j+1}(z) = m_j(z) - \zeta_j z^{-t}n_j(z); \qquad \zeta_j = \lim_{z\to 0}\frac{z^t m_j(z)}{n_j(z)}.$$

Case (iii). If $m_j(0) = 0$, then

$$zm_{j+1}(z) = n_j(z), \qquad zn_{j+1}(z) = m_j(z).$$

A close examination reveals that an occurrence of Case (iii) without a prior occurrence of Case (ii) can happen only if Case (iii) occurs before any other case. Otherwise, if an instance of Case (ii) is encountered then 2ψ consecutive singular leading principal minors of the given matrix must be followed by a nonsingular leading principal minor. In that situation, occurrence of the Cases (ii) and (iii) will alternate ψ times each and then an occurrence of Case (i) must be encountered. All this reflects itself in the following formula:

$$(zw^*)^{2\psi+1}I_{j+2\psi+1}(z,w) =$$

$$- m_{j+2\psi+1}(z)m_{j+2\psi+1}^*(w)\left[\sum_{k=0}^{\psi}\zeta_{j+2k}^{(zw^*)^{2k}}\left[\frac{z^{2(\psi-k)+1} + (w^*)^{2(\psi-k)+1}}{\Delta(z+w^*)}\right]\right] + I_j(z,w).$$

So the block diagonal entry

$$D_j = \begin{bmatrix} 0 & 0 & \cdots & \cdots & \cdots & 0 & \zeta_j \\ 0 & \ddots & \ddots & \ddots & 0 & -\zeta_j & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \zeta_{j+2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \zeta_j & \ddots & \ddots & \ddots & \vdots \\ 0 & -\zeta_j & \ddots & \ddots & \ddots & \ddots & 0 \\ \zeta_j & 0 & \zeta_{j+2} & \cdots & \cdots & 0 & \zeta_{j+2\psi} \end{bmatrix}_{(2\psi+1)\times(2\psi+1)} \tag{30}$$

and the corresponding $2\psi+1$ columns of the triangular factor L must be the $n\times(2\psi+1)$ Toeplitz matrix whose first column is

$$[\underbrace{0,\ldots,0}_{j}, m_{j+2\psi+1,0}, 0, m_{j+2\psi+2,0}, \ldots, m_{j+2\psi+1,2q}, 0, 0, \ldots, 0]^T_{n\times 1},$$

where

$$m_{j+2\psi+1}(z) = \sum_{k=0}^{q} m_{j+2\psi+1,\,2k} z^{2k}.$$

However, an independent occurrence of Case (iii) will reflect itself as

$$I_{j+1}(z,w) = zw^* I_j(z,w).$$

Then

$$d_j = 0 \quad \text{and} \quad l_j(z) = z^j.$$

We have just connected the recursions in $\{m_j(z)\}$ and $\{n_j(z)\}$ with the factors of triangular factorization. Extensions of the procedure just described to find a "modified triangular factorization" of an arbitrary SMQH matrix are straightforward. Extensions for factoring Bezout matrices (associated with the imaginary axis Bezoutians) derived from a complex polynomial are also straightforward and will be presented elsewhere (see [19]).

5. Inversion of Hankel and QH matrices: Block generating function approach. Consider a nonsingular Hankel matrix of order n $H_{n-1} = [h_{i+j}]_{i,j=0}^{n-1}$, where h_k are arbitrary ($k = 0,1,2,\ldots,2n-2$). We shall define the $2n \times 2n$ Hankel block matrix

$$H_b = \left[\begin{array}{c|c} H_{n-1} & I \\ \hline I & O \end{array}\right]_{2n\times 2n}. \tag{31}$$

H_{n-1} is the first $n \times n$ principal submatrix of H_b and the corresponding block Schur complement is $-H_{n-1}^{-1}$. We define the following semi-infinite matrix

$$H_{b,\infty} = \left[\begin{array}{c|c} H_{n-1,\infty} & I_\infty \\ \hline I_\infty & O_\infty \end{array}\right], \tag{32}$$

where $H_{n-1,\infty}$ is the semi-infinite extension of H_{n-1}, I_∞ is a semi-infinite identity matrix, and O_∞ is a semi-infinite zero matrix. It is useful to associate a semi-infinite block structured matrix

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ M_{21} & M_{22} \end{array}\right]$$

(with structured semi-infinite blocks M_{ij}, $1 \le i,j \le 2$) with a block "generating" function

$$M(z,w) = \left[\begin{array}{cccc|cccc} 1 & z & z^2 & \dots & O & \dots & \dots & O \\ O & \dots & \dots & O & 1 & z^{-1} & z^{-2} & \dots \end{array}\right] \left[\begin{array}{c|c} M_{11} & M_{12} \\ M_{21} & M_{22} \end{array}\right] \cdot \left[\begin{array}{cccc|cccc} 1 & w & w^2 & \dots & O & \dots & \dots & O \\ O & \dots & \dots & O & 1 & w^{-1} & w^{-2} & \dots \end{array}\right]^* . \tag{33}$$

Then the block generating function (which is obtained by replacing each of the semi-infinite Hankel blocks by their generating functions)

$$H_{b,\infty}(z,w) = \left[\begin{array}{c|c} H_{n-1,\infty}(z,w) & \frac{-w^*}{z-w^*} \\ \frac{z}{z-w^*} & O \end{array}\right]_{2\times 2}, \tag{34}$$

where

$$H_{n-1,\infty}(z,w) = \frac{zh(z) - w^*h^*(w)}{z-w^*}, \tag{35}$$

such that

$$h(z) = \sum_{k=0}^{2n-2} h_k z^k.$$

For now we restrict ourselves to real symmetric matrices only, so that $h(z) = h^*(z^*)$. For simplicity assume that $h_0 \neq 0$ for now. Then note that

$$H_{b,\infty}(z,w) = \frac{\left[\begin{array}{c|c} zh(z) & 1 \\ z & 0 \end{array}\right] \left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right] \left[\begin{array}{c|c} wh(w) & 1 \\ w & 0 \end{array}\right]^*}{z-w^*}. \tag{36a}$$

Clearly, then

$$H_{b,\infty}(z,w) = \frac{G_b(z)JG_b^*(w)}{z-w^*}, \tag{36b}$$

where

$$J = \left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right] \quad \text{and} \quad G_b(z) = \left[\begin{array}{c|c} zh(z) & 1 \\ z & O \end{array}\right].$$

Next we shall define a matrix Q_b to be BQH if its block generating function $Q_b(z,w)$ can be represented in the form

$$Q_b(z,w) = \frac{1}{z-w^*} \left[\begin{array}{c|c} a(z) & b(z) \\ c(z) & d(z) \end{array}\right] \left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right] \left[\begin{array}{c|c} a(w) & b(w) \\ c(w) & d(w) \end{array}\right]^*, \tag{37}$$

where $a(z)$, $b(z)$, $c(z)$, and $d(z)$ are all polynomials (functions) in z and z^{-1}.

THEOREM 5.1. *The Schur complement of the* 1×1 *leading principal submatrix of* $H_{b,\infty}$ *is* BQH.

Proof. If $h_1 = 0$, then the Schur complement does not exist. So we assume $h_1 = 1$. The block generating function $H^c_{b,\infty}(z,w)$ of this Schur complement is

$$\begin{bmatrix} z & | & O \\ O & | & 1 \end{bmatrix} H^c_{b,\infty}(z,w) \begin{bmatrix} w^* & | & O \\ O & | & 1 \end{bmatrix} = H_{b,\infty}(z,w) - \begin{bmatrix} H_{n-1,\infty}(z,0) \\ 1 \end{bmatrix} [H_{n-1,\infty}(0,0)]^{-1} \begin{bmatrix} H_{n-1,\infty}(0,w) & | & 1 \end{bmatrix},$$

which implies that

(38)
$$H^c_{b,\infty}(z,w) = \frac{1}{z-w^*} \begin{bmatrix} h(z) & | & h_1(z) \\ z & | & -1 \end{bmatrix} \begin{bmatrix} O & | & 1 \\ -1 & | & O \end{bmatrix} \begin{bmatrix} h(w) & | & h_1(w) \\ w & | & -1 \end{bmatrix}^*,$$

where we have defined

$$h_1(z) = \sum_{k=0}^{2n-3} h_{k+1} z^k. \qquad \square$$

Since $H^c_{b,\infty}$ is BQH, a question naturally arises about the structure of the Schur complement of a BQH matrix. This is answered in the following theorem.

THEOREM 5.2. *The Schur complement of the* 1×1 *leading principal submatrix of a* BQH *matrix is also* BQH.

Proof. Define the partition

$$Q_b(z,w) = \begin{bmatrix} q_{b,11}(z,w) & | & q_{b,12}(z,w) \\ q_{b,21}(z,w) & | & q_{b,22}(z,w) \end{bmatrix}.$$

The generating function $Q^c_b(z,w)$ of the Schur complement is given by

(39)
$$\begin{bmatrix} z & | & O \\ O & | & 1 \end{bmatrix} Q^c_b(z,w) \begin{bmatrix} w^* & | & O \\ O & | & 1 \end{bmatrix} = Q_b(z,w) - \begin{bmatrix} q_{b,11}(z,0) \\ q_{b,21}(z,0) \end{bmatrix} q^{-1}_{b,11}(0,0) \begin{bmatrix} q_{b,11}(0,w) & | & q_{b,12}(0,w) \end{bmatrix},$$

where $Q_b(z,w)$ in (37) is the block generating function of BQH matrix Q_b. At this point, we make the same normalizing assumption $a(0) = 0$ as in §3. Then

$$\begin{bmatrix} q_{b,11}(z,0) \\ q_{b,21}(z,0) \end{bmatrix} = \begin{bmatrix} z^{-1}a(z)b^*(0) \\ z^{-1}c(z)b^*(0) \end{bmatrix} = b^*(0) \begin{bmatrix} a_1(z) \\ c_1(z) \end{bmatrix}, \text{ say.}$$

It is clear that $a_1(z)$ and $c_1(z)$ are polynomials in z and z^{-1}, respectively. Now if $Q_b(0,0) \neq 0$, then after some algebra we get

(40)
$$Q^c_b(z,w) = \frac{|b(0)|^2 \begin{bmatrix} a_1(z)b_1^*(w) - b_1(z)a_1^*(w) & | & -(b_1(z)w^*c_1^*(w) - a_1(z)d_1^*(w)) \\ b_1^*(w)zc_1(z) - a_1^*(w)d_1(z) & | & zc_1(z)d_1^*(w) - w^*c_1^*(w)d_1(z) \end{bmatrix}}{z-w^*}$$
$$= \frac{|b(0)|^2 \begin{bmatrix} a_1(z) & | & b_1(z) \\ zc_1(z) & | & d_1(z) \end{bmatrix} \begin{bmatrix} O & | & 1 \\ -1 & | & O \end{bmatrix} \begin{bmatrix} a_1(w) & | & b_1(w) \\ wc_1(w) & | & d_1(w) \end{bmatrix}^*}{z-w^*},$$

where we have defined

$$b_1(z) = [a_1(0)b(z) - b(0)a_1(z)]/z \quad \text{and} \quad d_1(z) = [d(z)a_1(0) - c_1(z)b(0)]. \tag{41}$$

It is clear that $b_1(z)$ is a polynomial since the numerator in (41) has a root at the origin, while $d_1(z)$ is a polynomial in z^{-1}. This completes the proof of the theorem, since $Q_b^c(z,w)$ is clearly BQH. □

Thus making any distinction between the block Hankel and BQH families is not material in the context of Schur complementation.

5.0.1. Generator for H_{n-1}^{-1}. It is clear that n steps of Schur reduction on the block generating function $H_{b,\infty}(z,w)$ would directly lead to the generating function of H_{n-1}^{-1} by way of the computed polynomials (functions) $c_{n-1}(z)$ and $d_{n-1}(z)$ starting with $c_0(z) = z$ and $d_0(z) = 0$ as in (36). In particular,

$$\begin{aligned}
&H_{n-1}^{-1}(z^{-1}, w^{-1}) \\
&\quad = [1\ z^{-1}\ z^{-2}\ \ldots] H_{n-1}^{-1} [1\ w^{-1}\ w^{-2}\ \ldots]^* \\
&\quad = -\frac{[c_{n-1}(z)d_{n-1}^*(w) - d_{n-1}(z)c_{n-1}^*(w)]}{z - w^*} \\
&\quad = -\frac{[w^{-*}c_{n-1}^*(w)z^{-1}d_{n-1}(z) - w^{-*}c_{n-1}^*(w)z^{-1}\gamma_{n-1}(z)]}{z^{-1} - w^{-*}} \\
&\quad = \frac{[z^{-1}c_{n-1}(z) \mid z^{-1}d_{n-1}(z)] \left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right] [w^{-1}c_{n-1}(w) \mid w^{-1}d_{n-1}(w)]^*}{z^{-1} - w^{-*}}.
\end{aligned}$$

Thus the generator for H_{n-1}^{-1} must be $[z^{-1}c_{n-1}(z)\ \ z^{-1}d_{n-1}(z)]$.

5.1. Singularities and block Schur complements. The Schur complement $Q_b^c(z,w)$ of the 1×1 leading principal submatrix of a BQH matrix cannot be computed if $[1\ 0]Q_b(0,0)[1\ 0]^* = Q(0,0) = 0$ (see (39)). Since we are interested in inverting Q, the first two cases (see §3.1) of singularity are not material. The third case, which requires computing a block Schur complement, must be understood in terms of its impact on the block generating function procedure.

5.1.1. The structure of Q_b. Let

$$\begin{aligned}
Q_b(z,w) &= \frac{\left[\begin{array}{c|c} a(z)b^*(w) - b(z)a^*(w) & a(z)d^*(w) - b(z)c^*(w) \\ c(z)b^*(w) - d(z)a^*(w) & c(z)d^*(w) - d(z)c^*(w) \end{array}\right]}{z - w^*} \\
&= \left[\begin{array}{c|c} Q(z,w) & Q_Y^*(w,z) \\ Q_Y(z,w) & Q_I(z,w) \end{array}\right], \text{ say.}
\end{aligned} \tag{42}$$

Let $a(z) = z^t a_t(z)$ (see (19) and Lemma 3.2). Since congruence preserves rank profile and inertia the BQH form $Q_b(z,w)$ defined as follows:

$$\begin{aligned}
&\left[\begin{array}{c|c} b(z) & O \\ O & z^{-1}d(z) \end{array}\right] R_b(z,w) \left[\begin{array}{c|c} b^*(w) & O \\ O & w^{-*}d^*(w) \end{array}\right] \\
&\quad = \left[\begin{array}{c|c} R_H(z,w) & R_Y^*(w,z) \\ R_Y(z,w) & R_I(z,w) \end{array}\right], \text{ say,}
\end{aligned} \tag{43}$$

must have the same rank profile and inertia as $Q_b(z,w)$. Let $R_b(z,w)$ be partitioned as

$$R_b(z,w) = \left[\begin{array}{c|c} R_H(z,w) & R_Y^*(w,z) \\ R_Y(z,w) & R_I(z,w) \end{array}\right], \text{ say.} \tag{44}$$

5.1.2. Structures of R_H, R_Y, and R_I. The structure of R_H has been studied in §3. Let us examine the structure of R_Y first.

$$R_Y(z,w) = \frac{y(z) - (w^*)^t g_t^*(w)}{1 - z^{-1}w^*}, \tag{45a}$$

where (see (18))

$$\begin{aligned} y(z) &= c(z)/d(z) = \textstyle\sum_{i=0}^{\infty} y_i z^{-i} \text{ and} \\ g_t(z) &= a_t(z)/b(z) = \textstyle\sum_{i=0}^{\infty} g_{t+i} z^i. \end{aligned} \tag{45b}$$

The $t \times t$ block Schur complementation on R_H modifies R_Y and produces R_Y^c as follows:

$$\left[\begin{array}{c|ccc} O & \cdot & \cdot & \cdot \\ O & \cdot & R_Y^c & \cdot \\ O & \cdot & \cdot & \cdot \end{array}\right] = R_Y - \mathcal{Y}\mathcal{U}(\underline{p})\mathcal{G}^T, \tag{46}$$

where $\mathcal{Y}$ is a t columns wide lower triangular Toeplitz matrix with the first column $[y_0^* \; y_1^* \; \dots \; \dots]^*$, $\mathcal{G}_t$ is as in (18) and $U(\underline{p})$ is a $t \times t$ upper triangular Toeplitz matrix with first row $[p_1 \; p_2 \; \dots \; p_t]$. The parameters $\{p_j\}_1^t$ are as in (16). It can be shown that the generating function $R_Y^c(z,w)$ of the above matrix R_Y^c is given by

(47a)

$$(w^*)^t R_Y^c(z,w) = R_Y(z,w) - z^{-(t-1)} y(z) \left[\sum_{k=0}^{t-1} p_{t-k} \left(\frac{z^{k+1} - (w^*)^{k+1}}{z - w^*}\right) (zw^*)^{t-1-k}\right] g_t^*(w).$$

After some algebra, (47a) reduces to

$$R_Y^c(z,w) = \frac{y(z)[1 - p^*(w)g^*(w)] - g_t^*(w)[1 - z^{-t}p(z)y(z)]}{1 - z^{-1}w^*}, \tag{47b}$$

where $p(z) = p_1 + p_2 z + \cdots + p_t z^{t-1}$. Since $c(z) = y(z)d(z)$, $a_t(z) = g_t(z)b(z)$, and $z^t b_t(z) = b(z)[1 - p(z)g_t(z)]$ (see (21)), we get

$$Q_Y^c(z,w) = \frac{c(z)b_t^*(w) - a_t^*(w)d_t(z)}{z - w^*}, \tag{48}$$

where we have defined $d_t(z) = d(z) - z^{-t}p(z)c(z)$. Next we examine the structure of R_I.

$$R_I(z,w) = \frac{y(z) - y^*(w)}{w^{-*} - z^{-1}}, \tag{49}$$

where $y(z)$ has been defined above. Then the $t \times t$ block Schur complementation on R_H modifies R_I and produces R_I^c as follows:

$$R_I^c = R_I - \mathcal{Y}\tilde{I}_t\mathcal{P}\tilde{I}_t\mathcal{Y}^*, \tag{50}$$

where $\mathcal{Y}$ is as defined before and $\mathcal{P}$ is as in (18). It can be shown that the generating function $R_I^c(z,w)$ of the above matrix R_I^c is given by

$$\begin{aligned} R_I^c(z,w) &= R_I(z,w) - z^{-(t-1)}y(z) \\ &\quad \cdot \left[\textstyle\sum_{k=0}^{t-1} p_{t-k} \left(\frac{z^{k+1} - (w^*)^{k+1}}{z - w^*}\right) (zw^*)^{t-1-k}\right] y^*(w)(w^*)^{-(t-1)} \\ &= \frac{y(z)\,[1 - (w^*)^{-t}p^*(w)y^*(w)] - [1 - z^{-t}p(z)y(z)]\,y^*(w)}{w^{-*} - z^{-1}}. \end{aligned}$$

Then the generating function of the $t \times t$ block Schur complement Q_I^c of Q_I is given by

$$R_I^c(z,w) = \frac{c(z)d_t^*(w) - d_t(z)c^*(w)}{z - w^*}. \tag{51}$$

Thus the $t \times t$ block Schur complement Q_I^c of Q_I must be BQH, viz.,

$$Q_I^c(z,w) = \frac{\left[\begin{array}{c|c} a_t(z) & b_t(z) \\ c_t(z) & d_t(z) \end{array}\right]\left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right]\left[\begin{array}{c|c} a_t(w) & b_t(w) \\ c_t(w) & d_t(w) \end{array}\right]^*}{(z - w^*)}, \text{ say,} \tag{52}$$

where $c_t(z) = c(z)$ and $d_t(z) = d(z) - z^{-t}p(z)c(z)$.

Since (21) provides a recursive procedure for computing $b_t(z)$ it may be possible to obtain one such recursion for computing $d_t(z)$ too. Next we show that it is indeed so.

5.1.3. Recursions for updating $a_t(z)$, $b_t(z)$, $c_t(z)$, and $d_t(z)$. Since $c_t(z) = c(z)$ and $d_t(z) = d(z) - z^{-t}p(z)c(z)$, the following linear recursion

$$\begin{aligned} &c_0(z) = c(z) \quad \text{and} \quad d_0(z) = d(z), \\ &c_{j+1}(z) = c_j(z), \\ &d_{j+1}(z) = d_j(z) - p_{j+1}z^{-(t-j)}c_j(z), \end{aligned} \tag{53}$$

recursively computes $c_t(z)$ and $d_t(z)$ using the $\{p_j\}$ in (16). Then we can combine the recursions (21) with the above recursion (53) and write

$$\left[\begin{array}{c|c} z & O \\ O & 1 \end{array}\right]\left[\begin{array}{c|c} a_{j+1}(z) & b_{j+1}(z) \\ c_{j+1}(z) & d_{j+1}(z) \end{array}\right] = \left[\begin{array}{c|c} a_j(z) & b_j(z) \\ c_j(z) & d_j(z) \end{array}\right]\left[\begin{array}{c|c} 1 & -p_{j+1}z^{-(t-j)} \\ O & 1 \end{array}\right]. \tag{54}$$

5.2. Polynomial recursions for Schur complementation. We can rewrite (40) as follows:

$$\begin{aligned} Q_b^c(z,w) &= \frac{|b(0)|^2\left[\begin{array}{c|c} a_1(z) & b_1(z) \\ zc_1(z) & d_1(z) \end{array}\right]\left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right]\left[\begin{array}{c|c} a_1(w) & b_1(w) \\ wc_1(w) & d_1(w) \end{array}\right]^*}{(z - w^*)} \\ &= \frac{\left[\begin{array}{c|c} z & O \\ O & 1 \end{array}\right]\left[\begin{array}{c|c} a(z) & b(z) \\ c(z) & d(z) \end{array}\right]\Theta(z)\left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right]\Theta^*(w)\left[\begin{array}{c|c} a(w) & b(w) \\ c(w) & d(w) \end{array}\right]^*}{z - w^*}, \end{aligned} \tag{55}$$

where the 2×2 matrix $\Theta(z)$ assumes three different forms that are discussed below.

5.2.1. $\Theta(z)$ for a strongly regular step. It is clear that a strongly regular step of Schur complementation is translated into a univariate map $G_b(z) \rightarrow G_b(z)\Theta(z)$ of form

$$\left[\begin{array}{c|c} z & O \\ O & 1 \end{array}\right] G_{b_{j+1}}(z) = G_{b_j}(z)\Theta_j(z)\ . \tag{56}$$

Let us define the partitions,

$$G_{b_j}(z) = \left[\begin{array}{c|c} a_j(z) & b_j(z) \\ c_j(z) & d_j(z) \end{array}\right]\ . \tag{57}$$

It should be noted that the transformation $\Theta_j(z)$ is completely determined by $a_j(z)$ and $b_j(z)$.

Case 1a. $a_j(0) \neq 0$, $b_j(0) \neq 0$. This is only a transformation step (see §3.2 for details), for which the recursion is

$$\begin{aligned} z a_{j+1}(z) &= a_j(z) - \zeta_j b_j(z), \\ b_{j+1}(z) &= b_j(z), \end{aligned} \tag{58}$$

where $\zeta_j = a_j(0)/b_j(0)$. This implies that

$$\Theta_j(z) = \left[\begin{array}{c|c} 1 & O \\ -\zeta_j & 1 \end{array}\right]. \tag{59}$$

Case 1b. $a_j(0) \neq 0$, $b_j(0) = 0$. This is only an exchange step (see §3.2 for details), for which the recursion is

$$\begin{aligned} a_{j+1}(z) &= b_j(z), \\ b_{j+1}(z) &= -a_j(z). \end{aligned} \tag{60a}$$

This implies that

$$\Theta_j(z) = \left[\begin{array}{c|c} O & -1 \\ 1 & O \end{array}\right]. \tag{60b}$$

Case 2. $a_j(0) = 0$, $b_j(0) \neq 0$, $\lim_{z\to 0} z^{-1} a_j(z) \neq 0$. This is also a strongly regular step for which the recursion is

$$\begin{aligned} z a_{j+1}(z) &= a_j(z), \\ z b_{j+1}(z) &= b_j(z) - \zeta_j z^{-1} a_j(z), \end{aligned} \tag{61}$$

where

$$\zeta_j = \lim_{z\to 0} \frac{b_j(z)}{z^{-1} a_j(z)}.$$

This implies that

$$\Theta_j(z) = \left[\begin{array}{c|c} 1 & -\zeta_j z^{-1} \\ O & 1 \end{array}\right]. \tag{62}$$

5.2.2. $\Theta(z)$ for a nonstrongly regular step. *Case* 2. $a_j(z) = k b_j(z)$ for some constant k. Since $Q_j(z, w) = 0$ all the following principal submatrices of Q are singular. Thus at this step there is no need to proceed any further.

Case 3. $a_j(0) = 0$, $b_j(0) \neq 0$ and there exists a positive integer $t_j > 1$, such that $\lim_{z\to 0} z^{-t_j} a_j(z) \neq 0$. In this case we need to find a $t \times t$ block Schur complement. At this point one needs to make repeated (t times) use of the linear recursion

$$\begin{aligned} z a_{j+1}(z) &= a_j(z), \\ z b_{j+1}(z) &= b_j(z) - \zeta_j z^{-(t-j)} a_j(z), \\ c_{j+1}(z) &= c_j(z), \\ d_{j+1}(z) &= d_j(z) - \zeta_j z^{-(t-j)} c_j(z), \end{aligned} \tag{63}$$

where

$$\zeta_j = \lim_{z\to 0} z^{t-j} b_j(z)/a_j(z) = p_{j+1}.$$

This implies that the univariate map $G_b(z) \to G_b(z)\Theta(z)$ assumes the form

$$\left[\begin{array}{c|c} z & O \\ O & 1 \end{array}\right] G_{b_{j+1}}(z) = G_{b_j}(z)\Theta_j(z), \tag{64a}$$

where

$$\Theta_j(z) = \left[\begin{array}{c|c} 1 & -\zeta_j z^{-(t-j)} \\ O & 1 \end{array}\right]. \tag{64b}$$

Case 4. $a_j(0) = b_j(0) = 0$. Since $Q_j(z,0) = 0 = Q_j(0,w)$, Q_j and all the following principal submatrices are singular, there is no need to proceed any further.

Example. Next we shall consider inverting a 3×3 nonsingular real Hankel matrix whose 2×2 principal minor is zero. Consider the 3×3 symmetric Hankel matrix T with first and last rows [1 1 1] and [1 2 2], respectively. The starting generator

$$G_0(z) = \left[\begin{array}{c|c} a_0(z) & b_0(z) \\ c_0(z) & d_0(z) \end{array}\right] = \left[\begin{array}{c|c} z(1+z+z^2+2z^3+2z^4) & 1 \\ z & O \end{array}\right]. \tag{65a}$$

Thus we have a strongly regular step and hence using (65a)–(65b), we get

$$\zeta_0 = 1, \quad t_0 = 1, \quad \text{and} \quad \Theta_0(z) = \left[\begin{array}{c|c} 1 & -z^{-1} \\ O & 1 \end{array}\right],$$

which implies

$$G_1(z) = \left[\begin{array}{c|c} a_1(z) & b_1(z) \\ c_1(z) & d_1(z) \end{array}\right] = \left[\begin{array}{c|c} 1+z+z^2+2z^3+2z^4 & -(1+z+2z^2+2z^3) \\ z & -1 \end{array}\right]. \tag{65b}$$

Its clear that the next step is just a transformation step. Thus, using (58)–(59), we get

$$\zeta_1 = -1 \quad \text{and} \quad \Theta_1(z) = \left[\begin{array}{c|c} 1 & O \\ 1 & 1 \end{array}\right],$$

which implies

$$G_2(z) = \left[\begin{array}{c|c} a_2(z) & b_2(z) \\ c_2(z) & d_2(z) \end{array}\right] = \left[\begin{array}{c|c} -z^2+2z^4 & -(1+z+z^2+2z^3) \\ z-1 & -1 \end{array}\right]. \tag{65c}$$

This takes us into a nonstrongly regular step. Using (64a)–(64b), we get

$$\zeta_2 = 1, \quad t_2 = 2, \quad \text{and} \quad \Theta_2(z) = \left[\begin{array}{c|c} 1 & -z^{-2} \\ O & 1 \end{array}\right],$$

which gives

$$G_3(z) = \left[\begin{array}{c|c} a_3(z) & b_3(z) \\ c_3(z) & d_3(z) \end{array}\right] = \left[\begin{array}{c|c} -z+2z^3 & -(1+3z+2z^2) \\ z-1 & -(1+z^{-1}-z^{-2}) \end{array}\right] \tag{65d}$$

and brings us back to a strongly regular step. Next, using (61)–(62), we get

$$\zeta_3 = 1, \quad t_3 = 1, \quad \text{and} \quad \Theta_3(z) = \left[\begin{array}{c|c} 1 & -z^{-1} \\ O & 1 \end{array}\right],$$

which implies

$$G_4(z) = \left[\begin{array}{c|c} a_4(z) & b_4(z) \\ c_4(z) & d_4(z) \end{array}\right] = \left[\begin{array}{c|c} -1+2z^2 & -(3+4z) \\ z-1 & -(2-z^{-2}) \end{array}\right]. \tag{65e}$$

The generator for the inverse of H must be $[z^{-1}c_4(z),\ z^{-1}d_4(z)] = [1-z^{-1},\ -2z^{-1}+z^{-3}]$.

A straightforward calculation verifies this claim.

5.3. QH matrices and admissibility condition. From our previous experience with QT matrices (see [13] and [20]), we may expect that inverting a QH matrix may require some preprocessing. This is generally true. However, for a special class of QH matrices, it is not required. We shall call this family of matrices admissible.

For an admissible $Q_{n-1,\infty}(z,w)$, the given $a(z)$ and $b(z)$ are such that there exist complex numbers λ and μ satisfying

$$\lambda a(z) + \mu b(z) = 1.$$

Next choose

$$Q_b(z,w) = \frac{1}{z-w^*}\left[\begin{array}{c|c} a(z) & b(z) \\ \mu z & -\lambda z \end{array}\right]\left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right]\left[\begin{array}{c|c} a(w) & b(w) \\ \mu w & -\lambda w \end{array}\right]^*, \tag{66}$$

which implies that

$$Q_{b,\infty} = \left[\begin{array}{c|c} Q_{n-1,\infty} & I_\infty \\ I_\infty & O_\infty \end{array}\right]. \tag{67}$$

Hence after n steps of Schur reduction, one obtains the generators for the inverse.

It is shown in Appendix A that for an arbitrary nonsingular QH matrix Q_{n-1}, it is $Q_{n-1}^{-\sharp}$ and not Q_{n-1}^{-1} that possesses the same displacement rank. This naturally raises the question: Is it possible to compute $Q_{n-1}^{-\sharp}$ in a similar fashion? The answer is yes, and we provide an explanation below.

It is clear that

$$Q_{n-1} = L_{n-1}\{\underline{b}-\underline{a}\}HL_{n-1}^*\{\underline{b}-\underline{a}\}, \tag{68a}$$

where H is the first $n \times n$ principal submatrix of the infinite Hankel matrix of the Markov parameters of $(b(z)+a(z))/(b(z)-a(z))$ (assuming invertibility of $L_{n-1}\{\underline{b}-\underline{a}\}$; otherwise we shall use $-\underline{a}$ instead of $\underline{a}$).

So

$$\begin{aligned} Q_{n-1}^{-\sharp} &= L_{n-1}^{-1}\{\underline{b}-\underline{a}\}\tilde{I}H^{-1}\tilde{I}L_{n-1}^{-*}\{\underline{b}-\underline{a}\}, \\ &= L_{n-1}^{-1}\{\underline{b}-\underline{a}\}\tilde{I}L_{n-1}^{*}\{\underline{b}-\underline{a}\}Q_{n-1}^{-1}L_{n-1}\{\underline{b}-\underline{a}\}\tilde{I}L_{n-1}^{-*}\{\underline{b}-\underline{a}\}. \end{aligned} \tag{68b}$$

Choose

$$Q_b(z,w) = \frac{1}{z-w^*}\left[\begin{array}{c|c} a(z) & b(z) \\ z & z \end{array}\right]\left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right]\left[\begin{array}{c|c} a(w) & b(w) \\ w & w \end{array}\right]^*, \tag{69}$$

which implies that

$$Q_{b,\infty} = \left[\begin{array}{c|c} Q_{n-1,\infty} & L_{n-1,\infty}\{\underline{b}-\underline{a}\} \\ L^*_{n-1,\infty}\{\underline{b}-\underline{a}\} & O_\infty \end{array}\right] . \tag{70}$$

Hence after n steps of Schur reduction, one obtains the generators for $L^*_{n-1}\{\underline{b}-\underline{a}\}Q^{-1}_{n-1}L_{n-1}\{\underline{b}-\underline{a}\}$. It is clear that if $[z^{-1}c_n(z)\ ,\ z^{-1}d_n(z)]$ is the generator for this matrix then $[d^\sharp_n(z)\ ,\ c^\sharp_n(z)]$ is the generator of $\tilde{I}L^*_{n-1}\{\underline{b}-\underline{a}\}Q^{-1}_{n-1}L_{n-1}\{\underline{b}-\underline{a}\}\tilde{I}$. It is possible to directly compute $c^\sharp_n(z)$ and $d^\sharp_n(z)$ instead of $c_n(z)$ and $d_n(z)$; thus we can also compute the generator for $Q^{-\sharp}_{n-1}$ directly as explained below.

If

$$\left[\begin{array}{c|c} c_{j+1}(z) & d_{j+1}(z) \end{array}\right] = \left[\begin{array}{c|c} c_j(z) & d_j(z) \end{array}\right]\Theta_j(z),$$

then we must have

$$\left[\begin{array}{c|c} c^\sharp_{j+1}(z) & d^\sharp_{j+1}(z) \end{array}\right] = \left[\begin{array}{c|c} c^\sharp_j(z) & d^\sharp_j(z) \end{array}\right]\left[\Theta^\sharp_j(z)\right]^T .$$

Given $a(z)$ and $b(z)$, it is possible to find $c_e(z)$ such that

$$[a(z)-b(z)]c_e(z) = 1 + O(z^n) . \tag{71}$$

Partition $c_e(z)$ as $c_e(z) = \hat{c}_e(z) + O(z^n)$. Then the recursions

$$\left[\begin{array}{c|c} a_{j+1}(z) & b_{j+1}(z) \end{array}\right] = \left[\begin{array}{c|c} a_j(z) & b_j(z) \end{array}\right]\Theta_j(z),$$

$$\left[\begin{array}{c|c} c^\sharp_{j+1}(z) & d^\sharp_{j+1}(z) \end{array}\right] = \left[\begin{array}{c|c} c^\sharp_j(z) & d^\sharp_j(z) \end{array}\right]\left[\Theta^\sharp_j(z)\right]^T .$$

with

$$c^\sharp_0(z) = d^\sharp_0(z) = \hat{c}_e(z) \tag{72}$$

would be sufficient and the Schur procedure would compute the generator $[d^\sharp_n(z)\ ,\ c^\sharp_n(z)]$ for $Q^{-\sharp}$ in n steps.

6. Solution to linear equations. It is clear that the solutions to the Yule–Walker type equations are obtained directly from the generators of the inverse. However, solution to a system of linear equations with an arbitrary right-hand side requires extension of the notion block QH generating function (37) to

$$Q_b(z,w) = \frac{1}{z-w^*}\left[\begin{array}{c|c} a(z) & b(z) \\ c(z) & d(z) \end{array}\right]\left[\begin{array}{c|c} O & 1 \\ -1 & O \end{array}\right]\left[\begin{array}{c|c} a(w) & b(w) \\ p(w) & q(w) \end{array}\right]^* , \tag{73}$$

where $a(z)$, $b(z)$, $c(z)$, $d(z)$, $p(z)$, and $q(z)$ are all polynomials (functions) in z and z^{-1}.

The linear equation

$$Q\underline{x} = \underline{y}$$

for an arbitrary column vector $\underline{y}$ can be solved by choosing $c(z) = c_e(z) = d(z)$ (see (71)–(72)) and $p(z) = c_e(z)y(z) = q(z)$, where $y(z) = [1\ z\ z^2\ \dots\ z^n]\underline{y}$. It develops that the Schur complements can be computed via trivial modification of the linear recursions in §5.2. In fact, the $\Theta_j(z)$s in (56)–(64b) remain the same while the Schur complementation steps are translated into two univariate maps $G^i_b(z) \to G^i_b(z)\Theta(z)$, $1 \le i \le 2$ of form

$$\left[\begin{array}{c|c} z & O \\ O & 1 \end{array}\right] G^i_{b_{j+1}}(z) = G^i_{b_j}(z)\Theta_j(z). \tag{74}$$

Let us define the partitions

$$G^1_{b_j}(z) = \left[\begin{array}{c|c} a_j(z) & b_j(z) \\ c_j(z) & d_j(z) \end{array}\right] \tag{75a}$$

and

$$G^2_{b_j}(z) = \left[\begin{array}{c|c} a_j(z) & b_j(z) \\ p_j(z) & q_j(z) \end{array}\right]. \tag{75b}$$

7. Concluding remarks. A new approach that provides a unified framework for fast and completely recursive procedures for computing modified triangular factorization of Hankel, QH, and SMQH matrices has been developed.

This recursive procedure is based on the fact that the Schur complement (or the block Schur complement) of the top-left entry (or block) of a Hankel or a QH matrix is QH, while that of a SMQH matrix is SMQH; except for Lev-Ari and Kailath [12] or Chun [2], earlier papers did not utilize this fact. The procedure presented here does not require computation of inner products and parallelizes well. Determination of the size of a block factor is done by counting the number of repeated zeros at the origin of a polynomial (see also [3]).

It also turns out that the block diagonal entries are either lower triangular Hankel matrices[1] (for Hankel and QH cases) or products of a signature matrix and a lower triangular Hankel matrix (for SMQH matrices); see (24) and (30). The inertia of such matrices (and hence the inertia of the original matrices) is determinable by inspection using certain rules due to Iohvidov [7] (see Appendix B). Computation of the inertia of SMQH matrices has applications in the problems of stability checking of linear time invariant continuous time systems and root distribution of polynomials with regard to the imaginary axis (see [11], [19], and [20]).

Using the results on modified triangular factorization, we have extended the Schur complement-based approach of Chun [2] for solving linear equations and inverting strongly regular Hankel matrices to Hankel and QH matrices with arbitrary rank profile. This leads to simultaneous derivation of the type of recursions of Lanczos (Schur) and Berlekamp–Massey (Levinson). The inversion procedures will work as long as the underlying matrices are nonsingular. Unlike most previous approaches (see Heinig and Rost [6]) no special requirement on the rank profile is needed.

Since theoretical detection of a singularity and the number of consecutive zero minors require infinite precision, it is clear that a floating point implementation of the algorithm would contain serious problems. This raises the issue of devising a numerically sound general algorithm that could deal with "nearly singular cases." Such a question goes beyond the scope of the paper and deserves a thorough investigation.

Appendix A. Derivation of (17). From (15)–(16),

$$p(z)\tilde{g}_t(z) = 1, \tag{A.1}$$

where

$$p(z) = \sum_{j=0}^{t-1} p_{j+1} z^j \quad \text{and} \quad \tilde{g}_t(z) = \sum_{j=0}^{t-1} g_{j+t} z^j.$$

[1] A matrix is called lower triangular Hankel if it has only zero entries above the main antidiagonal.

Define

$$g_{2t}(z) = \sum_{j=0}^{\infty} g_{j+2t} z^j \quad \text{then} \quad g_t(z) = \tilde{g}_t(z) + z^t g_{2t}(z), \qquad \Rightarrow \quad g_t(z) = \sum_{j=0}^{\infty} g_{j+t} z^j.$$

But if the first $t-1$ leading principal minors of Q are singular so are the first $t-1$ leading principal minors of F_H, which implies that $z^t g_t(z) = g(z)$. Therefore under this condition,

$$\frac{a(z)}{b(z)} = g(z) = z^t[\tilde{g}_t(z) + z^t g_{2t}(z)]. \tag{A.2}$$

Then from (A.1)–(A.2),

$$\frac{p(z)a_t(z)}{b(z)} = 1 + z^t p(z) g_{2t}(z), \qquad \Rightarrow p(z)a_t(z) = b(z) + z^t p(z) g_{2t}(z) b(z). \tag{A.3}$$

Since the second term $z^t p(z) g_{2t}(z) b(z)$ does not affect the coefficients of $\{z^0, \ldots, z^{t-1}\}$, the entries $\{p_j\}_{j=1}^t$ can be computed via the matrix equation (17) (see §3.1).

Appendix B. Iohvidov's inertia rules. If none of the principal minors of a matrix is zero, computing inertia is easy. Computing inertia is generally hard if a number of zero principal minors are followed by a nonzero principal minor. However, for Hankel matrices this difficulty can be avoided using a nice set of rules due to Iohvidov [7].

Iohvidov's rules. Let the sequence of successive principal minors of an $n \times n$ Hankel matrix H_{n-1} contain an isolated group of p (≥ 1) zeros:

$$(\Delta_{h-1} \neq 0), \quad \Delta_h = \Delta_{h+1} = \cdots = \Delta_{h+p-1} = 0, \quad (\Delta_{h+p} \neq 0).$$

Also, let a number of extra positive and negative eigenvalues that H_{h+p} has as compared to H_{h-1} be $\pi(h-1, h+p)$ and $\nu(h-1, h+p)$, respectively. These can be computed by formulas in Table B.1 where $\theta = (-1)^{p/2}$ Sign $(\Delta_{h+p})/(\Delta_{h-1})$.

TABLE B.1

	p odd	p even
$\pi(h-1, h+p)$	$\frac{p+1}{2}$	$\frac{p+1+\theta}{2}$
$\nu(h-1, h+p)$	$\frac{p+1}{2}$	$\frac{p+1-\theta}{2}$

However, this rule simplifies even further, since it turns out that θ = Sign ζ, where ζ is the entry on the main antidiagonal of the corresponding block diagonal factor (see (24)).

Acknowledgments. The authors would also like to thank Professor Hanoch Lev-Ari of the Northeastern University for his suggestions for improvement and constructive criticism. The first author would like to thank Drs. Victor B. Lawrence, Joseph G. Kneuer, William H. Ninke, and Bryan D. Ackland of the AT&T Bell Laboratories for encouragement and support.

REFERENCES

[1] E. R. BERLEKAMP, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[2] J. CHUN, *Fast Array Algorithms for Structured Matrices*, Ph.D. thesis, Stanford University, Stanford, CA, June 1989.

[3] T. CITRON, *Algorithms and Architectures for Error Correcting Codes*, Ph.D. thesis, Stanford University, Stanford, CA, August 1986.

[4] W. B. GRAGG, *Matrix interpretations and applications of the continued fraction algorithm*, Rocky Mountain J. Math., 4 (1974), pp. 213–225.

[5] W. B. GRAGG AND A. LINDQUIST, *On the partial realization problem*, Linear Algebra Appl., 50 (1983), pp. 277–319.

[6] G. HEINIG AND K. ROST, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Akademie-Verlag, Berlin, 1984.

[7] I. IOHVIDOV, *Hankel and Toeplitz Matrices and Forms*, Birkhauser-Verlag, Basel, Switzerland, 1982.

[8] M. KREIN AND M. NAIMARK, *The method of symmetric and Hermitian forms in the theory of the separation of roots of algebraic equations*, Linear Multilinear Algebra, 10 (1981), pp. 265–308. (Originally in Russian, Kharkov, 1936.)

[9] S. KUNG, *Multivariable and Multidimensional Systems*, Ph.D. thesis, Stanford University, Stanford, CA, June 1977.

[10] G. LABAHN, D. CHOI, AND S. CABAY, *The inverses of block Hankel and block Toeplitz matrices*, SIAM J. Comput., 19 (1990), pp. 93–123.

[11] H. LEV-ARI, Y. BISTRITZ, AND T. KAILATH, *Generalized Bezoutians and families of efficient zero-location procedures*, IEEE Trans. Circuits and Systems, 1991, pp. 170–186.

[12] H. LEV-ARI AND T. KAILATH, *Triangular factorization of structured Hermitian matrices*, Operator Theory: Advances and Applications, 18 (1986), pp. 301–324.

[13] D. PAL AND T. KAILATH, *Fast triangular factorization and inversion of Hermitian Toeplitz and related matrices with arbitrary rank profile*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1016–1042.

[14] J. PHILLIPS, *The triangular decomposition of Hankel matrices*, Math. Comp., 25 (1971), pp. 599–602.

[15] J. RISSANEN, *Recursive identification of linear systems*, SIAM J. Control, 9 (1971), pp. 420–430.

[16] ———, *Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with applications to factoring positive polynomials*, Math. Comput., 27 (1973), pp. 147–154.

[17] ———, *Solution of linear equations with Hankel and Toeplitz matrices*, Numer. Math., 22 (1974), pp. 361–366.

[18] W. F. TRENCH, *An algorithm for inversion of finite Hankel matrices*, Soc. Indust. Appl. Math., 13 (1965), pp. 1102–1107.

[19] D. PAL AND T. KAILATH, *Displacement structure approach to singular root distribution problems: The imaginary axis case*, IEEE Transactions on Circuits and Systems, 1994, to appear.

[20] D. PAL, *Fast Algorithms for Structured Matrices with Arbitrary Rank Profile*, Ph.D. thesis, Stanford University, Stanford, CA, May 1990.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 479–489, April 1994

008

THE ARNOLDI METHOD FOR NORMAL MATRICES*

THOMAS HUCKLE†

Abstract. For large Hermitian matrices the preconditionend conjugate gradient algorithm and the Lanczos algorithm are the most important methods for solving linear systems and for computing eigenvalues. There are various generalizations to the nonsymmetric case based on the Arnoldi method or on the nonsymmetric Lanczos algorithm, e.g., generalized minimal residual ((GMRES), residual minimization in a Krylov space) and conjugate gradient squared ((CGS), a biorthogonalization algorithm adapted from the biconjugate gradient method) for linear equations and the incomplete orthogonalization method and the look-ahead Lanczos algorithm for computing eigenvalues. The aim of this paper is to analyse the Arnoldi method applied to a normal matrix. It is shown that for a normal matrix A, the Arnoldi projection H_p is a normal upper Hessenberg matrix if and only if A is (1)-normal. Only in this case, H_p is tridiagonal and the Arnoldi method has the same properties as in the Hermitian case.

Key words. large linear systems, normal matrices, Krylov subspace methods, Arnoldi method, Lanczos method

AMS subject classifications. 65F10

1. Introduction.

Normal matrices appear in some applications, such as in the solution of the complex Helmholtz equation [7]. Furthermore, there exist methods for normalizing a given general matrix, for example, by Jacobi-like algorithms. In this paper, we examine the Arnoldi method applied to normal matrices. The Arnoldi process [1] is a method for transforming a general real or complex $n \times n$ matrix A to Hessenberg form. Set

$$K_p := \text{span}\ \{b, Ab, \ldots, A^{p-1}b\}, \qquad p = 1, 2, \ldots,$$

the Krylov subspace of order p related to matrix A and vector b, and H_p the orthogonal projection of A on K_p. The orthonormal vectors v_i, $i = 1, 2, \ldots, v_1 = b/\|b\|$ that form a basis of K_p can be computed iteratively by equations

$$h_{p+1,p}v_{p+1} = Av_p - \sum_{i=1}^{p} h_{i,p}v_i, \qquad h_{i,p} = v_i^H Av_p \quad \text{for } i = 1, \ldots, p. \tag{1}$$

The matrices H_p and V_p built up from $h_{i,k}, i, k = 1, \ldots, p$ and $v_i, i = 1, \ldots, p$ can be used to give estimates for the solution of the linear system $Ax = b$ and the eigenvalues of A [16], [18]. The eigenvalues of H_p are the Ritz values of A with respect to K_p.

For the following, denote by p^* the minimal p with $K_p = K_{p+1}$ or the maximal p with $\dim(K_p) = p$.

ARNOLDI ALGORITHM
$v_1 = b/\|b\|$;
for $j := 1, p$
  $w := Av_j$;
  for $i := 1, j$
    $h_{i,j} := v_i^H Av_j; w := w - h_{i,j}v_i$;
  end

* Received by the editors September 30, 1991; accepted for publication (in revised form) November 4, 1992.

† Institut für Angewandte Mathematik und Statistik, Universität Würzburg, D-97074 Würzburg, Germany (huckle@vax.rz.uni-wuerzburg.dbp.de).

$h_{j+1,j} := \|w\|$; if $h_{j+1,j} = 0$, then stop; $v_{j+1} := w/h_{j+1,j}$;
end

For this algorithm the following holds (see, e.g., [11], [13], [17]).

(i) $v_1, \ldots, v_p$ form an orthonormal basis of K_p; $H_p = V_p^H A V_p$ is an upper $p \times p$ Hessenberg matrix.

(ii) If $h_{j+1,j} = 0$ for an integer j and the algorithm stops after j steps, then $j = p^*$ and the eigenvalues of H_j are eigenvalues of A.

(iii) The Ritz values of A with respect to K_p are the eigenvalues $\lambda^{(p)}$ of H_p, and the Ritz vectors are $z^{(p)} = V_p y^{(p)}$, where $y^{(p)}$ are the normalized eigenvalues of H_p associated with $\lambda^{(p)}$.

(iv) If $e_p = (0, \ldots, 0, 1)^T$, then

$$AV_p = V_p H_p + h_{p+1,p} v_{p+1} e_p^T, \tag{2}$$

and the residual of the Ritz pair $(\lambda^{(p)}, z^{(p)})$ satisfies

$$\left\| \left(A - \lambda^{(p)} I \right) z^{(p)} \right\| = \left| h_{p+1,p} e_p^T y^{(p)} \right| . \tag{3}$$

The matrices that appear in the Arnoldi algorithm can be also represented by means of orthogonal projections [3], [4]. Let π_p be the orthogonal projection on the subspace K_p, and A_p the orthogonal projection of A on K_p; thus $A_p = \pi_p A|_{K_p}$. The operator A_p can be characterized by the equations

$$A_p(A^i b) = \begin{cases} A^{i+1} b & \text{for } i < p-1, \\ \pi_p A^p b = \sum_{j=0}^{p-1} \varepsilon_j A^j b & \text{for } i = p-1, \end{cases} \tag{4}$$

where ε_j and γ_{p+1} must be chosen in such a way that $q_{p+1} := \gamma_{p+1}(A^p - \pi_p A^p)b$ is orthogonal to K_p and has Euclidean length 1. Then, for $p < p^*, q_{p+1} \neq 0$. With

$$F_p := \begin{pmatrix} 0 & 0 & \cdots & 0 & \varepsilon_0 \\ 1 & 0 & \cdots & 0 & \varepsilon_1 \\ 0 & 1 & & 0 & \varepsilon_2 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & & \cdots & 1 & \varepsilon_{p-1} \end{pmatrix} ,$$

$L_p := (b, Ab, \ldots, A^{p-1}b)$ and $Q_p := (q_1, \ldots, q_p)$, there exists a matrix R with $L_p = Q_p R$, and there holds

$$AL_p = L_p F_p + q_{p+1} e_p^T$$

or

$$AQ_p = Q_p(RF_pR^{-1}) + (1/r_{p,p}) q_{p+1} e_p^T . \tag{5}$$

Comparing (5) with (1) and (2) we see that for $p \leq p^*$ we can set $Q_p = V_p$ and then $H_p = Q_p^T A Q_p = RF_pR^{-1}$. Hence, F_p and H_p have the same spectrum and the characteristic polynomial of H_p is of the form (see, e.g., [15])

$$p(x) = x^p - \sum_{i=0}^{p-1} \varepsilon_i x^i .$$

For symmetric A the Arnoldi method coincides with the symmetric Lanczos method [2], [12] and H_p is tridiagonal. In the general case, H_p is a full upper Hessenberg matrix, and thus the Arnoldi algorithm often has to be used with restarts or with incomplete orthogonalization [16]. In this last method the new vector v_{j+1} is computed by orthogonalizing Av_j against the previous $k+1$ vectors $v_{j-k}, \ldots, v_j$, rather than against all the previous vectors $v_1, \ldots, v_j$. Then, the resulting Hessenberg matrix is a band matrix.

The GMRES algorithm [18] for solving $Ax = b$ is based on the Arnoldi method applied to the matrix A with starting vector $v_1 = r_0/\|r_0\|$, $r_0 = Ax_0 - b$. In this process, one computes that $z \in \text{span}\,(r_0, Ar_0, \ldots, A^k r_0)$, for which the residual $\|b - A(x_0 + z)\|_2$ takes its minimal value.

For nonsymmetric matrices the Arnoldi method and the nonsymmetric Lanczos algorithm lead to different methods for computing eigenvalues and solutions of linear systems [9], [14].

2. Is H_p normal for normal A? For symmetric A the Arnoldi method yields a symmetric tridiagonal matrix H_p. If A is unitary, then the operator A_p, the orthogonal projection of A on the Krylov subspace $K_p(A, b)$, is nearly unitary [3]. This means that for $p < p^*$, there holds

$$A_p A_p^H = I + E \quad \text{with } \operatorname{rank}(E) = 1 \ .$$

Furthermore, A and A_p are both unitary if and only if $p = p^*$. This consideration leads to the question whether for normal A, A_p is normal, too.

THEOREM 1. *Let A be normal and $1 < p < p^*$. Then, A_p is normal too if and only if*

$$A^H b \in K_2(A, b) \ ,$$

or there exist γ_0 and γ_1 with

$$A^H b = \gamma_0 b + \gamma_1 A b \ .$$

Proof. Let $q_{i+1} = (A^i - \pi_i A^i)b$ be the orthogonal vectors that constitute a basis of the Krylov subspaces $K_j(A, b)$. Thus, $q_{i+1} \perp K_i$ and there exists a decomposition

$$A^H b = c + q \quad \text{with } c = \sum_{j=0}^{p-1} \gamma_j A^j b \quad \text{and} \quad q \perp K_p \ . \tag{6}$$

Furthermore, there holds

$$A_p^H = (\pi_p A \pi_p)^H = \pi_p A^H \pi_p = \pi_p A^H \quad \text{in } K_p \ .$$

The condition that A_p is normal is equivalent to the equations

$$A_p A_p^H (A^i b) = A_p^H A_p (A^i b), \qquad i = 0, 1, \ldots, p-1 \ . \tag{7}$$

For $i < p - 1$, the left and right sides of (7) can be written in the form

$$\begin{aligned} A_p A_p^H A^i b &= \pi_p A \pi_p A^H A^i b = \pi_p A \pi_p A^i A^H b \\ &= \pi_p A \pi_p A^i (c + q) = \pi_p A \pi_p \sum_{j=0}^{p-1} \gamma_j A^{i+j} b + \pi_p A \pi_p A^i q \end{aligned} \tag{8}$$

and

$$A_p^H A_p A^i b = \pi_p A^H \pi_p A A^i b = \pi_p A^H \pi_p A^{i+1} b = \pi_p A^H A^{i+1} b$$

(9)

$$= \pi_p A^{i+1} A^H b = \pi_p A^{i+1}(c+q) = \pi_p \sum_{j=0}^{p-1} \gamma_j A^{i+j+1} b + \pi_p A^{i+1} q \ .$$

For the following, let us suppose that A_p is normal and $i < p-1$. For $i = 0$, (6), (7), and (8) yield $\pi_p A q = \pi_p q = 0$, and thus $Aq \perp K_p$. With $b \in K_p$, we get

$$0 = (Aq)^H b = q^H (A^H b) = q^H (c+q) = \|q\|^2 \ ,$$

and hence, $q = 0$.

Now we prove by induction that $\gamma_{p-i} = 0$ for $i = 0, 1, \ldots, p-2$: the case when $i = 0$ is obvious. Now assume that the proposition holds for an integer $i-1$ with $0 \le i-1 \le p-3$, and $\gamma_p = \cdots = \gamma_{p-i+1} = 0$. Thus, we have

$$A^H b = \sum_{j=0}^{p-i} \gamma_j A^j b \ .$$

With (4), equation (8) can be written in the form

$$\begin{aligned} A_p A_p^H A^i b &= \pi_p A \sum_{j=0}^{p-i-1} \gamma_j A^{j+i} b + \gamma_{p-i} \pi_p A \pi_p A^p b \\ &= \pi_p \sum_{j=0}^{p-i-1} \gamma_j A^{j+i+1} b + \gamma_{p-i} \pi_p A (A^p b - q_{p+1}) \\ &= \pi_p \sum_{j=0}^{p-i} \gamma_j A^{j+i+1} b - \gamma_{p-i} \pi_p A q_{p+1} \ , \end{aligned}$$

and (9) yields

$$A_p^H A_p A^i b = \pi_p \sum_{j=0}^{p-i} \gamma_j A^{i+j+1} b \ .$$

With (7) it follows that $\gamma_{p-i} \pi_p A q_{p+1} = 0$ and, therefore, $\gamma_{p-i} A q_{p+1} \perp K_p$. For $A^i b$ we get

$$\begin{aligned} 0 &= (\gamma_{p-i} A q_{p+1})^H A^i b = \bar{\gamma}_{p-i} q_{p+1}^H (A^H A^i b) = \bar{\gamma}_{p-i} q_{p+1}^H A^i A^H b \\ &= \bar{\gamma}_{p-i} q_{p+1}^H \sum_{j=0}^{p-i} \gamma_j A^{j+i} b = \bar{\gamma}_{p-i} q_{p+1}^H * \bar{\gamma}_{p-i} A^p b \\ &= |\gamma_{p-i}|^2 q_{p+1}^H (\pi_p A^p b + q_{p+1}) = |\gamma_{p-i}|^2 \|q_{p+1}\|^2 \end{aligned}$$

and thus, because of $p < p^*$, $\gamma_{p-i} = 0$.

All in all, we have proved that for A and A_p normal, $A^H b \in K_2$.

Now, let us assume that A is a normal matrix and $A^H b = \gamma_0 b + \gamma_1 A b$. Then, for $i < p-1$ there follows

$$\begin{aligned} A_p A_p^H A^i b &= \pi_p A \pi_p A^i (\gamma_0 b + \gamma_1 A b) = \pi_p A^{i+1} (\gamma_0 b + \gamma_1 A b) \\ &= \pi_p A^{i+1} A^H b = \pi_p A^H A^{i+1} b = \pi_p A^H \pi_p A \pi_p A^i b = A_p^H A_p A^i b \ , \end{aligned}$$

and furthermore, for $i = p-1$ and using (4), we get

$$\begin{aligned} A_p A_p^H A^{p-1} b &= \pi_p A \pi_p (\gamma_0 A^{p-1} + \gamma_1 A^p) b = \gamma_0 \pi_p A^p b + \gamma_1 \pi_p A \pi_p A^p b \\ &= \sum_{j=0}^{p-1} \varepsilon_j \gamma_0 A^j b + \gamma_1 \pi_p A \sum_{j=0}^{p-1} \varepsilon_j A^j b = \sum_{j=0}^{p-1} \varepsilon_j \pi_p A^j (\gamma_0 b + \gamma_1 A b) \\ &= \sum_{j=0}^{p-1} \varepsilon_j \pi_p A^j A^H b = \pi_p A^H \sum_{j=0}^{p-1} \varepsilon_j A^j b = \pi_p A^H \pi_p A^p b = A_p^H A_p A^{p-1} b \ . \end{aligned}$$

Hence, (7) is fulfilled, and A_p is normal too. □

Remark. A_1 is always normal, and, for normal A, A_{p^*} is normal, too.

COROLLARY 1. *Let $1 < p < p^*$ and A normal. Then A_p is normal too if and only if $A_p^H = \gamma_0 I_p + \gamma_1 A_p$ holds in K_p.*

Proof. From Theorem 1 we get $A^H b = (\gamma_0 I + \gamma_1 A) b$ and thus for $i = 0, 1, \ldots, p-1$, it holds that

$$A_p^H (A^i b) = \pi_p A^i (\gamma_0 b + \gamma_1 A b) = \gamma_0 A^i b + \gamma_1 \pi_p A^{i+1} b = (\gamma_0 I_p + \gamma_1 A_p) A^i b \ .$$

On the other hand, $A_p^H = \gamma_0 I_p + \gamma_1 A_p$ yields immediately $A_p^H A_p = A_p A_p^H$. □

COROLLARY 2. *If A and A_p are normal for an integer p with $1 < p < p^*$, then A_i is normal for all $i \le p^*$.*

COROLLARY 3. *Let A be normal. Then A_p is normal for every b and $p \le p^*$ if and only if $A^H = \gamma_0 I + \gamma_1 A$.*

Proof. Let x_j, $j = 1, \ldots, n$ be orthogonal eigenvectors connected with eigenvalues λ_j of A and set $b = \sum_{j=1}^n x_j$. From $A^H b = \gamma_0 b + \gamma_1 A b$, we get

$$\sum_{j=1}^n \bar{\lambda}_j x_j = \sum_{j=1}^n (\gamma_0 + \gamma_1 \lambda_j) x_j \ .$$

Therefore, the eigenvalues of A satisfy the equation $\bar{\lambda}_j = \gamma_0 + \gamma_1 \lambda_j$ for fixed γ_0, γ_1, and it holds that $A^H = \gamma_0 I + \gamma_1 A$.

On the other hand, a normal matrix A with $A^H = \gamma_0 I + \gamma_1 A$ satisfies the assumptions of Theorem 1 and, therefore, A_p is normal. □

For normal A, the representation $A^H = \gamma_0 I + \gamma_1 A$ yields further properties of γ_0 and γ_1. Suppose that A has two different eigenvalues μ and ν. Then it follows that $\bar{\mu} - \bar{\nu} = \gamma_1 (\mu - \nu)$ and, therefore, γ_1 has absolute value 1, say $\gamma_1 = e^{i\psi}$. Furthermore, with $\gamma_0 = \bar{\mu} - e^{i\psi}\mu$, the number $i\gamma_0 e^{-i\psi/2}$ must be real.

Theorem 1 shows that for unitary matrices A, A_p is not normal.

THEOREM 2. *Let A be unitary and $1 < p < p^*$ for a given b. Then A_p is not normal.*

Proof. Let $1 < p < p^*$. From Theorem 1 we get $A^H b = (\gamma_0 I + \gamma_1 A) b$, and this leads to

$$(\gamma_1 A^2 + \gamma_0 A - I) b = 0 \ .$$

Now it is possible to choose orthogonal eigenvectors x_j of A connected with pairwise different eigenvalues λ_j, such that b has the representation

$$b = \sum_{j=1}^k \alpha_j x_j \quad \text{and} \quad \alpha_j \neq 0 \ . \tag{10}$$

This yields $\gamma_1\lambda_j^2 + \gamma_0\lambda_j - 1 = 0$ for $j = 1, \ldots, k$. Therefore, there must hold $k \leq 2$, and this implies that $p^* \leq 2$. □

The representation (10) of b shows how p^* depends on b and shows that $p^* = k$. This is a consequence of the equation

$$(b, Ab, \ldots, A^{p-1}b) = (\alpha_1 x_1, \ldots, \alpha_k x_k) \begin{pmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{p-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{p-1} \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_k & \cdots & \lambda_k^{p-1} \end{pmatrix} .$$

A Theorem of Gantmacher [8] states that a matrix A is normal if and only if there exists a polynomial p of degree $\leq n$ with $A^H = p(A)$. For what follows we set

$$n(A) := \text{the minimal degree of a polynomial } p \text{ with } A^H = p(A) . \tag{11}$$

It is easy to see that $n(A)$ is less or equal to the degree of the minimal polynomial of A. With this notation we get the following theorem.

THEOREM 3. *Let A be normal and $1 < p < p^*, p - 1 \geq k > 1$ for given b. Furthermore, assume that $A^H b = p(A)b$ with a polynomial of degree $\leq k$. Then it holds that*

$$A_p^H A_p - A_p A_p^H =: E \quad \textit{and} \quad \text{rank } (E) \leq k .$$

Proof. For $i = 0, 1, \ldots, p - 1 - k$, we get

$$A_p A_p^H A^i b = \pi_p A \pi_p A^i p(A) b = \pi_p A^{i+1} A^H b$$

and

$$A_p^H A_p A^i b = \pi_p A^H \pi_p A A^i b = \pi_p A^{i+1} A^H b .$$

Therefore, $Ex = 0$ for all $x \in K_{p-k}(A, b)$. Hence, the null space of E is at least of dimension $p - k$. □

The matrices that satisfy $B^H = \gamma_0 I + \gamma_1 B$ form an important class with interesting properties. Faber and Manteuffel [6] showed that these matrices coincide with the class of matrices that can be written in the form

$$B = aI + e^{i\phi} S \quad \text{with Hermitian } S . \tag{12}$$

Thereby, (12) means that B is normal, and the eigenvalues of B are collinear and lie on a straight line $a + e^{i\psi} t, t$ real. The class (12) is the weakest generalisation of the class of Hermitian matrices. Furthermore, γ_0, γ_1 and a, ϕ are connected by

$$\gamma_1 = e^{-2i\phi} \quad \text{and} \quad \text{imaginary part}(ae^{-i\phi}) = -\frac{i}{2}\gamma_0 e^{i\phi} . \tag{13}$$

Following [6], these matrices are the only matrices for which one can define a conjugate gradient method with three-term recursion. This class appears also in applications, e.g., in connection with the complex Helmholtz equation [7].

Now, the representation (10) allows a new formulation of Theorem 1.

COROLLARY 4. *For normal A, A_p is also normal, $1 < p < p^*$, if and only if in the representation* (10) *of b all eigenvalues λ_i, $i = 1, \ldots, k$, are collinear.*

Proof. Theorem 1 and (10) show that A_p is normal if and only if

$$\bar{\lambda}_i = \gamma_0 + \gamma_1 \lambda_i, \qquad i = 1, \ldots, k \ .$$

But this means that all λ_i lie on a straight line $a + e^{i\psi}t$ with a and ϕ determined by (13). □

Hence, for normal A, A_p is also normal if and only if all eigenvalues of A_{p^*}—and therefore all eigenvalues of A_p, $p = 1, \ldots, p^*$—are collinear.

Furthermore, there holds the following theorem.

THEOREM 4. *Nondecomposable, normal, tridiagonal matrices are of the form* (12) *and have collinear eigenvalues.*

Proof. With

$$T = \begin{pmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & a_{n-1} & b_{n-1} \\ & & & c_{n-1} & a_n \end{pmatrix} ,$$

the main diagonal of

$$TT^H = T^H T \ , \tag{14}$$

shows that $|c_j| = |b_j|$ for $j = 1, \ldots, n-1$. Considering the subdiagonal elements $a_{j,j-2}$, we can see that $c_j = e^{i\delta}\bar{b}_j$ with fixed δ. All b_j and c_j are different from zero and thus (14) yields

$$a_{j+1} - a_j = e^{i\delta}(\bar{a}_{j+1} - \bar{a}_j) \ .$$

Therefore, for all $j = 1, \ldots, n-1$, there holds

$$a_{j+1} - a_j = r_j e^{i\delta/2}, \qquad r_j \text{ real} \ ,$$

or

$$a_j = a_1 + \sum_{k=1}^{j-1} r_k e^{i\delta/2} \ .$$

All in all, we get

$$T = a_1 I + e^{i\delta/2} \begin{pmatrix} 0 & e^{-i\delta/2}b_1 & & & \\ e^{i\delta/2}\bar{b}_1 & r_1 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \sum_{k=1}^{n-2} r_k & e^{-i\delta/2}b_{n-1} \\ & & & e^{-i\delta/2}\bar{b}_{n-1} & \sum_{k=1}^{n-1} r_k \end{pmatrix} . \qquad \square$$

3. Properties of the projected matrix H_p for normal A. For the following, let A be a normal $n \times n$ matrix. In §2 we showed that the matrix H_p that arises in the Arnoldi process is generally no longer normal. But H_{p^*} is also normal. Therefore,

there exists a polynomial of degree $n(A_{p^*})$, defined in (11), with $A_{p^*}^H = p(A_{p^*})$ or $A^H b = p(A)b$. With $q_i = v_i$ defined by the Arnoldi algorithm, there follows

$$\overline{h_{i,j}} = q_j^H (A^H q_i) = q_j^H (p(A)b) = 0$$

for $i > n(A_{p^*}) + j$. Thereby, p is a polynomial of degree less or equal $n(A_{p^*}) + i - 1$. Hence, we have proved the following theorem.

THEOREM 5. *The upper Hessenberg matrix generated by the Arnoldi process has upper bandwidth* $n(A_{p^*}) \leq n(A)$.

Remarks. If A_p and also H_p are normal for $1 < p < p^*$ and have the form (12), then there holds $n(A_{p^*}) \leq 1$, and H_p is a normal tridiagonal matrix for every p. Therefore, H_p is of the form described in Theorem 4. In particular, the linear system that must be solved in the GMRES algorithm is of tridiagonal form if A satisfies (12).

Faber and Manteuffel showed in [6] that for a normal matrix B with $n(B) > 1$, the degree of the minimal polynomial of B is less than or equal to $n(B)^2$. Hence, in this case, $n(A_{p^*})$ will be very large and the band structure of H_p will be visible only for large p.

Further properties of H_p result from the consideration that H_p is a leading principal submatrix of the normal upper Hessenberg matrix H_{p^*}.

LEMMA 1. *Let A be normal and partitioned in the form*

$$A = \begin{pmatrix} B & C \\ D & E \end{pmatrix}$$

with quadratic matrices B and E. Then, $\|C\|_F = \|D\|_F$, and the departure of normality of B is given by

$$\left\|BB^H - B^H B\right\|_F \leq \|D\|_F^2 + \|C\|_F^2 = 2\|C\|_F^2 \ .$$

Proof. Equation $AA^H = A^H A$ yields

$$BB^H - B^H B = D^H D - CC^H \ . \tag{15}$$

Considering only the trace of (15) and estimating the right side of (15) proves the lemma. □

Hence, for the upper Hessenberg matrix H_p generated by the Arnoldi process, we get

$$\left\|H_p H_p^H - H_p^H H_p\right\|_F \leq 2\left|h_{p+1,p}\right|^2 \ .$$

Furthermore, there holds the theorem that follows.

THEOREM 6. *For normal A, the departure of normality of H_p satisfies*

$$\left\|H_p H_p^H - H_p^H H_p\right\|_F \leq 2\|A\|^2 \sqrt{n-p} \ .$$

Proof. It holds that

$$H_p H_p^H - H_p^H H_p = Q_p^H \left(A(\pi_p - I_p)A^H + A^H(I_p - \pi_p)A\right) Q_p \ . \qquad \square$$

Without loss of generality, from now on we assume that $p^* = n$ holds. Lemma 1 shows that for the normal Hessenberg matrix $H_n = (h_{i,j})_{i,j=1}^n$,

$$\left|h_{p+1,p}\right|^2 = \sum_{i=1}^{p} \sum_{j=p+1}^{n} \left|h_{i,j}\right|^2 \tag{16}$$

for $p = 1, 2, \ldots, n-1$. For what follows denote by D_i the mean value of the squared absolute values of the elements on the ith superdiagonal,

$$D_i = \frac{1}{n-i} \sum_{j=1}^{n-i} |h_{j,i+j}|^2 \ ,$$

and M_i is the maximum of this value. In the same way, set D the mean value of the squared subdiagonal elements,

$$D = \frac{1}{n-1} \sum_{j=1}^{n-1} |h_{j+1,j}|^2 \ ,$$

and M is the maximum. By summing (16) and taking into consideration the multiplicities of the superdiagonal elements in this sum, we get

$$D = \sum_{i=1}^{n-1} \frac{i(n-i)}{(n-1)} D_i \ .$$

For each superdiagonal this gives

$$D_i \leq \frac{(n-1)}{i(n-i)} D, \qquad i = 1, \ldots, n-1$$

and

$$M_i \leq \frac{n-1}{i} D, \qquad M_i \leq M \ .$$

More precisely, with Lemma 1 we get even

$$|h_{i,i+j}| \leq \min_{k=1}^{j} |h_{i+k,i+k-1}| \ ,$$

especially for $i = 1$ and $i = n-1$

$$|h_{1,2}| \leq |h_{2,1}| \quad \text{and} \quad |h_{1,n}| \leq \min_{k=1}^{n-1} |h_{k+1,k}| \ .$$

Then, using

$$\sum_{i=1}^{n-1} (n-i) D_i \leq \sum_{i=1}^{n-1} i(n-i) D_i \leq (n-1) D \ ,$$

we can estimate the Frobenius norm of H_n by

$$\|H_n\|_F^2 \leq n D_0 + 2(n-1) D \leq n M_0 + 2(n-1) M \ .$$

The mean value of the squared superdiagonal elements is bounded from above by

$$\frac{2D}{n} \leq \frac{2M}{n} \ .$$

Because of

$$\left(\sum_{k=1}^{m} |a_k| \right)^2 \leq m \sum_{k=1}^{m} |a_k|^2 \ ,$$

the mean value of the absolute values of these elements is bounded by

$$\sqrt{2D/n} \le \sqrt{2M/n} \, .$$

For the submatrix H_p, $p \le p^*$, (16) yields

$$|h_{k+1,k}|^2 \ge \sum_{i=1}^{k} \sum_{j=k+1}^{p} |h_{i,j}|^2 \tag{17}$$

for $k = 1, 2, \ldots, p-1$. Let us denote by d, d_i, m, and m_i the quantities of H_p corresponding to D, D_i, M, and M_i. Therefore,

$$d_i = \frac{1}{p-i} \sum_{j=1}^{p-i} |h_{j,j+i}|^2, \qquad d = \frac{1}{p-1} \sum_{j=1}^{p-1} |h_{j+1,j}|^2 \, ,$$

and m and m_i are the associated maxima. Then, from (17) we get the following theorem.

THEOREM 7. *For normal A, the matrix H_p generated by the Arnoldi process has the following properties*:

(i) $d_i \le [(p-1)/i(p-i)]d$ *and* $m_i \le (p-1/i)d$;

(ii) $\|H_p\|_F^2 \le pd_0 + 2(p-1)d \le pm_0 + 2(p-1)m$;

(iii) *the mean value of all squared superdiagonal elements of H_p is less than or equal to $2d/p$*;

(iv) *the mean value of the absolute values of all superdiagonal elements of H_p is less than or equal to $\sqrt{2d/p}$.*

(v) $|h_{i,i+j}| \le \min_{k=1}^{j} |h_{k+i,k+i-1}|$.

Similar results can be found in [5]. If we number the superdiagonals in such a way that the main diagonal is associated with $i = 0$, and $h_{1,p}$ is the diagonal with $i = p$, then Theorem 7(i) states that the mean value of all elements in a superdiagonal is maximal for $i = 1$ and $i = p$ and is minimal for $i = \lfloor p/2 \rfloor$. Furthermore, by (v), we get an upper bound for $|h_{i,i+j}|$ that is decreasing for increasing j. This gives an explanation for the observation of Saad [16] that in many examples the elements $|h_{i,j}|$ become slowly smaller as j increases with i fixed.

A normal matrix with extreme behaviour in the sense of Theorem 7 is the unitary Frobenius matrix

$$\begin{pmatrix} 0 & & & 1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix} . \tag{18}$$

This matrix is also an example with bad behaviour for the Arnoldi method and for GMRES with starting vector e_1; the Ritz values are all zero up to the last iteration, and both algorithms give no improvement of the starting values up to that point.

In the Hermitian case, we get for the Ritz values generated by the Arnoldi process from (3), that for every Ritz value $\lambda_i^{(p)}$ there exists an eigenvalue λ of A with

$$\left| \lambda_i^{(p)} - \lambda \right| \le \left| h_{p+1,p} e_p^T y_i^{(p)} \right| \, . \tag{19}$$

Furthermore, all Ritz values and eigenvalues can be ordered in such a way that (see [10], [13])

$$\sum_{i=1}^{p} \left|\lambda_{\pi(i)} - \lambda_i^{(p)}\right|^2 \leq 2\,|h_{p+1,p}|^2 \ . \tag{20}$$

In view of (3), equation (19) remains true for normal A. But, example (18) shows that (20) is not always true for normal A.

4. Conclusions. The Arnoldi method applied to a normal matrix leads to upper Hessenberg matrices H_p with special properties. But H_p is tridiagonal only for the well-known class (12). For this class, the Arnoldi method shows the same behaviour as for symmetric matrices and is thus superior to all other Lanczos-type methods. In nearly all the remaining cases, the convergence behaviour of the Arnoldi method can be very poor.

Acknowledgment. I would like to thank Roland Freund for bringing some important references to my attention and also the referees for some helpful comments.

REFERENCES

[1] W. G. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
[2] J. CULLUM AND R. WILLOUGHBY, *Lanczos Algorithms for Large Symmetric Eigenvalue Computation Vol.* I Theory, Birkhäuser, Boston, 1985.
[3] G. CYBENKO, *Restrictions of normal operators, Padé approximation and autoregressive time series*, SIAM J. Math. Anal., 15 (1984), pp. 753–767.
[4] ———, *An explicit formula for Lanczos polynomials*, Linear Algebra Appl., 88/89, (1987), pp. 99–115.
[5] P. J. EBERLEIN AND C. P. HUANG, *Global convergence of the QR algorithm for unitary matrices with some results for normal matrices*, SIAM J. Numer. Anal., 12 (1975), pp. 97–104.
[6] V. FABER AND T. MANTEUFFEL, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
[7] R. FREUND, *On conjugate gradient type methods and polynomial preconditioners for a class of complex non-Hermitian matrices*, Numer. Math., submitted.
[8] F. R. GANTMACHER, *Matrizentheorie*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1986.
[9] M. GUTKNECHT, *A completed theory of the unsymmetric Lanczos process and related algorithms Part* I, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 594–639.
[10] W. KAHAN, *Inclusion theorems for clusters of eigenvalues of Hermitian matrices*, University of Toronto, Ontario, Canada, 1967.
[11] S. KANIEL, *Estimates for some computational techniques in linear algebra*, Math. Comput., 20 (1966), pp. 369–378.
[12] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–282.
[13] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[14] B. N. PARLETT, D. R. TAYLOR, A. LIU ZHISHUN, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp., 44 (1985), pp. 105–124.
[15] A. RUHE, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.
[16] Y. SAAD, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.
[17] ———, *On the rates of convergence of the Lanczos and the block-Lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.
[18] Y. SAAD, M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 490–495, April 1994

009

UNICITY OF BIPROPORTION*

LOUIS DE MESNARD†

Abstract. The biproportion of S on margins of M is called the intern composition law, K: $(S, M) \mapsto X = K(S, M)/X = ASB$.

A and B are diagonal matrices, algorithmically computed, providing the respect of margins of M. Biproportion is an empirical concept. In this paper, the author shows that any algorithm used to compute a biproportion leads to the same result. Then the concept is unique and no longer empirical. Some special properties are also indicated.

Key words. biproportion, biproportional, RAS, updating matrices

AMS subject classifications. 15A15, 14N05, 65Q05

1. Introduction. For many years, economists have used biproportion. For example, R. Stone and M. Bacharach used the RAS method in the sixties, as well as many others (see a short survey in [3]). This method is useful when we want to update or filter a matrix, by projecting this matrix on the margins of another matrix, (i.e., equip it with the margins of another matrix).

For a long time biproportion lacked theoretical basis: RAS is a purely empirical method.[1] It is possible to give theoretical foundations to biproportion [3], but numerous algorithms must exist to compute a biproportion. It is not satisfactory because of this multiplicity of algorithms, biproportion remains empirical. However, we do not want to compare different algorithms from the standpoint of numerical computation as in [2].

In this paper, we show that any possible algorithm leads to the same result. This settles the theory that biproportion is unique as proportion and is no longer an empirical concept. Some parts of arguments and demonstrations still come from [3].

1. Definition of biproportion.

1.1. Ordinary proportion. We present this well-known concept in a special form. Let x and s be two real vectors. There exists a *proportion* behind these two vectors, that is to say, x is proportional to s with the ratio k, if

$$x_i = ks_i \quad \forall i,$$

where k is the ratio of the margins

$$k = \frac{x.}{s.} \quad \text{with } x. = \sum_i x_i \quad \text{and} \quad s. = \sum_i s_i.$$

1.2. Biproportion. Let E be the set of $(n \times m)$ matrices of real positive numbers. S and M are two matrices belonging to this set, and A and B are two diagonal matrices having the same dimensions. The margins of M are the numbers $m_{i.}$ and $m_{.j}$ so that

$$\sum_j m_{ij} = m_{i.} \quad \forall i \quad \text{and} \quad \sum_i m_{ij} = m_{.j} \quad \forall j.$$

* Received by the editors November 25, 1991; accepted for publication (in revised form) November 10, 1992.

† Faculty of Economics, University of Dijon, 4 Boulevard Gabriel, 21000 Dijon, France (lmesnard@satie.u-bourgogne.fr).

[1] The term RAS comes from rAs, where r and s are vectors and A is a matrix.

We call *biproportion* of S on margins of M, the intern composition law K, of $E \times E \mapsto E$, such that

$$(S, M) \overset{K}{\mapsto} X = K(S, M): \quad X = ASB \quad \text{and} \quad \sum_j x_{ij} = m_{i.} \quad \forall i \quad \text{and} \quad \sum_i x_{ij} = m_{.j} \quad \forall j.$$

With Bacharach, we can say that X is biproportional to S according to M [1].

The solution can be found, for example, using the algorithm (A_i and B_j are scalars)

$$A_i = m_{i.}\left(\sum_j B_j s_{ij}\right)^{-1} \quad \forall i \quad \text{and} \quad B_j = m_{.j}\left(\sum_i A_i s_{ij}\right)^{-1} \quad \forall j.$$

We call these equations the K algorithm; they satisfy marginal constraints of M:

$$\sum_j x_{ij} = A_i \sum_j B_j s_{ij} = m_{i.}\left(\sum_j B_j s_{ij}\right)^{-1} \sum_j B_j s_{ij} = m_{i.} \quad \forall i,$$

$$\sum_i x_{ij} = B_j \sum_i A_i s_{ij} = m_{.j}\left(\sum_i A_i s_{ij}\right)^{-1} \sum_i A_i s_{ij} = m_{.j} \quad \forall j.$$

Many justifications may be found. For example, we may use the theory of information, transforming it a little [3]:

$$\text{Min} \sum_i \sum_j x_{ij} \operatorname{Log} \frac{x_{ij}}{s_{ij}} \quad \text{with} \quad \sum_i x_{ij} = m_{.j} \quad \forall j, \quad \text{and} \quad \sum_j x_{ij} = m_{i.} \quad \forall i.$$

Many other algorithms may be used such as RAS or modified RAS [2].

2. Unicity of biproportion. The algorithm K corresponding to the equations of A_i and B_j allows one to calculate biproportion[2] as it is a converging process toward the solution. We may think that an undetermined number of other algorithms can play this role. Some algorithms, like RAS, are known. Bachem and Korte [2] demonstrate identity of the algorithms of RAS and K, but do not demonstrate universality of biproportion, because they only demonstrate RAS. The question is whether or not all of them give the same solution as K. If not, biproportion seriously lacks consistency, because its result depends on the choices of the algorithm. Otherwise, biproportion appears as universal, and so has a very stable basis.

2.1. Idempotency. Biproportion of a matrix on its own margins is a neutral operation.

THEOREM 1. *$K(S, S') = S$, where S' has the same margins as S. In particular, we have $K(S, S) = S$.*

Proof. Suppose, without loss of generality, that we start the algorithm of biproportion by $B'_j = 1$ for all j. At the first step, we obtain the following equilibrating factors A'_i:

$$A'_i(1) = \frac{s_{i.}}{\sum_j B'_j(0) s_{ij}} = \frac{s_{i.}}{\sum_j s_{ij}} = 1 \quad \forall i,$$

because S is equipped with its own margins. Then,

$$B'_j(1) = \frac{s_{.j}}{\sum_i A'_i(0) s_{ij}} = \frac{s_{.j}}{\sum_i s_{ij}} = 1 \quad \forall j.$$

[2] The conditions of convergence are given, and unicity of the solution is proved in [3].

By induction, we can show that the two series (A') and (B') are uniformly unitary. Let us suppose that $A'_i(t) = 1$ for all i, and $B'_j(t) = 1$ for all j is true until t, and let us verify that this is true for $t + 1$:

$$A'_i(t+1) = \frac{s_{i.}}{\sum_j B'_j(t) s_{ij}} = \frac{s_{i.}}{\sum_j s_{ij}} = 1 \quad \forall i$$

and

$$B'_j(t+1) = \frac{s_{.j}}{\sum_i A'_i(t+1) s_{ij}} = \frac{s_{.j}}{\sum_i s_{ij}} = 1 \quad \forall j.$$

Therefore, $A' = l$ and $B' = l$. □

2.2. Composition of biproportions and associativity. Biproportion is not linear regarding to the variables, even if it is a generalisation of the proportion, which is linear because it is one dimensional. In an ordinary proportion, a double use of proportion remains a proportion:

$$x = sm \quad \text{and} \quad x' = xm' \Rightarrow x' = smm' \Rightarrow x' = sm''.$$

In the same way, a double use of biproportion remains a biproportion: biproportion is globally linear. Furthermore, the composition of two biproportions is identical to the second biproportion.

THEOREM 2. $K[K(S, M), M'] = K[S, M']$. *Note that it could be generalised to compositions of biproportions*

$$K\{K \cdots K[K(S, M), M'] \ldots, M^{(n)}\} = K\{S, M^{(n)}\}.$$

Proof. Let S and M be two matrices. We write a first biproportion

$$X = K(S, M) = ASB \text{ with } x_{ij} = A_i s_{ij} B_j \quad \forall i, j,$$

and

$$A_i = m_{i.} \left(\sum_j B_j s_{ij} \right)^{-1} \quad \forall i,$$

$$B_j = m_{.j} \left(\sum_i A_i s_{ij} \right)^{-1} \quad \forall j.$$

Consider then the composition with a second biproportion

$$X' = K(X, M') = A'XB' \text{ with } x'_{ij} = A'_i x_{ij} B'_j \quad \forall i, j,$$

and

$$A'_i = m'_{i.} \left(\sum_j B'_j x_{ij} \right)^{-1} \quad \forall i,$$

$$B'_j = m'_{.j} \left(\sum_i A'_i x_{ij} \right)^{-1} \quad \forall j.$$

Let us calculate A' and B':

$$A'_i = m'_{i.}\left(\sum_j B'_j A_i s_{ij} B_j\right)^{-1} \quad \forall i,$$

$$\Leftrightarrow A'_i A_i = m'_{i.}\left(\sum_j B'_j B_j s_{ij}\right)^{-1} \quad \forall i,$$

$$\Leftrightarrow A''_i = m'_{i.}\left(\sum_j B''_j s_{ij}\right)^{-1} \quad \forall i,$$

with

$$A'_i A_i = A''_i \quad \text{and} \quad B'_j B_j = B''_j.$$

In the same way

$$B''_j = m'_{.j}\left(\sum_i A''_i s_{ij}\right)^{-1} \quad \forall j$$

and

$$x'_{ij} = A'_i A_i s_{ij} B'_j B_j = A''_i s_{ij} B''_j \quad \forall i, j.$$

Therefore, X', biproportional to $K(S, M)$ according to M', is also biproportional to S according to M'. □

Also, the intern composition law K is associative.

COROLLARY. *It holds that* $K[K(S\,M), M'] = K\{S, [K(M, M')]\}$.

Proof. The proof is clear. $K[K(S, M), M'] = K(S, M')$ and $K\{S, [K(M, M')]\} = K(S, M')$. □

2.3. Unicity. We have the following lemma.

LEMMA 1. *If* K^q *is any nonspecified algorithm (the form of U and V is unknown), with* $X = K^q(S, M) = USV$, *and if* $X' = K(X, M) = AXB$, *then* $X' = X$, *i.e.,* $K[K^q(S, M), M] = K^q(S, M)$. *As a special case, we get* $K[K(S, M), M] = K(S, M)$.

Proof. $K[K^q(S, M), M] = K(X, M)$. X has the same margins as M, then $K(X, M) = K(X, X)$. From the property of idempotency, $K(X, X) = X = K^q(S, M)$. □

Whatever algorithm is used, let us show that one biproportion is identical to another. Then choosing K or any other algorithm is of no consequence.

THEOREM OF UNICITY. *If* K^q *is any nonspecified algorithm (the form of U and V is unknown), with* $X = K^q(S, M) = USV$, *then U and V may be put in the standard form K.*

Proof. Let $X' = K(X, M) = AXB$; K is the specified algorithm. According to Lemma 1, this composition of biproportions is a neutral operation. Therefore, matrices A and B are the identity matrix: $A_i = 1$ for all i and $B_j = 1$ for all j.

Then, as $x_{ij} = U_i s_{ij} V_j$ for all ij

$$A_i = \frac{m_{i.}}{\sum_j B_j x_{ij}} \Leftrightarrow 1 = \frac{m_{i.}}{\sum_j U_i s_{ij} V_j} \Leftrightarrow U_i = \frac{m_{i.}}{\sum_j s_{ij} V_j} \quad \forall i,$$

$$B_j = \frac{m_{.j}}{\sum_i A_i x_{ij}} \Leftrightarrow 1 = \frac{m_{.j}}{\sum_i U_i s_{ij} V_j} \Leftrightarrow V_j = \frac{m_{.j}}{\sum_i s_{ij} U_i} \quad \forall j.$$

Therefore, U_i and V_j have the same form as the equilibrating factors of the biproportion. □

This result was obtained without specifying the algorithm of the projector K^q. It is sufficient to know that K^q is a biproportion of S on the margins of M.

We may apply this result to RAS. The solution of K and the solution of RAS are identical. RAS is an algorithm that plays the role of K^q, and the proof we have given does not need to be specified in terms of its algorithm. Then, the necessary and sufficient conditions of existence and convergence of the solution (not presented here, see [1] or [3]) are identical for K, for RAS, and for any biproportional algorithm. Nonnegativity of elements of S and M is required.

Biproportion appears to be as universal as proportion is in the one-dimensional world. This should end the discussion about the theoretical character of biproportion.

3. Some properties specific to biproportion.

3.1. Ineffectiveness of separability. Let us show that if we multiply every term of a same row or column of the matrix S by the same value, we do not change anything in a biproportion. Therefore such a modification of the s_{ij} is ineffective because it is separable in rows and columns.

We name *separable modification* of the terms of a matrix S with n rows and m columns, a modification of S that can be reduced to the left product by a diagonal matrix U of size n, and to the right product by another diagonal matrix V of size m: $S' = USV$. A separable modification does not change a biproportion.

THEOREM 3. *Let* $X = K(S, M) = ASB$ *and* $X' = K(S', M) = A'S'B'$ *with* $S' = USV$. *Then* $X' = ASB = K(S, M)$.

Proof. It holds that $X' = A'S'B' = A'USVB' = A''SB''$, putting $A'' = A'U$ and $B'' = B'V$.

Let us prove that $A'' = A$ and $B'' = B$.

$$A'_i = m_{i.}\left(\sum_j B'_j s'_{ij}\right)^{-1} \quad \forall i, j,$$

$$\Rightarrow A'_i = m_{i.}\left(\sum_j B'_j u_i s_{ij} v_j\right)^{-1} \quad \forall i, j,$$

$$\Rightarrow A''_i = m_{i.}\left(\sum_j B''_j s_{ij}\right)^{-1} \quad \forall i, j,$$

and

$$B'_j = m_{.j}\left(\sum_i A'_i s'_{ij}\right)^{-1} \quad \forall i, j,$$

$$\Rightarrow B'_j = m_{.j}\left(\sum_i A'_i u_i s'_{ij} v_j\right)^{-1} \quad \forall i, j,$$

$$\Rightarrow B''_j = m_{.j}\left(\sum_i A''_i s_{ij}\right)^{-1} \quad \forall i, j.$$

Then, the iterative form will be

$$B''_j(t) = m_{.j}\left(\sum_i A''_i(t) s_{ij}\right)^{-1} \quad \forall i, j,$$

$$A''_i(t+1) = m_{i.}\left(\sum_j B''_j(t) s_{ij}\right)^{-1} \quad \forall i, j.$$

If we start with $A_i''(1) = A_i(1)$, we get

$$B_j''(1) = B_j(1) \quad \forall j,$$

$$A_i''(2) = A_i(2) \quad \forall i,$$

and by induction

$$B_j''(t) = B_j(t) \quad \forall j,$$

$$A_i''(t+1) = A_i(t+1) \quad \forall i.$$

Therefore, in the limit, $A'' = A$ and $B'' = B$. Then, $X' = A''SB'' = ASB = X$. □

3.2. Nonreciprocality. Proportion has the property of *reciprocality*. If a vector V_2 is proportional to a vector V_1 according to k, then V_1 is proportional to V_2, according to $\frac{1}{k}$, and

$$V_2 - V_1 k = 0 \Rightarrow V_1 - V_2/k = 0.$$

On the other hand, biproportion does not verify the property of reciprocality. (The proof is given by counter-examples in [3].) Consider two matrices T_1 and T_2.

$$T_{2ij} - K(T_{1ij}, T_{2ij}) \# T_{1ij} - K(T_{2ij}, T_{1ij}) \quad \forall i, j.$$

If the indices 1 and 2 represent periods of time, we can describe the first term $T_{2ij} - K(T_{1ij}, T_{2ij})$ as *prospective*, and the second term $T_{1ij} - K(T_{2ij}, T_{1ij})$ as *retrospective*. In practical applications, nonreciprocality implies that we must calculate twice, once in the prospective way and once in the retrospective way, and compare the results [3].

Another difficulty is due to nonreciprocality. As proportion is reciprocal, if $V_2 = V_1 k$, we can retrieve V_1 knowing k. As biproportion is not reciprocal, given a biproportion $X = K(S, M)$, we cannot retrieve S without knowing the initial margins of S, even if we know M, because of the transcendent nature of K.

The property of nonreciprocality could be used to build a coding system for matrices. To code a matrix is to use a transformation of it that is sufficiently hard to reverse, to prevent a return from the transformed matrix toward the original matrix. The transformed matrix X (it could be a matrix of numbers, such as a computer image) is then coded. Only authorised persons could retrieve the original matrix S, knowing the margins of S; with the simple proportion applied to a vector, any person could know the elements of this vector with a factor of proportionality and so, nothing is coded.

However, the system is limited because the coding person who knows the matrix S to be coded also knows the decoding key, i.e., the margins of S.

4. Conclusion. Since the choice of the algorithm of biproportion is indifferent, biproportion appears as universal as proportion. Then, biproportion is no longer empirical, and it is possible to make the systematic mathematical treatment of biproportion. However, biproportion is not a simple generalisation of proportion: it is iteratively computed, it is only partially linear, and it does satisfy reciprocality.

REFERENCES

[1] M. BACHARACH, *Biproportional Matrices and Input-Output Change*, Cambridge University Press, Cambridge, UK, 1970.

[2] A. BACHEM AND B. KORTE, *On the* RAS-*algorithm*, Comput., 23 (1979), pp. 189–198.

[3] L. DE MESNARD, *Dynamique de la structure industrielle française*, Economica, Paris, 1990.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 496–518, April 1994

010

CHARACTERIZATIONS OF SCALING FUNCTIONS: CONTINUOUS SOLUTIONS*

DAVID COLELLA† AND CHRISTOPHER HEIL‡

Abstract. A dilation equation is a functional equation of the form $f(t) = \sum_{k=0}^{N} c_k f(2t-k)$, and any nonzero solution of such an equation is called a scaling function. Dilation equations play an important role in several fields, including interpolating subdivision schemes and wavelet theory. This paper obtains sharp bounds for the Hölder exponent of continuity of any continuous, compactly supported scaling function in terms of the joint spectral radius of two matrices determined by the coefficients $\{c_0, \ldots, c_N\}$. The arguments lead directly to a characterization of all dilation equations that have continuous, compactly supported solutions.

Key words. dilation equation, joint spectral radius, scaling function, two-scale difference equation, wavelet

AMS subject classifications. 26A16, 39A10

Functional equations of the form

$$f(t) = \sum_{k=0}^{N} c_k f(2t-k) \tag{1}$$

play an important role in several fields, including wavelet theory and interpolating subdivision schemes. Such equations are referred to as *dilation equations* or *two-scale difference equations*, and any nonzero solution f is called a *scaling function.* The coefficients $\{c_0, \ldots, c_N\}$ may be real or complex; if they are real then the scaling function f will be real-valued.

In this paper we obtain sharp bounds for the Hölder exponent of continuity of any continuous, compactly supported scaling function. Our arguments lead directly to a characterization of all dilation equations that have continuous, compactly supported solutions. These methods also enable us to examine how certain properties of scaling functions, such as the Hölder exponent, behave as a function of the coefficients, and we provide several examples to illustrate the basic structure present. Our work was inspired by an early preprint of [DL2], in which sufficient conditions for the existence of continuous, compactly supported scaling functions were obtained and lower bounds for the Hölder exponent of continuity were derived. In that paper the assumption was made that the coefficients satisfy

$$\sum_k c_{2k} = \sum_k c_{2k+1} = 1. \tag{2}$$

Conditions and bounds were then expressed in terms of the joint spectral radius $\hat{\rho}(T_0|_V, T_1|_V)$ of two matrices T_0, T_1 (determined by the coefficients $\{c_0, \ldots, c_N\}$) restricted to a certain subspace V of $\mathbf{C}^N$. We have extended these results in the

* Received by the editors January 27, 1992; accepted for publication (in revised form) July 30, 1992.

† The MITRE Corporation, McLean, Virginia 22102 (colella@mitre.org).

‡ Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 and The MITRE Corporation, McLean, Virginia 22102. Current address: School of Mathematics, Georgia Tech, Atlanta, Georgia 30332-0160 (heil@math.gatech.edu). This author's work was partially supported by National Science Foundation grant DMS-9007212.

sense that all dilation equations possessing continuous, compactly supported solutions, without restriction on the coefficients, are now described in terms of a joint spectral radius $\hat{\rho}(T_0|_W, T_1|_W)$, with the subspace V replaced by a (possibly different) subspace W, and that sharp bounds for the Hölder exponent follow from the value of $\hat{\rho}(T_0|_W, T_1|_W)$. The replacement of V by W is not without cost: W depends explicitly on the coefficients $\{c_0, \ldots, c_N\}$, while V is independent of them. This dependency nonetheless yields a number of interesting facts about the behavior of scaling functions as a function of the coefficients $\{c_0, \ldots, c_N\}$. For example, we show that the maximum Hölder exponent of continuity is not continuous as a function of the coefficients. We give several methods of determining the subspace W explicitly. The results presented here implicitly characterize those dilation equations having compactly supported, n-times differentiable solutions, and are generalizable to positive integer dilation factors other than two and to higher dimensions. An application of the results of this paper to the specific case $N \leq 3$ can be found in [CH1]. The sequel [CH2] discusses more general characterizations, including discontinuous and noncompactly supported scaling functions. Finally, we wish it noted that a revised version of [DL2] brought to our attention the work [MP2], which was the first work to provide necessary and sufficient conditions for the existence of continuous, compactly supported scaling functions. However, those results are not stated in terms of a joint spectral radius and do not yield estimates for the Hölder exponent of continuity; in addition, the behavior of the properties of scaling functions as a function of the coefficients is not examined.

We say that a function h on $\mathbf{R}$ is (globally) *Hölder continuous* if there exist constants α, K such that $|h(x) - h(y)| \leq K\,|x - y|^{\alpha}$ for all $x, y \in \mathbf{R}$. The constants α and K are referred to as a *Hölder exponent* and *Hölder constant* for h, respectively. We refer to

$$\alpha_{\max} = \sup\{\alpha : h \text{ is Hölder continuous with exponent } \alpha\}$$

as the *maximum Hölder exponent* of h, although it should be noted that the supremum need not be attained, i.e., h need not be Hölder continuous with exponent $\alpha = \alpha_{\max}$. We make similar definitions for *local Hölder continuity* at a point x, i.e., x is fixed in the inequality and only y varies.

The application of dilation equations to subdivision schemes is discussed in the important papers [CDM], [DD], [Du], [DGL], [MP1], and [MP2]. Our own interest in dilation equations arose from their application to wavelet theory. We briefly outline this relation below; the basic results can be found in the research papers [D], [L], [M], or in the expository surveys [H], [S]. Assume that the coefficients $\{c_0, \ldots, c_N\}$ satisfy (2) and

$$\sum_k c_k\,\bar{c}_{k+2j} = \begin{cases} 2, & \text{if } j = 0, \\ 0, & \text{if } j \neq 0, \end{cases} \tag{3}$$

where we take $c_k = 0$ if $k < 0$ or $k > N$. It can then be shown that there exists an integrable and square-integrable scaling function f. Define the *wavelet*

$$g(t) = \sum_k (-1)^k\,c_{N-k}\,f(2t - k), \tag{4}$$

and construct $g_{n,k}(t) = 2^{n/2}\,g(2^n t - k)$ by dilation and translation of g. If the scaling function f is orthogonal to its integer translates (i.e., $\int f(t)\,\overline{f(t-k)}\,dt = 0$ for $k \neq 0$), then $\{g_{n,k}\}_{n,k \in \mathbf{Z}}$ will form an orthonormal basis for $L^2(\mathbf{R})$ after a suitable normalization of g. It can be shown that for nearly all choices of coefficients $\{c_0, \ldots, c_N\}$

satisfying (2) and (3) the associated scaling function f is orthogonal to its integer translates and therefore determines a wavelet orthonormal basis.

Example 1. Consider the dilation equation where $N = 1$ and $c_0 = c_1 = 1$. Both (2) and (3) are satisfied, and the associated scaling function $f = \chi_{[0,1)}$ is orthogonal to its integer translates (where χ_E denotes the characteristic function of the set E). The wavelet g given by (4) is then $g = \chi_{[0,1/2)} - \chi_{[1/2,1)}$. The orthonormal basis $\{g_{n,k}\}$ determined by this g is known as the *Haar system.* □

The Haar system has the desirable property that the wavelet g is compactly supported. However, the fact that this wavelet is discontinuous severely limits its usefulness. An important problem for wavelet theory is therefore the construction of smooth, compactly supported wavelets. By (4), it suffices to construct smooth, compactly supported scaling functions. The construction of such scaling functions is also important in subdivision theory; however, in that framework (3) is irrelevant and (2) may or may not be assumed. Several authors have proved conditions for the existence of smooth, compactly supported scaling functions, with varying interpretations of "smooth," and with varying restrictions on the coefficients. As suggested by the "self-similar" nature of the dilation equation, a continuous scaling function f is often "fractal" in nature, in the sense that if it is n-times differentiable then its nth derivative is Hölder continuous with Hölder exponent strictly less than one. Daubechies and Lagarias have proved that compactly supported, infinitely differentiable scaling functions are impossible [DL1].

The methods used to prove the conditions referred to above generally fall into the following three categories or combinations thereof: Fourier transform methods (e.g., [D], [DL1], [E], [M], [V]), iterated function system methods (e.g., [W], [D], [DL1], [DD], [DGL]), and dyadic interpolation methods (e.g., [W], [DL2], [MP2]). In this paper we use the dyadic interpolation method to characterize all dilation equations that have continuous, compactly supported solutions, without any restrictions on the coefficients. However, since a number of facts that will be important to us are more easily proved using the Fourier transform technique, we briefly review that method. This technique is based on the equivalent form of the dilation equation on the Fourier transform side, namely,

$$\hat{f}(2\gamma) = m_0(\gamma)\,\hat{f}(\gamma), \tag{5}$$

where $m_0(\gamma) = (\frac{1}{2}) \sum c_k\, e^{ik\gamma}$ and $\hat{f}(\gamma) = \int f(t)\, e^{i\gamma t}\, dt$. If $m_0(0) = (\frac{1}{2}) \sum c_k = 1$ then it follows that

$$\hat{\mu}(\gamma) = \prod_{j=1}^{\infty} m_0(2^{-j}\gamma) \tag{6}$$

converges uniformly on compact sets to a continuous function and is a solution to (5). The inverse Fourier transform μ of this function is therefore a solution to (1), at least in the sense of distributions. One can show that μ has support contained in $[0, N]$, and is the only distributional solution to (1) with a continuous Fourier transform (up to multiplication by a constant). In particular, there can be at most one integrable solution to a dilation equation satisfying $\sum c_k = 2$ (up to multiplication by a constant and with uniqueness interpreted as usual as equality almost everywhere), and if one exists it will have the form (6) and have compact support. More generally, Daubechies and Lagarias have proved the following theorem.

THEOREM 1 (see [DL1]). *Let coefficients $\{c_0, \ldots, c_N\}$ be given. If there exists an integrable, compactly supported solution f to the dilation equation* (1), *then* $\operatorname{supp}(f) \subset$

$[0, N]$ *and there exists an integer* $n \geq 0$ *such that the following statements hold.*

(a) $\sum c_k = 2^{n+1}$.

(b) f *is unique up to multiplication by a constant and has Fourier transform* $\hat{f}(\gamma) = \gamma^n \prod_{j=1}^{\infty} 2^{-n} m_0(2^{-j}\gamma)$.

(c) *There is an integrable, compactly supported solution* F *to the dilation equation determined by the coefficients* $\{2^{-n}c_0, \ldots, 2^{-n}c_N\}$ *and (with the proper choice of scale)* f *is the* n*th distributional derivative of* F.

In particular, continuous, compactly supported scaling functions can exist only when $\sum c_k = 2^{n+1}$, and for $n > 0$ are the (usual) nth derivatives of scaling functions satisfying $\sum c_k = 2$. Dilation equations satisfying $\sum c_k = 2$ are therefore in some sense fundamental. Additional assumptions on the coefficients $\{c_0, \ldots, c_N\}$ can impose enough regularity on the infinite product in (6) so that the scaling function can be proved to be continuous or n-times differentiable, and to bound from below the Hölder exponent of continuity of the scaling function or its nth derivative. Eirola [E] has nicely demonstrated that Fourier transform methods are well suited to estimating Sobolev, rather than Hölder, exponents of continuity of scaling functions.

We turn now to the dyadic interpolation method and our own results. Since we are concerned mainly with questions of continuity, we assume that scaling functions, even if discontinuous, are defined for all points in $\mathbf{R}$ and satisfy the dilation equation at all points, not just almost everywhere, as is the case in the Fourier transform method. This has the seemingly paradoxical effect that a given dilation equation may have more than one distinct integrable, compactly supported solution. For example, if f is one such scaling function and if $c \in \mathbf{C}$ and $S \subset \mathbf{R}$ are given so that both S and its complement are measurable and invariant under the mapping $t \mapsto 2t - k$ for all $k \in \mathbf{Z}$, then

$$\tilde{f}(t) = \begin{cases} f(t), & t \in S, \\ c\,f(t), & t \notin S, \end{cases}$$

is also an integrable, compactly supported scaling function. However, by Theorem 1, either S or its complement must have measure zero (more fundamentally, this also follows from ergodicity considerations). This leads us to pose the following problem, which affects the interpretation of Theorem 4 at the end of this paper: Is it possible that such differing representatives of a compactly supported scaling function may have differing properties, e.g., if one representative is unbounded, must all representatives be unbounded?

The dyadic interpolation method is based on this key observation: if the values of the scaling function at the integers are known then the dilation equation determines the values of the scaling function at the half-integers, and by recursion at every *dyadic point* $x = k/2^n$ where $k, n \in \mathbf{Z}$. If f is continuous then this determines its values at all points (thereby giving an easily programmable method for graphing a continuous scaling function). Daubechies and Lagarias [DL2] and Micchelli and Prautzsch [MP2] independently implemented this recursion via products of two $N \times N$ matrices, and used this implementation to obtain conditions for the existence of continuous scaling functions. We explain this now, using the notation of [DL2].

Given coefficients $\{c_0, \ldots, c_N\}$, define the $N \times N$ matrices T_0 and T_1 by $(T_0)_{ij} =$

c_{2i-j-1} and $(T_1)_{ij} = c_{2i-j}$, i.e.,

$$T_0 = \begin{pmatrix} c_0 & 0 & 0 & \cdots & 0 & 0 \\ c_2 & c_1 & c_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & c_N & c_{N-1} \end{pmatrix}$$

and

$$T_1 = \begin{pmatrix} c_1 & c_0 & 0 & \cdots & 0 & 0 \\ c_3 & c_2 & c_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & c_N \end{pmatrix}.$$

Given $x \in [0,1]$ with $x \neq \frac{1}{2}$, define

$$\tau x = 2x \bmod 1 = \begin{cases} 2x, & 0 \leq x < \frac{1}{2}, \\ 2x - 1, & \frac{1}{2} < x \leq 1, \end{cases}$$

i.e., if $x \neq \frac{1}{2}$ and $x = .d_1 d_2 \ldots$ is a binary expansion of x then $\tau x = .d_2 d_3 \ldots$. The value of $\tau(\frac{1}{2})$ is undefined. Note that a dyadic point x has two binary expansions, one ending in infinitely many zeros, and one ending in infinitely many ones. The former expansion will be termed the *upper* or *finite* binary expansion, and the latter the *lower* binary expansion. All nondyadic points have unique binary expansions.

The following result allows questions about scaling functions to be translated into questions about certain vector-valued (specifically, $\mathbf{C}^N$-valued) functions on $[0,1]$. The definition of *Hölder continuity* for vector-valued functions is analogous to the definition for ordinary functions, with the absolute value replaced by any norm on $\mathbf{C}^N$. The Hölder exponent of continuity for a vector-valued function is independent of the choice of norm.

PROPOSITION 1. *Let coefficients* $\{c_0, \ldots, c_N\}$ *be given.*

(a) *Assume* f *is a scaling function with* $\operatorname{supp}(f) \subset [0, N]$. *Define the vector-valued function* $v : [0,1] \to \mathbf{C}^N$ *by*

$$v(x) = \begin{pmatrix} f(x) \\ f(x+1) \\ \vdots \\ f(x+N-1) \end{pmatrix}. \tag{7}$$

Then v *satisfies*

$$v_{i+1}(0) = v_i(1), \qquad i = 1, \ldots, N-1, \tag{8}$$

$$v(x) = T_0 v(\tau x), \qquad 0 < x < \frac{1}{2}, \tag{9}$$

$$v(x) = T_1 v(\tau x), \qquad \frac{1}{2} < x < 1, \tag{10}$$

where $v_i(x)$ *is the* i*th component of* $v(x)$. *If* $f(0) = f(N) = 0$ (*e.g., if* f *is continuous*)

then v also satisfies

$$v_1(0) = v_N(1) = 0, \tag{11}$$

$$v(0) = T_0 v(0), \tag{12}$$

$$v(1) = T_1 v(1), \tag{13}$$

$$v\left(\frac{1}{2}\right) = T_0 v(1) = T_1 v(0). \tag{14}$$

If f is continuous then so is v. If f is Hölder continuous with Hölder exponent α, then the same is true of v.

(b) *Assume $v : [0,1] \to \mathbf{C}^N$ is a vector-valued function satisfying* (8)–(14). *Define the function f by*

$$f(x) = \begin{cases} 0, & x \le 0 \text{ or } x \ge N, \\ v_i(x), & i-1 \le x \le i,\ i = 1, \dots, N. \end{cases} \tag{15}$$

Then f is a scaling function with $\operatorname{supp}(f) \subset [0, N]$. *If v is continuous then so is f. If v is Hölder continuous with Hölder exponent α then the same is true of f.*

A vector-valued function $v : [0,1] \to \mathbf{C}^N$ satisfying (8)–(14) will be called a *scaling vector*. If v is a scaling vector then (9), (10), and (12)–(14) can be summarily written

$$v(x) = T_{d_1} v(\tau x), \qquad 0 \le x \le 1, \tag{16}$$

where $x = .d_1 d_2 \dots$ is any binary expansion of x, since, by (14), the ambiguity at $x = \frac{1}{2}$ is nonproblematic.

Note that if f is a scaling function supported in $[0, N]$ and v is defined by (7) then (12)–(14) follow immediately from (11). Also, note from (1) that c_0, $c_N \neq 1$ implies $f(0) = f(N) = 0$. Thus, for c_0, $c_N \neq 1$ there is an exact equivalence between scaling functions and scaling vectors. If $c_0 = 1$ then it is possible that $f(0) \neq 0$, and in this case it is easy to see that (12)–(14) may fail (a specific example is the Haar system, i.e., $N = 1$, $c_0 = c_1 = 1$, and $f = \chi_{[0,1)}$). Similar remarks hold if $c_N = 1$. However, it follows directly from the dilation equation that if $|c_0| \ge 1$ then f must be discontinuous at 0, and if $|c_N| \ge 1$ then f must be discontinuous at N.

If v is a scaling vector then, by (11), $v(0) = (0, a_1, \dots, a_{N-1})^{\mathrm{t}}$ for some $a = (a_1, \dots, a_{N-1})^{\mathrm{t}}$. It follows then from (12) that $Ma = a$, where M is the $(N-1) \times (N-1)$ submatrix of T_0 and T_1 defined by $M_{ij} = c_{2i-j}$, i.e.,

$$M = \begin{pmatrix} c_1 & c_0 & 0 & \cdots & 0 & 0 \\ c_3 & c_2 & c_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & c_N & c_{N-1} \end{pmatrix}. \tag{17}$$

Thus scaling vectors can only exist when M has 1 as an eigenvalue. This is true, for example, when (2) is satisfied, for then $(1, \dots, 1)$ is a left eigenvector for M for the eigenvalue 1.

CONSTRUCTION 1. We demonstrate that if 1 is an eigenvalue of M, then it is possible to construct a vector-valued function v, *defined for dyadic $x \in [0,1]$ only*, which satisfies (8)–(14). This construction need not be unique if the eigenvalue 1 has multiplicity greater than 1. In general, it may not be possible to extend this function to a scaling vector defined at all points in $[0,1]$ so that the associated scaling function defined by (15) is integrable. Equivalently, every dilation equation such that 1 is an

eigenvalue for M can be solved on the restricted domain of the dyadic points in $\mathbf{R}$, but not necessarily for all points in $\mathbf{R}$.

To construct v, let $a = (a_1, \ldots, a_{N-1})^t$ be any right eigenvector for M for the eigenvalue 1. Define $v(0) = (0, a_1, \ldots, a_{N-1})^t$ and $v(1) = (a_1, \ldots, a_{N-1}, 0)^t$; then (8) and (11) are satisfied. Given a dyadic $x \in (0,1)$, let $x = .d_1 \ldots d_m$ be its upper binary expansion, and define

$$v(x) = T_{d_1} \cdots T_{d_m} v(0). \tag{18}$$

Then (9), (10), and (12)–(14) follow immediately for dyadic x. □

As pointed out above, the matrix M must have 1 as an eigenvalue in order that a scaling vector exist; in particular, this must be the case if a continuous, compactly supported scaling function exists. Since the derivative of a differentiable scaling function is itself a scaling function for the dilation equation determined by the coefficients $\{2c_0, \ldots, 2c_N\}$, it follows immediately that if a compactly supported scaling function is n-times differentiable, then $1, 2^{-1}, \ldots, 2^{-n}$ must all be eigenvalues for M. In particular, $n < N - 1$, and no compactly supported scaling function can be infinitely differentiable.

Example 2. Consider the case $N = 3$. If $\{c_0, c_1, c_2, c_3\}$ satisfy (2) then $c_1 = 1 - c_3$ and $c_2 = 1 - c_0$. Restricting our attention to real-valued coefficients (as we will do in all specific examples in this paper), the collection of four-coefficient dilation equations satisfying (2) can therefore be identified with the (c_0, c_3)-plane, so that each point in the plane determines a four-coefficient dilation equation and conversely. We make this identification throughout when we discuss the case $N = 3$. Despite the fact that the dilation equation can be solved at dyadic points for any coefficient choice (c_0, c_3), it is proved in [CH2] that it is impossible to construct integrable, compactly supported scaling functions for any point (c_0, c_3) on or outside the ellipse shown in Fig. 1, with the single exception of the point $(1,1)$. Conversely, integrable, compactly supported scaling functions do exist for all points in the shaded region of Fig. 1. These scaling functions are continuous for those points in the shaded region that are also inside the triangle of Fig. 1, and are differentiable for those points lying on the solid portion of the dashed line [CH1]. A scaling function for the point $(1,1)$ is $\chi_{[0,3)}$. However, the function v defined in Construction 1 for this point is highly oscillatory, e.g., $v(x)$ takes each of the values $(0,1,1)^t$, $(1,0,1)^t$, and $(1,1,0)^t$ on a dense set of dyadic x. □

Let us now consider the question of continuity of scaling vectors. Given a scaling vector v and a dyadic point $x = .d_1 \ldots d_m$, consider points $y = .d_1 \ldots d_m d_{m+1} \ldots d_n$ close to x. If v is continuous then $T_{d_1} \cdots T_{d_m}\big(v(0) - v(\tau^m y)\big) = v(x) - v(y) \to 0$ as $y \to x$. This suggests that a characterization of continuity for scaling functions requires consideration of all possible products $T_{d_1} \cdots T_{d_m}$ of T_0 and T_1 operating on all possible differences $v(x) - v(y)$. The correct tool for this turns out to be the *joint spectral radius* of T_0, T_1 restricted to a certain subspace of $\mathbf{C}^N$. We therefore digress to define the joint spectral radius of general matrices and to give some of its properties relevant to the results in this paper. Let $\|\cdot\|$ be any norm on $\mathbf{C}^N$, with corresponding operator norm $\|A\| = \sup_{u \neq 0} \|Au\|/\|u\|$ defined for $N \times N$ matrices A.

DEFINITION 1. *The joint spectral radius* $\hat{\rho}(A_0, A_1)$ *of two matrices* A_0, A_1 *is*

$$\hat{\rho}(A_0, A_1) = \limsup_{m \to \infty} \hat{\rho}_m,$$

where

$$\hat{\rho}_m = \hat{\rho}_m(A_0, A_1) = \max_{d_j = 0,1} \|A_{d_1} \cdots A_{d_m}\|^{1/m}.$$

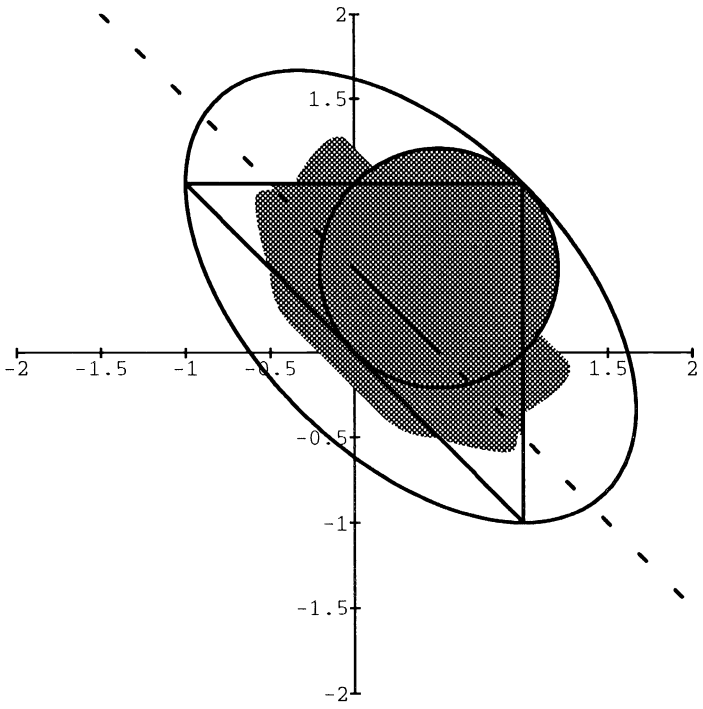


FIG. 1. *The (c_0, c_3)-plane, identified with real-valued, four-coefficient dilation equations satisfying* (2).

This definition generalizes the usual spectral radius $\rho(A)$ of a single matrix A, which is given by the formula

$$\rho(A) = \limsup_{m\to\infty} \|A^m\|^{1/m} = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}.$$

An extension of the joint spectral radius to larger collections of matrices, and to matrices restricted to subspaces, is made in the obvious way. The joint spectral radius was introduced by Rota and Strang [RS]; some recent papers include [BW] and [DL3].

The joint spectral radius is independent of the choice of norm $\|\cdot\|$, and of the choice of basis, that is, $\hat{\rho}(A_0, A_1) = \hat{\rho}(B_0, B_1)$ whenever $B_i = BA_iB^{-1}$ for any fixed invertible matrix B. Set

$$\hat{\sigma}_m = \hat{\sigma}_m(A_0, A_1) = \max_{d_j=0,1} \rho(A_{d_1} \cdots A_{d_m})^{1/m},$$

so that $\hat{\sigma}_m$ is the mth root of the largest absolute eigenvalue that occurs among all products of length m of the matrices A_0, A_1. Then $\hat{\sigma}_m \le \hat{\rho}(A_0, A_1) \le \hat{\rho}_m$ for every m. In particular, $\hat{\rho}(A_0, A_1) \ge \hat{\sigma}_1$ and generally one expects $\hat{\rho}(A_0, A_1)$ to be strictly larger than $\hat{\sigma}_1$. Wang [W] proves the nontrivial result that $\hat{\rho}(A_0, A_1) = \limsup \hat{\sigma}_m$. It therefore follows that $\sup \hat{\sigma}_m = \hat{\rho}(A_0, A_1) = \lim \hat{\rho}_m = \inf \hat{\rho}_m$. As each $\hat{\sigma}_m$ is a continuous function of the entries of A_0 and A_1, $\sup \hat{\sigma}_m$ is lower semicontinuous as a function of those entries. Similarly, $\inf \hat{\rho}_m$ is upper semicontinuous, and therefore $\hat{\rho}(A_0, A_1)$ is continuous as a function of the entries of A_0 and A_1 [HS].

The exact evaluation of the joint spectral radius is difficult, except in special cases such as the following.

LEMMA 1. *If A_0 and A_1 can be simultaneously symmetrized, i.e., there exists an invertible matrix B such that BA_iB^{-1} is real and symmetric for $i = 0, 1$, then* $\hat{\rho}(A_0, A_1) = \hat{\sigma}_1 = \max\{\rho(A_0), \rho(A_1)\}$.

The conclusion of Lemma 1 holds more generally when A_0, A_1 can be simultaneously Hermitianized.

It is easy to see that given any $\theta > \hat{\rho}(A_0, A_1)$, there exists a constant $C > 0$ such that $\hat{\rho}_m \leq C^{1/m}\,\theta$ for every m. This need not be true with $\theta = \hat{\rho}(A_0, A_1)$. One case where it is true with $\theta = \hat{\rho}(A_0, A_1)$ is if A_0, A_1 are simultaneously symmetrizable, for then $\hat{\sigma}_1 = \hat{\rho}_1$ (using the Euclidean space norm), and therefore $\hat{\rho}_m \leq \hat{\rho}_1 = \hat{\sigma}_1 \leq \hat{\rho}(A_0, A_1)$ for every m.

Another case in which some simplification of the joint spectral radius occurs is the following.

LEMMA 2. *If A_0, A_1 can be simultaneously block upper-triangularized, i.e., there exists an invertible matrix B such that BA_iB^{-1} has the block form*

$$BA_iB^{-1} = \begin{pmatrix} C_i^1 & & * \\ & \ddots & \\ 0 & & C_i^k \end{pmatrix}, \qquad i = 0, 1,$$

for some square submatrices $C_i^1, \ldots, C_i^k$, then $\hat{\rho}(A_0, A_1) = \max_{j=1,\ldots,k}\{\hat{\rho}(C_0^j, C_1^j)\}$.

In particular, if each block C_i^j is a single number—meaning that A_0, A_1 can be simultaneously upper-triangularized—then $\hat{\rho}(A_0, A_1) = \hat{\sigma}_1 = \max\{\rho(A_0), \rho(A_1)\}$. The proof of Lemma 2 follows easily from block matrix multiplication and consideration of the eigenvalues of the products $BA_{d_1} \cdots A_{d_m}B^{-1}$.

We return now to the matrices T_0, T_1 and the scaling vector v. Assume that (2) holds; then $(1, \ldots, 1)$ is a common left eigenvector for both T_0 and T_1. Hence the $(N-1)$-dimensional subspace

$$V = \{u \in \mathbf{C}^N : u_1 + \cdots + u_N = 0\}$$

is invariant under both of those operators since it is the orthogonal complement of $(1, \ldots, 1)$ in $\mathbf{C}^N$. By (18), V contains every difference $v(x) - v(y)$ for dyadic x, $y \in [0, 1]$. This observation suggests consideration of the joint spectral radius $\hat{\rho}(T_0|_V, T_1|_V)$ of T_0, T_1 restricted to the subspace V. Note that since V is invariant under both T_0 and T_1, there exists a change-of-basis matrix B such that

$$BT_iB^{-1} = \begin{pmatrix} S_i & * \\ 0 & 1 \end{pmatrix}, \qquad i = 0, 1, \tag{19}$$

where S_0, S_1 are $(N-1)\times(N-1)$ matrices, and $\hat{\rho}(T_0|_V, T_1|_V) = \hat{\rho}(S_0, S_1)$. A sufficient condition for continuity and a lower bound for the maximum Hölder exponent of continuity is provided by the following theorem of Daubechies and Lagarias.

THEOREM 2 (see [DL2]). *Assume that coefficients $\{c_0, \ldots, c_N\}$ satisfying (2) are given. If $\hat{\rho}(T_0|_V, T_1|_V) < 1$ then a continuous scaling vector v exists, and is Hölder continuous with $\alpha_{\max} \geq -\log_2 \hat{\rho}(T_0|_V, T_1|_V)$. The exponent $\alpha = -\log_2 \hat{\rho}(T_0|_V, T_1|_V)$ is allowed if there exists a constant $C > 0$ such that $\hat{\rho}_m \leq C^{1/m}\,\hat{\rho}(T_0|_V, T_1|_V)$ for every m.*

We emphasize that the value $-\log_2 \hat{\rho}(T_0|_V, T_1|_V)$ in Theorem 2 is only a lower bound for the maximum Hölder exponent, i.e., the theorem does not imply that v

cannot be Hölder continuous for exponents $\alpha > -\log_2 \hat{\rho}(T_0|_V, T_1|_V)$.

Note that if (2) does not hold, then V need not be invariant under both T_0 and T_1, and therefore $\hat{\rho}(T_0|_V, T_1|_V)$ need not be well defined. The necessary and sufficient conditions for the existence of continuous, compactly supported scaling functions presented below do not depend on (2).

The problem of characterizing all choices of coefficients $\{c_0, \ldots, c_N\}$ for which $\hat{\rho}(T_0|_V, T_1|_V) < 1$ is difficult. Of course, $\hat{\rho}(T_0|_V, T_1|_V) \leq \hat{\rho}_m$ for each m and $\hat{\rho}_m$ depends on only finitely many matrices; however, the number of matrices involved grows exponentially with m and therefore $\hat{\rho}_m$ can be computed in practice only for small m. A recursive algorithm, based on the *building blocks* idea of [DL2], for bounding a joint spectral radius from above with minimal computation is given in [CH1].

Theorem 2 implies that continuous scaling vectors exist for every coefficient choice in the set

$$C_V = \{\{c_0, \ldots, c_N\} \text{ satisfying (2)} : \hat{\rho}(T_0|_V, T_1|_V) < 1\}.$$

This is an open set since $\hat{\rho}(T_0|_V, T_1|_V)$ is a continuous function of the coefficients $\{c_0, \ldots, c_N\}$. The lower bound $-\log_2 \hat{\rho}(T_0|_V, T_1|_V)$ for the maximum Hölder exponent given in Theorem 2 is also a continuous function of the coefficients. However, we show in Example 7 that $\alpha_{\max}$ itself is *not* continuous as a function of the coefficients, even for coefficients in C_V. Despite this, we can prove that if (2) holds then the corresponding scaling functions change in a continuous manner (with respect to the sup-norm) as the coefficients change within C_V. This result takes advantage of the fact that V is independent of the coefficients.

PROPOSITION 2. *Fix $\{c_0, \ldots, c_N\} \in C_V$ and let f be the corresponding scaling function. For each $\varepsilon > 0$ there exists a $\delta > 0$ such that if $\{\tilde{c}_0, \ldots, \tilde{c}_N\}$ satisfies (2) and $|c_i - \tilde{c}_i| < \delta$ for $i = 0, \ldots, N$ then there is a continuous, compactly supported scaling function $\tilde{f}$ corresponding to $\{\tilde{c}_0, \ldots, \tilde{c}_N\}$ with $\sup |f(t) - \tilde{f}(t)| < \varepsilon$.*

Proof. For each $\eta > 0$ set

$$U_\eta = \{\{\tilde{c}_0, \ldots, \tilde{c}_N\} \text{ satisfying (2)} : |c_i - \tilde{c}_i| \leq \eta,\ i = 0, \ldots, N\}.$$

Since $\{c_0, \ldots, c_N\} \in C_V$, there must exist an integer m such that $\hat{\rho}(T_0|_V, T_1|_V) \leq \hat{\rho}_m < 1$. As $\hat{\rho}_m$ depends on only finitely many matrix products, there must be $\theta, \eta > 0$ such that if $\{\tilde{c}_0, \ldots, \tilde{c}_N\} \in U_\eta$ then $\hat{\rho}(\tilde{T}_0|_V, \tilde{T}_1|_V) \leq \tilde{\rho}_m < \theta < 1$, where $\tilde{T}_0$, $\tilde{T}_1$ are the matrices corresponding to $\{\tilde{c}_0, \ldots, \tilde{c}_N\}$ and $\tilde{\rho}_m = \max_{d_j=0,1} \|(\tilde{T}_{d_1} \cdots \tilde{T}_{d_m})|_V\|^{1/m}$. Therefore, for each $\{\tilde{c}_0, \ldots, \tilde{c}_N\} \in U_\eta$, there exists a corresponding continuous scaling vector $\tilde{v}$ that is Hölder continuous with exponent $\tilde{\alpha}$ satisfying

$$\tilde{\alpha} \geq \alpha = -\log_2 \theta.$$

Careful examination of the proof of Theorem 2 given in [DL2] reveals that corresponding Hölder constants $\tilde{K}$ for $\tilde{v}$ satisfy

$$\tilde{K} \leq \frac{2\tilde{C}\tilde{R}}{\tilde{\rho}_m},$$

where

$$\tilde{C} = \max_{r=0,\ldots,m-1} \left\{ \left(\frac{\tilde{\rho}_r}{\tilde{\rho}_m} \right)^r \right\}$$

and

$$\tilde{R} = \sup_{t \in [0,1]} \|\tilde{v}(t)\| \leq \left(\frac{\tilde{C}\,(\max\{\|\tilde{T}_0\|, \|\tilde{T}_1\|\} + 1)}{1 - \tilde{\rho}_m} + 1 \right) \|\tilde{v}(0)\|.$$

The quantities $\tilde{C}$ and $\tilde{R}$ are continuous as functions of the coefficients, so $K = \sup_{\{\tilde{c}_0,\ldots,\tilde{c}_N\}\in U_\eta} \tilde{K} < \infty$.

Now let ε be given and choose n large enough that $2^{-n\alpha}K < \frac{\varepsilon}{4}$. Let S_n be the set of all dyadic points in $[0,1]$ with finite binary expansions of length n or less. The functions $\tilde{v}$ are completely determined on S_n by $\tilde{v}(0)$ and the 2^n matrices of the form $\tilde{T}_{d_1}\cdots\tilde{T}_{d_n}$. Hence, there exists a $\delta < \eta$ such that $\sup_{x\in S_n} \|v(x) - \tilde{v}(x)\| < \frac{\varepsilon}{2}$ for all $\{\tilde{c}_0,\ldots,\tilde{c}_N\} \in U_\delta$. Given $y \in [0,1]$ arbitrary, there is an $x \in S_n$ with $|y-x| < 2^{-n}$, so

$$\begin{aligned}\|v(y) - \tilde{v}(y)\| &\le \|v(y) - v(x)\| + \|v(x) - \tilde{v}(x)\| + \|\tilde{v}(x) - \tilde{v}(y)\| \\ &\le K\,|y-x|^\alpha + \frac{\varepsilon}{2} + K\,|x-y|^\alpha \\ &< \varepsilon.\end{aligned}$$

The result now follows once we recall that all norms on $\mathbf{C}^N$ are equivalent, and note that the sup-norm for scaling vectors is equivalent to the sup-norm for the associated scaling functions. □

Theorem 2 states that the condition $\hat{\rho}(T_0|_V, T_1|_V) < 1$ implies the existence of a continuous, compactly supported scaling function. The following example, inspired by [W], shows that $\hat{\rho}(T_0|_V, T_1|_V) < 1$ is not, in general, necessary for the existence of a continuous, compactly supported scaling function.

Example 3. Fix N, choose coefficients $\{c_0,\ldots,c_N\}$ satisfying (2), and let f be the associated scaling function. Let $d > 1$ be any odd integer, set $\tilde{N} = Nd$, and define coefficients $\{\tilde{c}_0,\ldots,\tilde{c}_{\tilde{N}}\}$ by

$$\tilde{c}_k = \begin{cases} c_j, & \text{if } k = jd, \\ 0, & \text{if } k \ne jd. \end{cases}$$

Let $\tilde{T}_0$, $\tilde{T}_1$ be the matrices corresponding to $\{\tilde{c}_0,\ldots,\tilde{c}_{\tilde{N}}\}$, and let $\tilde{V}$ be the corresponding subspace of $\mathbf{C}^{\tilde{N}}$. By construction, the coefficients $\{\tilde{c}_0,\ldots,\tilde{c}_{\tilde{N}}\}$ satisfy (2), and the associated scaling function is $\tilde{f}(t) = f(\frac{t}{d})$. It must be the case that $\hat{\rho}(\tilde{T}_0|_{\tilde{V}}, \tilde{T}_1|_{\tilde{V}}) \ge \rho(\tilde{T}_0|_{\tilde{V}}) \ge 1$ since both $(1,1,\ldots,1)$ and $(1,0,0,1,0,0,\ldots,1,0,0)$ are left eigenvectors for $\tilde{T}_0$ and $\tilde{T}_1$ for the eigenvalue 1 and $\dim(\tilde{V}) = \tilde{N} - 1$. However, if $\hat{\rho}(T_0|_V, T_1|_V) < 1$ then f, and therefore $\tilde{f}$, will be continuous. The dilation equation defined by the coefficients $\{\tilde{c}_0,\ldots,\tilde{c}_{\tilde{N}}\}$ is referred to as a *stretched dilation equation.* Wang examined stretched dilation equations using iterated function system techniques. □

For a stretched dilation equation, the differences $v(x) - v(y)$ are contained in a lower-dimensional subspace of V, and therefore the value of $\hat{\rho}(T_0|_V, T_1|_V)$ may be determined by vectors not directly related to the scaling vector v. This suggests that better results might be obtained by replacing V with the subspace

$$\begin{aligned} W &= \operatorname{span}\{v(x) - v(y) : \text{dyadic } x, y \in [0,1]\} \\ &= \operatorname{span}\{v(x) - v(0) : \text{dyadic } x \in [0,1]\}. \end{aligned}$$

Note that W is invariant under T_0, T_1 by construction, without the need to assume (2). If (2) does hold then $W \subset V$. Note also that W depends explicitly on the scaling vector v, while V does not. We show in Theorem 3 below that $\hat{\rho}(T_0|_W, T_1|_W) < 1$ is sufficient to ensure the existence of a continuous scaling function, and we give several examples illustrating the distinction between V and W and its consequences. Moreover, by considering W instead of V, we obtain necessary conditions for the

existence of continuous scaling functions. These necessary conditions can be illustrated by the following example.

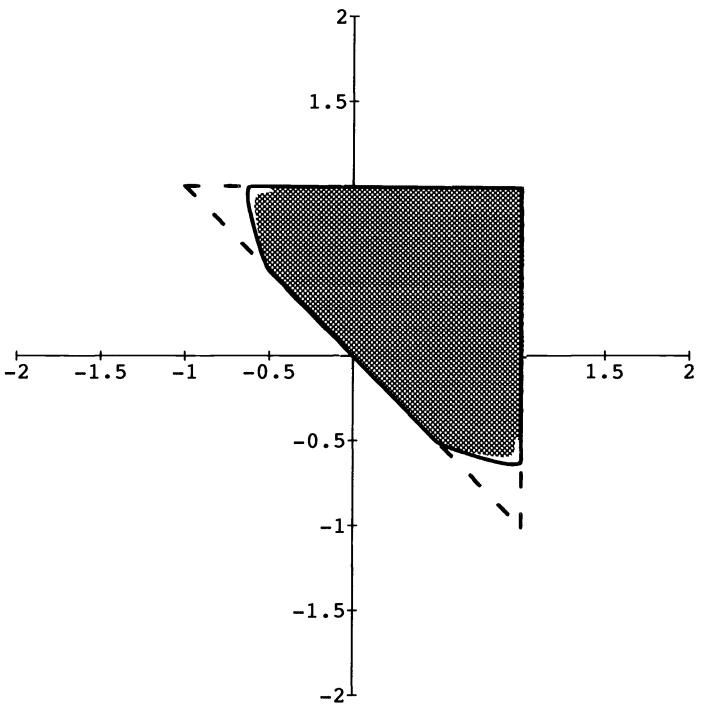


FIG. 2. *Subsets of the* (c_0, c_3)*-plane: region where either* $\hat{\rho}_{20} < 1$ *or* $\hat{\rho}(T_0|_V, T_1|_V) < 1$ *due to simultaneous symmetrization (shaded area); boundary of the region where* $\hat{\sigma}_{20} < 1$ *(solid curve).*

Example 4. Consider $N = 3$ as in Example 2. The shaded region in Fig. 2 is the union of the set of points (c_0, c_3) where $\hat{\rho}_{20}(T_0|_V, T_1|_V) < 1$ (obtained by numerical computation using the Euclidean norm) and the set of points (c_0, c_3) such that S_0, S_1 defined by (19) are simultaneously symmetrizable with $\hat{\rho}(T_0|_V, T_1|_V) = \hat{\rho}(S_0, S_1) < 1$. Theorem 2 therefore implies that continuous, compactly supported scaling functions exist for each point in the shaded region.

To derive a necessary condition, consider the behavior of the scaling vector v near the point 0. If v is continuous, then necessarily $v(2^{-m}) \to v(0)$ as $m \to \infty$. It is straightforward to check by evaluating $v(2^{-m}) = T_0^m v(1)$ directly that this occurs if and only if $|c_0|$, $|1 - c_0 - c_3| < 1$. Similarly, the continuity of v at 1 implies $|c_3|$, $|1 - c_0 - c_3| < 1$. The eigenvalues of $T_0|_V$, $T_1|_V$ combined are c_0, c_3, and $1 - c_0 - c_3$, so if v is continuous then $\hat{\sigma}_1(T_0|_V, T_1|_V) < 1$. Moreover, this analysis shows that $\|v(2^{-m}) - v(0)\|$ is on the order of $\hat{\sigma}_1^m$, so the maximum Hölder exponent of continuity of v is at most $-\log_2 \hat{\sigma}_1$. The region of points (c_0, c_3) such that $\hat{\sigma}_1 < 1$ is the interior of the triangle shown in Fig. 2, i.e, continuous scaling vectors are restricted to points in the interior of that triangle. □

The case $N = 3$ is simple enough that the result of Example 4 can be obtained directly by evaluating $v(2^{-m})$ as a function of the coefficients $\{c_0, \ldots, c_N\}$. Although this exact evaluation cannot be done at arbitrary points, the spirit of the approach of Example 4 is used in the proof of Theorem 3(b) below to obtain that continuity

implies $\hat{\sigma}_m(T_0|_W, T_1|_W) < 1$ for every m. Specifically, the matrices T_0, T_1 are replaced by products $T = T_{d_1} \cdots T_{d_m}$, the eigenvalues c_0, c_3, $1 - c_0 - c_3$ of $T_0|_V$, $T_1|_V$ by eigenvalues of $T|_W$, and the points 2^{-m}, 0 by points X_m, Y_m depending on the product T. We must replace V by W to ensure that $v(X_m) - v(Y_m)$ will have a component in the required eigenspace of T. This was unnecessary for the case $N = 3$ since the dilation equations for that case are very restrictive, cf. Example 5. For $N = 3$, the set of points (c_0, c_3) such that $\hat{\sigma}_{20} < 1$ is the interior of the solid curve shown in Fig. 2, i.e., continuous scaling vectors are restricted not only to the interior of the triangle, but to the interior of the region bounded by the solid curve.

Since $\hat{\rho}(T_0|_W, T_1|_W) = \sup \hat{\sigma}_m$, the preceding remarks suggest that the value of $\hat{\rho}(T_0|_W, T_1|_W)$ is essentially the single determining factor for the existence of a continuous scaling function. This is made precise in the following theorem. Recall that scaling vectors can exist only when the matrix M defined by (17) has 1 as an eigenvalue.

THEOREM 3. *Assume the coefficients* $\{c_0, \ldots, c_N\}$ *are such that* 1 *is an eigenvalue for* M.

(a) *Let* v *be the vector-valued function defined for dyadic* x *constructed in Construction* 1. *If* $\hat{\rho}(T_0|_W, T_1|_W) < 1$ *then* v *extends to a continuous scaling vector.*

(b) *If* v *is any continuous scaling vector then* $\hat{\rho}(T_0|_W, T_1|_W) < 1$. *In this case* v *is Hölder continuous with* $\alpha_{\max} = -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$, *and the exponent* $\alpha = \alpha_{\max}$ *is achieved if and only if there exists a constant* $C > 0$ *such that* $\hat{\rho}_m \le C^{1/m} \hat{\rho}(T_0|_W, T_1|_W)$ *for every* m.

The proof of Theorem 3 is given below; we first make some remarks on its statement, consider how to determine W, and examine the relationship between V and W for those dilation equations satisfying (2).

First, note that the subspace W depends explicitly on v. Thus, in the hypothesis of Theorem 3(a), the vector-valued function of Construction 1 must be determined before W and $\hat{\rho}(T_0|_W, T_1|_W)$ can be evaluated.

Second, in the hypotheses of Theorem 3(b), note that if v is a continuous scaling vector, then it agrees at dyadic points with the vector-valued function given in Construction 1. Thus

$$C_W = \{\{c_0, \ldots, c_N\} \text{ satisfying (2)} : \hat{\rho}(T_0|_W, T_1|_W) < 1\}$$

is the precise set of all dilation equations satisfying (2) that have continuous, compactly supported solutions, cf. the definition of C_V following Theorem 2. Example 7 below presents a specific case where $C_W \neq C_V$ for reasons other than trivial cases such as stretched dilation equations.

Finally, note that since the derivative of a differentiable scaling function is itself a solution of another dilation equation (with coefficients multiplied by two), Theorem 3 implicitly characterizes those dilation equations having compactly supported, n-times differentiable solutions. This higher-order characterization, and the fact that computation of the joint spectral radius can often be simplified when the coefficients satisfy $\sum c_k = 2$ and the *sum rules* $\sum (-1)^k k^j c_k = 0$ for $j = 0, \ldots, n$, is elaborated on in [HC].

We now discuss several methods of determining the subspace W explicitly. Note that W is one subspace that is invariant under both T_0 and T_1. The following proposition provides a means of recognizing which invariant subspace is W by examination of the single vector $v(1) - v(0)$. This vector is easily computable since $v(0) = (0, a_1, \ldots, a_{N-1})^t$ and $v(1) = (a_1, \ldots, a_{N-1}, 0)^t$ for some eigenvector $a =$

$(a_1, \dots, a_{N-1})^t$ of the matrix M for the eigenvalue 1. If the multiplicity of this eigenvalue is one, then $v(0)$ and $v(1)$ are uniquely determined up to multiplication by a constant.

PROPOSITION 3. *W is the smallest subspace of $\mathbf{C}^N$ invariant under both T_0 and T_1 which contains the vector $v(1) - v(0)$.*

Proof. Let U be the smallest subspace of $\mathbf{C}^N$ invariant under both T_0 and T_1 which contains the vector $v(1) - v(0)$. Then $U \subset W$, and $v(1) = v(0) + u$ for some $u \in U$. Since $T_0 v(1) = T_1 v(0) = v(\frac{1}{2})$ we have $v(\frac{1}{2}) - v(0) = T_0 v(1) - v(0) = T_0 u \in U$. Continuing in this manner we obtain that if $x = .d_1 \dots d_m \in [0,1]$ is dyadic, then $v(x) - v(0) = T_{d_1} \cdots T_{d_m} v(0) - v(0) \in U$, whence $U \supset W$. □

The following corollary gives sufficient conditions for W to be all of V, assuming that (2) is satisfied. A *hyperplane* in $\mathbf{C}^N$ is any translate of a subspace of $\mathbf{C}^N$.

COROLLARY 1. *Assume (2) holds. If $\{v(x) : \text{dyadic } x \in [0,1]\}$ is not contained in any hyperplane in $\mathbf{C}^N$ of dimension $N-2$ or less, then $W = V$. In particular, if $\operatorname{span}\{v(x) : \text{dyadic } x \in [0,1]\} = \mathbf{C}^N$, then $W = V$.*

Proof. It follows from the hypotheses that $\{v(x) - v(0) : \text{dyadic } x \in [0,1]\}$ is not contained in any hyperplane of dimension $N-2$ or less. Since this set is contained in the $(N-1)$-dimensional subspace V, it follows that V is its span, and therefore is W. □

If the components of $v(0)$ add to zero then $v(x) \in V$ for every dyadic x and therefore $\operatorname{span}\{v(x) : \text{dyadic } x \in [0,1]\} \subset V \neq \mathbf{C}^N$, yet it may still be the case that $\{v(x) : \text{dyadic } x \in [0,1]\}$ is not contained in any hyperplane of dimension $N-2$ or less. If the components of $v(0)$ do not add to zero then $\operatorname{span}\{v(x) : \text{dyadic } x \in [0,1]\} = \mathbf{C}^N$ if and only if $\{v(x) : \text{dyadic } x \in [0,1]\}$ is not contained in any hyperplane of dimension $N-2$ or less. Lawton [L] proved that if (2) holds then an integrable, compactly supported scaling function f satisfies $\sum f(t+k) = \int f = \hat{f}(0)$ almost everywhere. Since f must also satisfy $\hat{f}(\gamma) = \hat{f}(0) \prod_1^\infty m_0(\gamma/2^j)$, cf. (6), it must be the case that $\hat{f}(0) \neq 0$, and thus $\sum_{i=1}^N v_i(x) = \hat{f}(0) \neq 0$ almost everywhere. In particular, if (2) holds and the components of $v(0)$ add to zero then v cannot be continuous.

Corollary 1 is easily implemented only for small values of N. In general, the hypotheses of the following proposition are easier to check.

PROPOSITION 4. *Assume (2) holds, and let $T = T_{d_1} \cdots T_{d_m}$ be any product of T_0, T_1 such that*

(a) *T has distinct eigenvalues, and*

(b) *there is some dyadic $z \in [0,1]$ such that $v(z)$ has a component in each of the eigenspaces of T.*

Let $z = .z_1 z_2 \dots$ be any binary expansion of z, and define $x_0, \dots, x_{N-1}$ by $x_0 = z$, $x_1 = .d_1 \dots d_m z_1 z_2 \dots$, $x_2 = .d_1 \dots d_m d_1 \dots d_m z_1 z_2 \dots$, etc. Then $\{v(x_0), \dots, v(x_{N-1})\}$ forms a basis for $\mathbf{C}^N$, and therefore $W = V$.

Proof. Note that $v(x_i) = T^i v(z)$ for $i = 0, \dots, N-1$. Let $\lambda_1, \dots, \lambda_N$ be the distinct eigenvalues of T; then, by hypothesis, $v(z) = u_1 + \cdots + u_N$ where each u_i is a nonzero eigenvector for T corresponding to the eigenvalue λ_i. Assume that $\{v(z), \dots, T^{N-1} v(z)\}$ was not a basis for $\mathbf{C}^N$; then there exist scalars $\alpha_1, \dots, \alpha_N$, not all zero, such that

$$0 = \alpha_1 v(z) + \alpha_2 T v(z) + \cdots + \alpha_N T^{N-1} v(z)$$

$$
\begin{aligned}
&= \alpha_1 u_1 + \cdots + \alpha_1 u_N \\
&\quad + \alpha_2 \lambda_1 u_1 + \cdots + \alpha_2 \lambda_N u_N \\
&\quad \vdots \\
&\quad + \alpha_N \lambda_1^{N-1} u_1 + \cdots + \alpha_N \lambda_N^{N-1} u_N.
\end{aligned}
$$

As $\{u_1, \ldots, u_N\}$ forms a basis for $\mathbf{C}^N$ we must therefore have $\alpha_1 + \alpha_2 \lambda_i + \cdots + \alpha_N \lambda_i^{N-1} = 0$ for $i = 1, \ldots, N$, i.e., $\lambda_1, \ldots, \lambda_N$ are roots of the polynomial $p(\lambda) = \alpha_1 + \alpha_2 \lambda + \cdots + \alpha_N \lambda^{N-1}$. As p has degree at most $N-1$, this is therefore a contradiction. □

Typically, the product T and the value of z in the hypotheses of Proposition 4 are chosen so as to simplify computations, as in the following example.

Example 5. Fix $N = 3$ as in Example 2 and consider Proposition 4 with $T = T_0$ and $z = 1$. The eigenvalues of T_0 are 1, c_0, and $1 - c_0 - c_3$; assume these are distinct. Corresponding eigenvectors of T_0 are $u_1 = v(0) = (0, c_0, c_3)^t$, $u_2 = (1 - 2c_0 - c_3, 2c_0 - 1, c_3)^t$, and $u_3 = (0, 1, -1)^t$. Up to multiplication by a constant, $v(1) = (c_0, c_3, 0)^t$. Since $1 - 2c_0 - c_3 \neq 0$,

$$
v(1) = u_1 + \frac{c_0}{1 - 2c_0 - c_3} u_2 + \frac{c_3 (1 - c_0 - c_3)}{1 - 2c_0 - c_3} u_3.
$$

Thus $v(1)$ has a component in each of the eigenspaces of T_0, and therefore $W = V$, provided that c_0, c_3, $1 - c_0 - c_3 \neq 0$.

Compare the above computation to the result proved in [CH1], that $\{v(x) : \text{dyadic } x \in [0,1]\}$ is not contained in any line in $\mathbf{C}^3$ if c_0, c_3, $1 - c_0 - c_3 \neq 0$; hence W is all of V in that case. When $1 - c_0 - c_3 = 0$, then W has dimension 1 and hence is a proper subset of V. This case is not the result of a stretched dilation equation, but rather arises from the presence of zero eigenvalues. In particular, the eigenvalues of $T_0|_V$ are c_0 and $1 - c_0 - c_3$ and the eigenvalues of $T_1|_V$ are c_3 and $1 - c_0 - c_3$. If $1 - c_0 - c_3 = 0$ then zero is a common eigenvalue of $T_0|_V$ and $T_1|_V$; moreover, the corresponding eigenspaces are identical. Note, however, that $\hat{\rho}(T_0|_V, T_1|_V) = \hat{\rho}(T_0|_W, T_1|_W)$ despite the fact that $V \neq W$, since the zero eigenvalue does not impact the value of the joint spectral radius. □

The next example is another simple illustration of how W may differ from V without the joint spectral radius being affected.

Example 6. Given coefficients $\{c_0, \ldots, c_N\}$ satisfying (2), define $\tilde{c}_i = c_i$ for $i = 0, \ldots, N$ and $\tilde{c}_{N+1} = 0$. Let f, v, W, etc., be the usual items associated with $\{c_0, \ldots, c_N\}$, and let $\tilde{f}$, $\tilde{v}$, $\tilde{W}$, etc., be the corresponding items associated with $\{\tilde{c}_0, \ldots, \tilde{c}_{N+1}\}$. Clearly $\tilde{f} = f$, yet $\tilde{W} \neq \tilde{V}$, even if $W = V$, since $\tilde{W} \subset \{u \in \mathbf{C}^{N+1} : u_1 + \cdots + u_N = 0, u_{N+1} = 0\} \neq \tilde{V}$. Wang [W] demonstrates that $\hat{\rho}(T_0|_V, T_1|_V) = \hat{\rho}(\tilde{T}_0|_{\tilde{V}}, \tilde{T}_1|_{\tilde{V}})$, and it follows similarly that $\hat{\rho}(T_0|_W, T_1|_W) = \hat{\rho}(\tilde{T}_0|_{\tilde{W}}, \tilde{T}_1|_{\tilde{W}})$. If $W = V$ then these four numbers are equal, despite the fact that $\tilde{W} \neq \tilde{V}$. □

The calculations in the preceding example are carried out by placing T_0 and T_1 in block upper-triangular form, as follows. Suppose we are given any dilation equation satisfying (2) such that W is a proper subspace of V, say $J = \dim(W) < N - 1$. Since both W and V are invariant under T_0 and T_1, there will be a change-of-basis matrix B such that BT_0B^{-1} and BT_1B^{-1} have the block upper-triangular forms

$$BT_iB^{-1} = \begin{pmatrix} P_i & * & * \\ 0 & Q_i & * \\ 0 & 0 & 1 \end{pmatrix}, \qquad i = 0, 1,$$

where the matrices P_i are $J \times J$ and the matrices Q_i are $(N-J-1) \times (N-J-1)$. Hence,

$$\hat{\rho}(T_0|_W, T_1|_W) = \hat{\rho}(P_0, P_1)$$

and

$$\hat{\rho}(T_0|_V, T_1|_V) = \max\{\hat{\rho}(P_0, P_1), \hat{\rho}(Q_0, Q_1)\}.$$

One such basis choice is the eigenvector basis for T_0 (if it exists), in which case T_0 is diagonalized and T_1 can be block upper-triangularized.

Our next example presents a class of dilation equations for which the distinction between V and W plays an important role in determining the continuity of the associated scaling vectors. In particular, this example illustrates the fact that the maximum Hölder exponent is not continuous as a function of the coefficients $\{c_0, \ldots, c_N\}$.

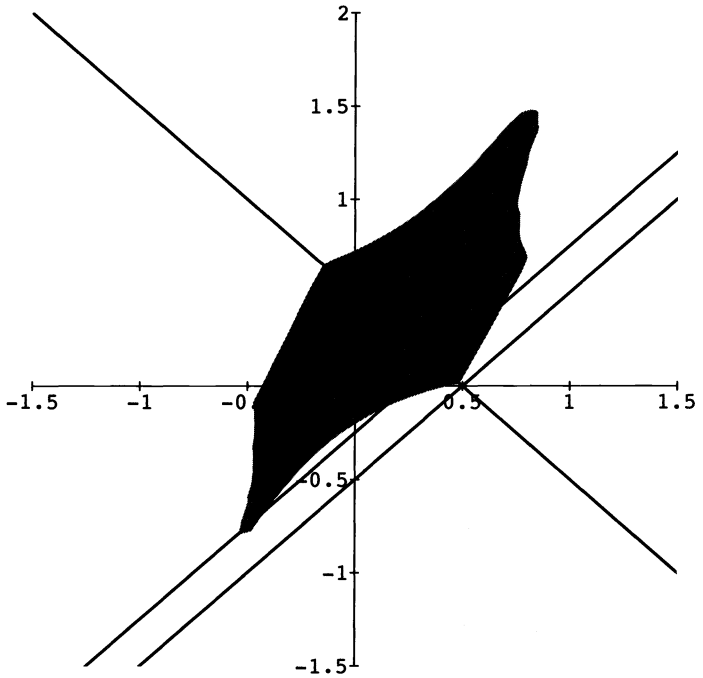


FIG. 3. *The (c_0, c_1)-plane, identified with symmetric, real-valued, seven-coefficient dilation equations satisfying* (2).

Example 7. Set $N = 6$ and consider the two-parameter family of all symmetric sets $\{c_0, c_1, \frac{1}{2} - c_0, 1 - 2c_1, \frac{1}{2} - c_0, c_1, c_0\}$ of real coefficients which satisfy (2). We identify these dilation equations with the (c_0, c_1)-plane. The shaded region in Fig. 3 is a numerical approximation of the region $\{(c_0, c_1) : \hat{\rho}_{16}(T_0|_V, T_1|_V) < 1\}$ (using the norm $\|u\| = |u_1| + \cdots + |u_6|$) and is therefore a subset of C_V, i.e., all points in this region give rise to continuous, compactly supported scaling functions. We discuss

several subfamilies of the (c_0, c_1)-plane below; in particular, the three lines shown in Fig. 3 are discussed in parts (b), (c), and (d).

(a) The point $(\frac{1}{2}, 0)$ corresponds to a stretched dilation equation, and the associated scaling vector is $v(x) = (x, x+1, x+2, 3-x, 2-x, 1-x)^t$. Therefore W is a line in $\mathbf{C}^6$ and hence is a proper subset of V. Although $\hat{\rho}(T_0|_V, T_1|_V) = 1$ we have $\hat{\rho}(T_0|_W, T_1|_W) = \frac{1}{2}$, and v is Hölder continuous with Hölder exponent 1. In particular, C_V is a proper subset of C_W.

(b) Consider now those dilation equations in the (c_0, c_1)-plane lying on the line $(c_0, c_1) = (\frac{1}{2} + \delta, -\delta)$, $\delta \in \mathbf{R}$. When $\delta = 0$ this is the stretched dilation equation discussed in part (a). When $\delta \neq 0$, Proposition 3 can be used to show that $\dim(W) = 3$. Furthermore, P_0, P_1 can be simultaneously block upper-triangularized to the form

$$\begin{pmatrix} R_i & * \\ 0 & \frac{1}{2} \end{pmatrix},$$

where R_0, R_1 are 2×2 matrices. If $\delta < -\frac{3}{8}$ or $\delta > -\frac{1}{4}$ then R_0, R_1 can be simultaneously symmetrized. The 2×2 matrices Q_0, Q_1 can be simultaneously symmetrized for all δ, whence

$$\hat{\rho}(T_0|_W, T_1|_W) = \max\left\{\tfrac{1}{2}, |\tfrac{1}{2} + \delta|, |\tfrac{1}{2} + 2\delta|\right\}$$

and

$$\hat{\rho}(T_0|_V, T_1|_V) = \max\left\{\tfrac{1}{2}, |\tfrac{1}{2} + \delta|, |\tfrac{1}{2} + 2\delta|, |1 + 3\delta|\right\},$$

at least for $\delta \notin [-\frac{3}{8}, -\frac{1}{4}]$. In particular,

$$\begin{cases} \hat{\rho}(T_0|_W, T_1|_W) = \hat{\rho}(T_0|_V, T_1|_V) = \frac{1}{2}, & -\frac{1}{4} < \delta < -\frac{1}{6}, \\ \hat{\rho}(T_0|_W, T_1|_W) = \frac{1}{2} < \hat{\rho}(T_0|_V, T_1|_V) < 1, & -\frac{1}{6} < \delta < 0, \\ \hat{\rho}(T_0|_W, T_1|_W) < 1 < \hat{\rho}(T_0|_V, T_1|_V), & 0 < \delta < \frac{1}{4}. \end{cases}$$

The stretched dilation equation $(\frac{1}{2}, 0)$ is therefore a boundary point of C_V, yet belongs to C_W, and there exist nonstretched dilation equations in C_W that are not in C_V.

(c) Consider next the dilation equations lying on the line $(c_0, c_1) = (\frac{1}{2} + \delta, \delta)$. Again, $\delta = 0$ corresponds to the stretched dilation equation $(\frac{1}{2}, 0)$. Applying Proposition 4 we find that $W = V$ for all $\delta \neq 0$ (we can ignore those finitely many δ for which the eigenvalues of T_0 are not distinct). Therefore $\hat{\rho}(T_0|_W, T_1|_W) \geq \rho(T_0|_W) > 1$ for $\delta \notin [(-3 - \sqrt{57})/8, -\frac{5}{4}]$, i.e., there do not exist any continuous scaling functions corresponding to $\delta \notin [(-3 - \sqrt{57})/8, -\frac{5}{4}]$. Thus, although $\hat{\rho}(T_0|_W, T_1|_W) < 1$ at the point $(\frac{1}{2}, 0)$, an arbitrarily small change in the coefficients can result in a dilation equation with $\hat{\rho}(T_0|_W, T_1|_W) > 1$, i.e., $\hat{\rho}(T_0|_W, T_1|_W)$ is *not* a continuous function of the coefficients since the subspace W can change abruptly. Proposition 2 with W in place of V is therefore false.

(d) Finally, consider those dilation equations lying on the line $(c_0, c_1) = (\frac{3}{8} + \delta, \frac{1}{8} + \delta)$. The point $(\frac{3}{8}, \frac{1}{8})$, corresponding to $\delta = 0$, also lies on the line considered in part (a), and satisfies $\hat{\rho}(T_0|_W, T_1|_W) = \frac{1}{2}$, $\hat{\rho}(T_0|_V, T_1|_V) = \frac{5}{8}$, and its associated scaling vector is continuous with Hölder exponent 1. For $\delta \neq 0$, Proposition 4 can be used to show that $W = V$. Since $\hat{\rho}(T_0|_V, T_1|_V)$ is a continuous function of the coefficients (c_0, c_1), given ε small we can find a $\delta_0 > 0$ such that $|\hat{\rho}(T_0|_V, T_1|_V) - \frac{5}{8}| < \varepsilon$ when $0 < |\delta| < \delta_0$. Thus the scaling vectors for $0 < |\delta| < \delta_0$ are continuous, with maximum Hölder exponent $\alpha \leq -\log_2(\frac{5}{8} - \varepsilon) < 1$. However, the maximum Hölder exponent corresponding to $\delta = 0$ is exactly 1. As a function of the coefficients

(c_0, c_1), the maximum Hölder exponent is therefore *not* continuous. By comparison, $\hat{\rho}(T_0|_V, T_1|_V) < 1$ for $0 \le |\delta| < \delta_0$, so by Proposition 2 the associated scaling vectors deform continuously with respect to the sup-norm as δ varies. □

In the examples we have considered, the sets W and V are equal except under restricted conditions. We therefore conjecture that for each fixed N, the set of coefficients $\{c_0, \ldots, c_N\}$ satisfying (2) such that $W \neq V$ is a set of measure zero in the set of all coefficients satisfying (2).

We turn now to the proof of Theorem 3, for which we require a definition and lemma. In the statement of Proposition 4, the matrix T was assumed to have distinct eigenvalues, and hence T had a full set of linearly independent eigenvectors. Therefore, the definition of "component" used in hypothesis (b) of Proposition 4 was obvious. Now let A be an arbitrary $N \times N$ matrix with complex entries. If $\lambda \in \mathbf{C}$ is an eigenvalue of A, then $U_\lambda = \{u \in \mathbf{C}^N : (A - \lambda)^k u = 0 \text{ for some } k > 0\}$ is an A-invariant subspace of $\mathbf{C}^N$, for if $u \in U_\lambda$, then $Au = (A - \lambda)u + \lambda u \in U_\lambda$. By standard Jordan decomposition techniques we can write $\mathbf{C}^N = U_\lambda \oplus Z$, where Z is a unique A-invariant subspace of $\mathbf{C}^N$ [Her]. Any vector $u \in \mathbf{C}^N$ can therefore be uniquely written $u = u_\lambda + z$ where $u_\lambda \in U_\lambda$ and $z \in Z$. We say that u *has a component in* U_λ if $u_\lambda \neq 0$; note this implies $u \neq 0$.

One of the basic tools of our analysis is given in the next lemma.

LEMMA 3. *Let A be an $N \times N$ matrix and let λ be any eigenvalue of A. If $u \in \mathbf{C}^N$ has a component in U_λ, then there is a constant $C > 0$ such that $\|A^n u\| \ge C\,|\lambda|^n$ for all $n > 0$.*

Proof. As all norms on $\mathbf{C}^N$ are equivalent, it suffices to prove the result for the Euclidean space norm $\|u\| = (|u_1|^2 + \cdots |u_N|^2)^{1/2}$.

Let $\mathbf{C}^N = U_\lambda \oplus Z$ be the standard Jordan decomposition induced by A. The subspace U_λ can be written $U_\lambda = U_1 \oplus \cdots \oplus U_p$, where each U_i is a nontrivial, A-invariant subspace of U_λ that cannot be further decomposed into A-invariant subspaces. As u has a component in U_λ it must therefore have a component in some U_i, say U_1. Let $Z_1 = U_2 \oplus \cdots \oplus U_p \oplus Z$, so $\mathbf{C}^N = U_1 \oplus Z_1$ and U_1, Z_1 are A-invariant.

Since U_1 is finite-dimensional there must exist a smallest positive integer m such that $(A - \lambda)^m w = 0$ for all $w \in U_1$. Let $u_1 \in U_1$ be such that $(A - \lambda)^{m-1} u_1 \neq 0$ and set $u_k = (A - \lambda)^{k-1} u_1$ for $k = 2, \ldots, m$. Since U_1 is indecomposable, $\{u_1, \ldots, u_m\}$ comprises a basis for U_1. Moreover, the $\{u_k\}$ satisfy the relationships

$$\begin{aligned} Au_k &= u_{k+1} + \lambda u_k \quad \text{for } k = 1, \ldots, m-1, \\ Au_m &= \lambda u_m. \end{aligned} \tag{20}$$

Consider now the vector u; we have $u = v_1 + \cdots + v_m + z$ where each v_k is a scalar multiple of u_k and $z \in Z_1$. Since u has a component in U_1, at least one of the v_k is nonzero. Let k^* be that positive integer such that $v_{k^*} \neq 0$ while $v_k = 0$ for $k < k^*$. Then (20) implies that

$$A^n u = \lambda^n v_{k^*} + \sum_{k > k^*} P_{k,n}\, v_k + A^n z, \tag{21}$$

where each $P_{k,n}$ is a polynomial in λ of degree at most n. Now let Z_2 denote the proper subspace of $\mathbf{C}^N$ generated by Z_1 and $\{u_k : k \neq k^*\}$. Since $\dim(Z_2) = N - 1$, there exist unique vectors $z^* \in Z_2$ and $u^* \in Z_2^{\perp}$ such that $v_{k^*} = u^* + z^*$. Moreover, $u^* \neq 0$ since $v_{k^*} \notin Z_2$. It follows then from (21) that for each $n > 0$,

$$A^n u = \lambda^n u^* + \lambda^n z^* + \sum_{k > k^*} P_{k,n}\, v_k + A^n z = \lambda^n u^* + z_n,$$

where $z_n \in Z_2$. Since u^* is orthogonal to z_n, we therefore have

$$\|A^n u\|^2 = \|\lambda^n u^*\|^2 + \|z_n\|^2 \geq |\lambda|^{2n} \|u^*\|^2. \qquad \square$$

The proof of Lemma 3 can be simplified if A is diagonalizable.

Proof of Theorem 3. (a) The fact that the hypothesis $\hat{\rho}(T_0|_V, T_1|_V) < 1$ in Theorem 2 can be replaced by $\hat{\rho}(T_0|_W, T_1|_W) < 1$ and that the assumption of (2) is then no longer necessary is a simple observation following from the proof of Theorem 2 given in [DL2]. Moreover, the same proof implies that v is Hölder continuous for each exponent $0 \leq \alpha < -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$, or $0 \leq \alpha \leq -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$ if there exists a constant $C > 0$ such that $\hat{\rho}_m \leq C^{1/m} \hat{\rho}(T_0|_W, T_1|_W)$ for every m. In particular, $\alpha_{\max} \geq -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$.

(b) Assume that v is a continuous scaling vector; then v is Hölder continuous for some exponent α in the range $0 \leq \alpha \leq 1$. Let K be any corresponding Hölder constant.

Choose any $m > 0$, and let $T = T_{d_1} \cdots T_{d_m}$ be any fixed product of the matrices T_0, T_1. Let λ be any eigenvalue of $T|_W$. By definition of W, there must be dyadic points $x, y \in [0,1]$ such that $v(x) - v(y)$ has a component in $U_\lambda = \{u \in W : (T-\lambda)^k u = 0 \text{ for some } k > 0\}$. Let $x = .x_1 x_2 \ldots$ and $y = .y_1 y_2 \ldots$ be binary expansions of x and y, respectively, and define the dyadic points X_k, Y_k for $k > 0$ by

$$\begin{aligned} X_1 &= .d_1 \ldots d_m x_1 x_2 \ldots, & Y_1 &= .d_1 \ldots d_m y_1 y_2 \ldots, \\ X_2 &= .d_1 \ldots d_m d_1 \ldots d_m x_1 x_2 \ldots, & Y_2 &= .d_1 \ldots d_m d_1 \ldots d_m y_1 y_2 \ldots, \end{aligned}$$

etc., so $v(X_k) = T^k v(x)$ and $v(Y_k) = T^k v(y)$ for all $k > 0$. We then have by Lemma 3 that

$$\|v(X_k) - v(Y_k)\| = \|T^k\big(v(x) - v(y)\big)\| \geq C\,|\lambda|^k$$

for some constant $C > 0$ independent of k. Since $|X_k - Y_k| \to 0$ as $k \to \infty$ and v is continuous, we must therefore have $|\lambda| < 1$. In fact, $|X_k - Y_k| = 2^{-mk}\,|x - y|$, so

$$\|v(X_k) - v(Y_k)\| \geq K_\lambda\,|X_k - Y_k|^{\alpha_\lambda}$$

for all k, where $\alpha_\lambda = -\log_2 |\lambda|^{1/m}$ and $K_\lambda = C\,|x-y|^{-\alpha_\lambda}$. However, by the Hölder continuity of v,

$$\|v(X_k) - v(Y_k)\| \leq K\,|X_k - Y_k|^{\alpha}$$

for all k, so $|X_k - Y_k|^{\alpha - \alpha_\lambda} \geq K_\lambda / K$ for all k. Since $|X_k - Y_k| \to 0$ as $k \to \infty$, this implies $\alpha \leq \alpha_\lambda$.

Taking the supremum over all eigenvalues of $T = T_{d_1} \cdots T_{d_m}$ for all choices of $d_j = 0, 1$, we obtain $\hat{\sigma}_m < 1$ and $\alpha \leq -\log_2 \hat{\sigma}_m$. Thus $\hat{\rho}(T_0|_W, T_1|_W) = \sup \hat{\sigma}_m \leq 1$ and $\alpha \leq -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$, so $\alpha_{\max} \leq -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$. Combining this with the proof of part (a), we obtain $\alpha_{\max} = -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$.

Suppose that $\alpha = \alpha_{\max}$. Then given any $x = .x_1 x_2 \ldots$ and $y = .y_1 y_2 \ldots$, we have

$$\|T_{d_1} \cdots T_{d_m}(v(x) - v(y))\| = \|v(X) - v(Y)\| \leq K\,|X - Y|^{\alpha} = K\,(2^{-m}\,|x-y|)^{\alpha},$$

where $X = .d_1 \ldots d_m x_1 x_2 \ldots$ and $Y = .d_1 \ldots d_m y_1 y_2 \ldots$. By considering a basis $\{v(x_j) - v(y_j)\}_{j=1}^J$ for W, we obtain that there is a constant $C > 0$ such that $\|T_{d_1} \cdots T_{d_m} w\| \leq C\,2^{-m\alpha}\,\|w\|$ for every $w \in W$, and therefore $\hat{\rho}_m \leq C^{1/m}\,2^{-\alpha} = C^{1/m}\,\hat{\rho}(T_0|_W, T_1|_W)$.

The proof is now complete except for the case of a dilation equation satisfying

$$\sup_m \hat{\sigma}_m = \hat{\rho}(T_0|_W, T_1|_W) = 1 \quad \text{and} \quad \hat{\sigma}_m < 1 \quad \text{for every } m. \tag{22}$$

Assume therefore that a continuous scaling vector exists for a dilation equation satisfying (22). Then each $\hat{\rho}_m \geq 1$, and so for every $m > 0$ there is some product of $T_0|_W$, $T_1|_W$ of length m having norm greater than or equal to 1. Using a diagonalization process, we can therefore construct a sequence $d_1, d_2, \ldots$ so that $S_m = T_{d_1} \cdots T_{d_m}$ satisfies $\|S_m|_W\| \geq 1$ for all m. Let $\{v(x_j) - v(y_j)\}_{j=1}^J$ be a basis for W. Then there exist vectors $u_m = \sum_{j=1}^J a_{m,j}\,(v(x_j) - v(y_j)) \in W$ such that $\|u_m\| = 1$ and $\|S_m u_m\| \geq 1$ for every m. Since W is finite dimensional, we can (by taking subsequences if necessary) suppose that for each j the limit $a_j = \lim_{m\to\infty} a_{m,j}$ exists. Then $S_m u_m \to u$ with $\|u\| = \lim_{m\to\infty} \|S_m u_m\| \geq 1$. Now, by the continuity of v, we have that $S_m v(x_j)$, $S_m v(y_j) \to v(x)$ as $m \to \infty$, where $x = .d_1 d_2 \ldots$, so

$$u = \lim_{m\to\infty} S_m u_m = \lim_{m\to\infty} \sum_{j=1}^J a_{m,j}\, S_m(v(x_j) - v(y_j)) = \sum_{j=1}^J a_j\,(v(x) - v(x)) = 0,$$

a contradiction. □

The proof of Theorem 3(b) raises an interesting question: Do there exist dilation equations such that (22) holds? A scaling function for such a dilation equation must be discontinuous by Theorem 3(b). More generally, the same existence question can be asked with the value 1 in (22) replaced by any positive number; we do not know of any specific examples of such dilation equations, i.e., dilation equations for which the supremum $\sup \hat{\sigma}_m$ is not achieved for any m. Note that $\sup \hat{\sigma}_m = \hat{\sigma}_1$ when simultaneous symmetrization occurs; however, the supremum may be achieved even though simultaneous symmetrization does not occur. For example, consider the dilation equation defined by $N = 3$ and $(c_0, c_3) = (\frac{1}{3}, \frac{1}{3})$. In this case $V = W$ and $\sup \hat{\sigma}_m = \hat{\sigma}_2 > \hat{\sigma}_1$. Lagarias and Wang have conjectured that the supremum must always be achieved; in [LW] they prove several results related to this conjecture.

Theorem 3 discusses only global Hölder exponents. It is possible for a local Hölder exponent at a particular point to be strictly greater than the maximum global Hölder exponent. For example, assume $\hat{\rho}(T_0|_W, T_1|_W) < 1$; then v is continuous by Theorem 3, and its maximum global Hölder exponent of continuity is $\alpha_{\max} = -\log_2 \hat{\rho}(T_0|_W, T_1|_W)$. Fix any product $T = T_{d_1} \cdots T_{d_m}$, and let z be any rational point whose binary expansion is of the form $z = .z_1 \ldots z_n d_1 \ldots d_m d_1 \ldots d_m \ldots$. If $y > z$ with $2^{-n-(k+1)m} \leq y - z < 2^{-n-km}$, then the first $n + km$ digits of y are the same as those of z (using the upper binary expansion of z if z is dyadic). Therefore

$$\begin{aligned}
\|v(y) - v(z)\| &= \|T_{z_1} \cdots T_{z_n} T^k \big(v(\tau^{n+km} y) - v(\tau^{n+km} z)\big)\| \\
&\leq C\, \|(T_{z_1} \cdots T_{z_n})|_W\|\, \|T^k|_W\| \\
&\leq C'\, \theta^k \\
&= C'\, \theta^{-(n+m)/m}\, (2^{-n-(k+1)m})^{-\log_2 \theta^{1/m}} \\
&\leq C'\, \theta^{-(n+m)/m}\, |y - z|^{-\log_2 \theta^{1/m}},
\end{aligned}$$

where $\rho(T|_W) < \theta < 1$ and C and C' are constants independent of k. Using a similar argument from the left (with the lower binary expansion for z if z is dyadic), we conclude that v is locally Hölder continuous at z at least for each exponent α_z in the range

$$0 \leq \alpha_z < \begin{cases} -\log_2 \rho(T|_W)^{1/m}, & \text{if } m > 1, \\ -\log_2 \max\{\rho(T_0|_W),\, \rho(T_1|_W)\}, & \text{if } m = 1. \end{cases} \tag{23}$$

Thus the maximum local Hölder exponent at z is bounded below by the right-hand side of (23). This quantity depends only on the single product T (or on T_0 and T_1 if z is dyadic), and therefore can be strictly greater than the maximum global exponent, which depends on all possible products of T_0 and T_1. The proof of Theorem 3(b) demonstrates that the right-hand side of (23) is the maximum local Hölder exponent at the point $x = .d_1 \dots d_m d_1 \dots d_m \dots$. If $T_0|_W$ and $T_1|_W$ are invertible, then

$$\|v(\tau^m y) - v(x)\| \leq \|(T_{z_1} \cdots T_{z_n}|_W)^{-1}\| \, \|v(y) - v(z)\|,$$

and therefore in this case the right-hand side of (23) is also the maximum local Hölder exponent at any z of the form $z = .z_1 \dots z_n d_1 \dots d_m d_1 \dots d_m \dots$.

Example 8. Consider $N = 3$ as in Example 2. Given coefficients (c_0, c_3), the maximum local Hölder exponent of continuity at any dyadic point is $\alpha_{\text{dyadic}} = -\log_2 \max\{|c_0|, |c_3|, |1 - c_0 - c_3|\}$. Consider the following specific examples.

(a) In [CH1] we determined that the scaling function determined by $(c_0, c_3) = (0.6, -0.2)$ is Hölder continuous with maximum global Hölder exponent somewhere in the range $0.598 \leq \alpha_{\max} \leq 0.600$. However, $\alpha_{\text{dyadic}} = -\log_2 0.6 \approx 0.737$. This is the largest maximum local Hölder exponent at dyadic points for (c_0, c_3) satisfying both (2) and (3).

(b) If $\max\{|c_0|, |c_3|, |1-c_0-c_3|\} < \frac{1}{2}$ then α_{dyadic} strictly exceeds 1. It follows that v is differentiable at all dyadic z with $v'(z) = 0$. If v was differentiable everywhere, then this would imply that v was identically constant, which is not the case. For example, for $(c_0, c_3) = (\frac{1}{3}, \frac{1}{3})$, the maximum global Hölder exponent is only $\alpha_{\max} \approx 0.891$. □

Finally, we use the techniques applied in the proof of Theorem 3 to make some observations about discontinuous scaling functions. In particular, we obtain the following theorem.

THEOREM 4. *Assume v is a scaling vector.*

(a) *If $\hat{\rho}(T_0|_W, T_1|_W) \geq 1$ with $\hat{\sigma}_m \geq 1$ for some m, then v is discontinuous. If $T_0|_W$, $T_1|_W$ are invertible, then discontinuities occur on a dense set of rational points in $[0, 1]$.*

(b) *If $\hat{\rho}(T_0|_W, T_1|_W) > 1$, then v is unbounded. If $T_0|_W$, $T_1|_W$ are invertible, then singularities occur on a dense set of rational points in $[0, 1]$.*

Proof. (a) If $\hat{\sigma}_m \geq 1$ for some m then there is some product $T = T_{d_1} \cdots T_{d_m}$ and some eigenvalue λ of $T|_W$ such that $|\lambda| \geq 1$. Let X_k, Y_k be as in the proof of Theorem 3(b); it follows then that $\|v(X_k) - v(Y_k)\| \geq C\,|\lambda|^k$ for some constant $C > 0$ independent of k. As X_k, $Y_k \to X = .d_1 \dots d_m d_1 \dots d_m \dots$, it follows that v cannot be continuous at X. An argument similar to the one following (23) shows that if $T_0|_W$, $T_1|_W$ are invertible, then discontinuities also occur at every point z of the form $z = .z_1 \dots z_n d_1 \dots d_m d_1 \dots d_m \dots$.

(b) If $\hat{\rho}(T_0|_W, T_1|_W) > 1$ then $\hat{\sigma}_m > 1$ for some m, and therefore there is some product $T = T_{d_1} \cdots T_{d_m}$ and some eigenvalue λ of $T|_W$ such that $|\lambda| > 1$. The remainder of the proof follows just as in part (a). □

As a final comment, suppose we are given a scaling vector v with $\hat{\rho}(T_0|_W, T_1|_W) > 1$. We have then by Theorem 4(b) that v is unbounded. The singularities constructed in the proof of this result occur only at rational points. Consider then the function $\tilde{v}$ obtained by setting $\tilde{v}(x) = v(x)$ for nonrational x and $\tilde{v}(x) = 0$ for rational x. This is also a scaling vector, and $\hat{\rho}(\tilde{T}_0|_{\tilde{W}}, \tilde{T}_1|_{\tilde{W}}) = 0$. However, $\tilde{v}$ is clearly discontinuous; note that Theorem 3(a) does not imply continuity since $\tilde{v}$ is not given at dyadic points by the method of Construction 1. Since $\hat{\rho}(\tilde{T}_0|_{\tilde{W}}, \tilde{T}_1|_{\tilde{W}}) < 1$, Theorem 4(b) does not

imply that $\tilde{v}$ is unbounded. We therefore ask whether it is possible that $\tilde{v}$ be bounded. As pointed out by the referee, the subspaces W or $\tilde{W}$ are probably not the best for dealing with this question. A more relevant subspace might be

$$\bigcap_S \operatorname{span}\{v(x) - v(y) : x, y \in S\},$$

where the intersection is taken over all subsets $S \subset [0,1]$ with measure one that are invariant under τ.

Acknowledgments. We thank George Benke of The MITRE Corporation and Gil Strang of MIT for many stimulating discussions and valuable insights on this subject.

REFERENCES

[BW] M. A. Berger and Y. Wang, *Bounded semi-groups of matrices*, Linear Algebra Appl., 166 (1992), pp. 21–27.

[CDM] A. Cavaretta, W. Dahmen, and C. A. Micchelli, *Stationary Subdivision*, Mem. Amer. Math. Soc., 93 (1991), pp. 1–186.

[CH1] D. Colella and C. Heil, *The characterization of continuous, four-coefficient scaling functions and wavelets*, IEEE Trans. Inf. Th., Special Issue on Wavelet Transforms and Multiresolution Signal Analysis, 38 (1992), pp. 876–881.

[CH2] ———, *Characterizations of scaling functions, II. Distributional and functional solutions*, manuscript.

[D] I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Comm. Pure Appl. Math., 41 (1988), pp. 909–996.

[DL1] I. Daubechies and J. Lagarias, *Two-scale difference equations: I. Existence and global regularity of solutions*, SIAM J. Math. Anal., 22 (1991), pp. 1388–1410.

[DL2] ———, *Two-scale difference equations: II. Local regularity, infinite products of matrices and fractals*, SIAM J. Math. Anal., 23 (1992), pp. 1031–1079.

[DL3] ———, *Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 161 (1992), pp. 227–263.

[DD] G. Deslauriers and S. Dubuc, *Symmetric iterative interpolation processes*, Constr. Approx., 5 (1989), pp. 49–68.

[Du] S. Dubuc, *Interpolation through an iterative scheme*, J. Math. Anal. Appl., 114 (1986), pp. 185–205.

[DGL] N. Dyn, J. A. Gregory, and D. Levin, *Analysis of uniform binary subdivision schemes for curve design*, Constr. Approx., 7 (1991), pp. 127–147.

[E] T. Eirola, *Sobolev characterization of solutions of dilation equations*, SIAM J. Math. Anal., 23 (1992), pp. 1015–1030.

[H] C. Heil, *Methods of solving dilation equations*, in Prob. and Stoch. Methods in Anal. with Appl., J. S. Byrnes, K. A. Hargreaves, and K. Berry, eds., NATO Adv. Sci. Inst. Ser. C, Math. Phys. Sci. 372, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1992, pp. 15–45.

[HC] C. Heil and D. Colella, *Dilation equations and the smoothness of compactly supported wavelets*, in Wavelets: Mathematics and Applications, J. Benedetto and M. Frazier, eds., CRC Press, Boca Raton, FL, 1993, pp. 161–200.

[HS] C. Heil and G. Strang, *Continuity of the joint spectral radius: Application to wavelets*, in Linear Algebra for Signal Processing, A. Bojanczyk and G. Cybenko, eds., IMA Vol. Math. Appl., Springer–Verlag, New York, to appear.

[Her] I. N. Herstein, *Topics in Algebra*, 2nd ed., John Wiley and Sons, New York, 1975.

[LW] J. C. Lagarias and Y. Wang, *The finiteness conjecture for the joint spectral radius*, Linear Algebra Appl., to appear.

[L] W. Lawton, *Tight frames of compactly supported affine wavelets*, J. Math. Phys., 31 (1990), pp. 1898–1901.

[M] S. G. MALLAT, *Multiresolution approximations and wavelet orthonormal bases for* $L^2(\mathbf{R})$, Trans. Amer. Math. Soc., 315 (1989), pp. 69–87.

[MP1] C. A. MICCHELLI AND H. PRAUTZSCH, *Refinement and subdivision for spaces of integer translates of compactly supported functions*, in Numerical Analysis, D. F. Griffith and G. A. Watson, eds., Academic Press, New York, 1987, pp. 192–222.

[MP2] ———, *Uniform refinement of curves*, Linear Algebra Appl., 114/115 (1989), pp. 841–870.

[RS] G. C. ROTA AND G. STRANG, *A note on the joint spectral radius*, Kon. Nederl. Akad. Wet. Proc. A, 63 (1960), pp. 379–381.

[S] G. STRANG, *Wavelets and dilation equations: a brief introduction*, SIAM Rev., 31 (1989), pp. 614–627.

[V] L. VILLEMOES, *Energy moments in time and frequency for two-scale difference equations*, SIAM J. Math. Anal., 23 (1992), pp. 1519–1543.

[W] Y. WANG, *Two-scale dilation equations and the cascade algorithm*, Random and Computational Dynamics, to appear.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 519–529, April 1994

011

LINEAR OPERATORS PRESERVING COMPLEX ORTHOGONAL EQUIVALENCE ON MATRICES*

ROGER A. HORN†, CHI-KWONG LI‡, AND DENNIS I. MERINO§

Abstract. Two complex matrices A and B are said to be (complex) orthogonally equivalent if there exist (complex) orthogonal matrices Q_1 and Q_2 such that $A = Q_1BQ_2$. In this note the authors obtain characterization of linear operators that preserve complex orthogonal equivalence.

Key words. linear preservers, complex orthogonal matrix

AMS subject classifications. 15A04, 15A57, 15A24

1. Introduction and statement of result.

Let $M_{m,n}$ be the set of all m-by-n matrices with complex entries; and we write $M_n \equiv M_{n,n}$. A matrix $Q \in M_n$ is said to be (complex) *orthogonal* if $Q^TQ = I$. Two matrices $A, B \in M_{m,n}$ are said to be (complex) *orthogonally equivalent*, denoted $A \sim B$, if $A = Q_1BQ_2$ for some orthogonal matrices $Q_1 \in M_m$ and $Q_2 \in M_n$. One checks that $\sim$ is an equivalence relation on $M_{m,n}$. We are interested in studying *linear orthogonal equivalence preservers* on $M_{m,n}$; that is, $T : M_{m,n} \to M_{m,n}$ and $T(A) \sim T(B)$ whenever $A \sim B$. We prove the following, which is our main result.

THEOREM 1.1. *Let T be a given linear operator on $M_{m,n}$. Then T preserves orthogonal equivalence if and only if there exist complex orthogonal $Q_1 \in M_m$ and $Q_2 \in M_n$ and a scalar $\alpha \in \mathbb{C}$ such that either:*

(1) *$T(A) = \alpha Q_1AQ_2$ for all $A \in M_{m,n}$; or*

(2) *$m = n$ and $T(A) = \alpha Q_1A^TQ_2$ for all $A \in M_n$.*

We have divided the proof of the theorem into three parts. In § 2, we establish that either $T = 0$ or T is nonsingular. In § 3, we show that T has the form asserted in the theorem, except that Q_1 and Q_2 are nonsingular, but are not necessarily orthogonal. Finally, in § 4, the theorem is proved.

Similar problems have been studied in [5] and [7], and we use their general approach. We analyse the *orbits*

$$\mathcal{O}(A) = \{X \in M_{m,n} : X \sim A\}$$

and their corresponding tangent spaces $\mathcal{T}_A$. It is known that these orbits are homogeneous differentiable manifolds [1]. As in [5] and [7], it is necessary to develop some special techniques to supplement the general approach in solving the problem. Unlike the relations considered in [7], little is known about complex orthogonal equivalence and a simple canonical form is not available. This makes the problem more difficult and more interesting. In fact, the results obtained in this paper may give more insight into, and better understanding of, the orthogonal equivalence relation.

* Received by the editors March 17, 1992; accepted for publication (in revised form) July 28, 1992.

† Department of Mathematics, The University of Utah, Salt Lake City, Utah 84112 (rhorn@math.utah.edu).

‡ Department of Mathematics, The College of William and Mary, Williamsburg, Virginia 23185 (ckli@cs.wm.edu). Research supported by National Science Foundation grants DMS 89-00922 and DMS 91-00344. This research was done when the author was visiting the Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland, during the Spring semester, 1991.

§ Department of Mathematics, Southeastern Louisiana University, Hammond, Louisiana 70402 (fmat1649@vega.selu.edu).

We denote the standard basis of $M_{m,n}$ by $\{E_{11}, E_{12}, \ldots, E_{m,n}\}$. When $E_{ij} \in M_n$, we set $F_{ij} \equiv E_{ij} - E_{ji}$ and, to avoid confusion when $E_{ij} \in M_m$, we set $G_{ij} \equiv E_{ij} - E_{ji}$. Note that all F_{ij} and G_{ij} are skew-symmetric matrices. We denote the standard basis of $\mathbb{C}^n$ by $\{e_1, \ldots, e_n\}$. The n-by-n identity matrix is denoted by I, or when necessary for clarity, by I_n. A vector $x \in \mathbb{C}^n$ is said to be *isotropic* if $x^T x = 0$.

2. Preliminary results. The following observation is used repeatedly in our arguments.

LEMMA 2.1. *Let* $\mathbf{V}$ *be a subspace of* $M_{m,n}$ *that is invariant under orthogonal equivalence, and let* $T : \mathbf{V} \to M_{m,n}$ *be a linear transformation that preserves orthogonal equivalence. Then*

$$T(\text{span } \mathcal{O}(A)) \subset \text{span } \mathcal{O}(T(A)) \text{ and } T(\mathcal{T}_A) \subset \mathcal{T}_{T(A)}$$

for every $A \in \mathbf{V}$. *Consequently, if* T *is nonsingular, then*

$$\dim \text{span } \mathcal{O}(A) \leq \dim \text{span } \mathcal{O}(T(A)) \text{ and } \dim \mathcal{T}_A \leq \dim \mathcal{T}_{T(A)}$$

for every $A \in \mathbf{V}$.

LEMMA 2.2. *Let* X, $Y \in M_{m,n}$ *be given, let* T *be a linear operator on* $M_{m,n}$ *that preserves orthogonal equivalence, and suppose* $X \in \mathcal{T}_X$. *If* $Y \notin \mathcal{T}_Y$, *then* $T(X) \notin \mathcal{O}(Y)$.

Proof. Suppose $T(X) = Q_1 Y Q_2$ for some orthogonal $Q_1 \in M_m$ and $Q_2 \in M_n$. Let $T_1 \equiv Q_1^T T Q_2^T$. Then $Y = Q_1^T T(X) Q_2^T = T_1(X) \in T_1(\mathcal{T}_X) \subset \mathcal{T}_{T_1(X)} = \mathcal{T}_Y$. □

Remark. The argument used to prove the preceding lemma can be used to obtain its conclusion under more general hypotheses: Let $\mathcal{G}$ be a given group of nonsingular linear operators on $M_{m,n}$, and say $A \sim B$ if $A = L(B)$ for some $L \in \mathcal{G}$. Let T be a given linear operator on $M_{m,n}$ such that $T(A) \sim T(B)$ whenever $A \sim B$. If $X \in \mathcal{T}_X$ and $Y \notin \mathcal{T}_Y$, then $T(X) \neq L(Y)$ for all $L \in \mathcal{G}$.

LEMMA 2.3. Span $\mathcal{O}(A) = M_{m,n}$ *for every nonzero* $A \in M_{m,n}$.

Proof. Let $A \in M_{m,n}$ be a given nonzero matrix. There exists orthogonal (in fact, permutation matrices) $P \in M_m$ and $Q \in M_n$ such that the (1,1)-entry of $B = PAQ$, say, b_{11}, is nonzero. For a given positive integer k, define the diagonal orthogonal matrix

$$D_k \equiv \text{diag}(1, -1, -1, \ldots, -1) \in M_k.$$

Then $4b_{11}E_{11} = (I_m + D_m)B(I_n + D_n) = B + D_m B + B D_n + D_m B D_n \in \text{span } \mathcal{O}(A)$, so $E_{11} \in \text{span } \mathcal{O}(A)$. Since there exist permutation (and hence, orthogonal) matrices $P_i \in M_m$ and $P_j \in M_n$ such that $E_{ij} = P_i E_{11} P_j$, every $E_{ij} \in \text{span } \mathcal{O}(A)$ and hence span $\mathcal{O}(A) = M_{m,n}$. □

LEMMA 2.4. *Let* T *be a given linear operator on* $M_{m,n}$. *If* T *preserves orthogonal equivalence, then either* $T = 0$ *or* T *is nonsingular.*

Proof. Suppose that ker T contains a nonzero matrix A. By Lemma 2.1,

$$T(\text{span } \mathcal{O}(A)) \subset \text{span } \mathcal{O}(T(A)) = \{0\}.$$

Now, Lemma 2.3 guarantees that span $\mathcal{O}(A) = M_{m,n}$. Hence, $T = 0$. □

We use the following result, which is Lemma (1) in [3].

LEMMA 2.5. *Let* $X_1, X_2 \in M_{n,k}$ *with* $1 \leq k \leq n$. *There exists a complex orthogonal* $Q \in M_n$ *such that* $X_1 = QX_2$ *if and only if the following two conditions are satisfied*:

(a) $X_1^T X_1 = X_2^T X_2$, *and*

(b) *there exists a nonsingular* $B \in M_n$ *such that* $X_1 = BX_2$.

Note that if $X_1, X_2 \in M_{n,k}$ have full rank, Lemma 2.5 ensures that there exists an orthogonal $Q \in M_n$ such that $X_1 = QX_2$ if and only if $X_1^T X_1 = X_2^T X_2$. In particular, for $n \geq 2$ and any given nonzero $z \in \mathbb{C}^n$, there are two possibilities:

(a) If $z^T z \neq 0$, then $z = \alpha Q e_1$ for some orthogonal $Q \in M_n$ and some nonzero $\alpha \in \mathbb{C}$; and

(b) if $z \neq 0$ and $z^T z = 0$, then $z = Q(e_1 + ie_2)$ for some orthogonal $Q \in M_n$.

Let $A \in M_{m,n}$ be given. Then $\mathcal{O}(A) = \{Q_1 A Q_2 : Q_1 \in M_m, Q_2 \in M_n, Q_1^T Q_1 = I_m$, and $Q_2^T Q_2 = I_n\}$. If $B \sim A$, then $A^T A$ is similar to $B^T B$ and AA^T is similar to BB^T. Suppose A has rank 1, so $A = xy^T$ for some $x \in \mathbb{C}^m$ and $y \in \mathbb{C}^n$. Then rank $A^T A = 0$ or 1 according to whether $x^T x$ is zero or nonzero, and rank $A^T A = 0$ or 1 according to whether $y^T y$ is zero or nonzero. Depending on the four possibilities for the pair (rank $A^T A$, rank AA^T), it follows that for some nonzero scalar $\alpha \in \mathbb{C}$, $\mathcal{O}(A)$ contains exactly one of the following: (a) αE_{11}; (b) $\alpha(E_{11} + iE_{12})$; (c) $\alpha(E_{11} + iE_{21})$; or (d) $\alpha(E_{11} - E_{22} + iE_{12} + iE_{21})$.

The same reasoning leads immediately to the following lemma.

LEMMA 2.6. *Let* m *and* n *be given integers with* $m, n \geq 2$, *let* $E_{11}, E_{12}, E_{21}, E_{22} \in M_{m,n}$ *be standard basis matrices, and let* $E \equiv E_{11} - E_{22} + iE_{12} + iE_{21}$. *Then*

(a) $\mathcal{O}(E_{11}) = \{q_1 q_2^T : q_1 \in \mathbb{C}^m$, $q_2 \in \mathbb{C}^n$, *and* $q_1^T q_1 = q_2^T q_2 = 1\}$.

(b) $\mathcal{O}(E_{11} + iE_{12}) = \{qy^T : q \in \mathbb{C}^m$, $y \in \mathbb{C}^n, q^T q = 1$, *and* $y^T y = 0\}$.

(c) $\mathcal{O}(E_{11} + iE_{21}) = \{xq^T : x \in \mathbb{C}^m$, $q \in \mathbb{C}^n, x^T x = 0$, *and* $q^T q = 1\}$.

(d) $\mathcal{O}(E) = \{xy^T : x \in \mathbb{C}^m$, $y \in \mathbb{C}^n$, *and* $x^T x = y^T y = 0\}$.

If $Q(t) \in M_n$ is a differentiable family of orthogonal matrices with $Q(0) = I$, then differentiation of the identity $Q(t)Q(t)^T = I$ at $t = 0$ shows that $Q'(0) + Q'(0)^T = 0$, that is, $Q'(0)$ is skew-symmetric. Conversely, if $B \in M_n$ is a given skew-symmetric matrix, then $Q(t) \equiv e^{tB}$ is a differentiable family of orthogonal matrices such that $Q(0) = I$ and $Q'(0) = B$ [6]. If $A \in M_{m,n}$ is given and $Q_1(t) \in M_m$ and $Q_2(t) \in M_n$ are given differentiable families of orthogonal matrices with $Q_1(0) = I_m$ and $Q_2(0) = I_n$, one computes that

$$\frac{d}{dt}\{Q_1(t) A Q_2(t)\}|_{t=0} = Q_1'(0)A + AQ_2'(0).$$

Thus, the tangent space to $\mathcal{O}(A)$ at A is given explicitly by

$$\mathcal{T}_A = \{XA + AY : X \in M_m, Y \in M_n, X + X^T = 0, \text{ and } Y + Y^T = 0\}. \tag{2.1}$$

DEFINITION 2.7. *Let* $A \in M_{m,n}$ *with* $n \geq 2$. *Then*

$$\mathcal{S}_A \equiv \{AF_{ij} : i = 1, 2, \text{ and } i < j \leq n\}.$$

Note that $\mathcal{S}_A \subset \mathcal{T}_A$ for every $A \in M_{m,n}$.

Suppose $A \in M_{m,n}$ and rank $A \geq 2$. Then there exists a permutation matrix $Q \in M_n$ such that the first two columns of AQ are linearly independent. Since $F_{ij} = E_{ij} - E_{ji} \in M_n$ is skew-symmetric, $\mathcal{S}_{AQ} \subset \mathcal{T}_{AQ}$ is a linearly independent subset with $2n - 3$ elements. Thus, dim $\mathcal{T}_A =$ dim $\mathcal{T}_{AQ} \geq 2n - 3$ whenever rank $A \geq 2$. A similar argument shows that dim $\mathcal{T}_A \geq 3n - 6$ whenever rank $A \geq 3$.

Now suppose $A \in M_{m,n}$, rank $A \geq 2$, and dim $\mathcal{T}_A = 2n - 3$. Let $Q \in M_n$ be a permutation matrix such that the first two columns of

$$B \equiv AQ = [\, b_1 \; b_2 \; b_3 \; \ldots \; b_n]$$

are independent. Then $\mathcal{S}_B \subset \mathcal{T}_B$ and dim span $\mathcal{S}_B = 2n-3 = \dim \mathcal{T}_A = \dim \mathcal{T}_B$, so $\mathcal{T}_B = \text{span } \mathcal{S}_B = \{BY : Y \in M_n \text{ and } Y + Y^T = 0\}$. If $n > 3$ and $j > 3$, note that

$$BF_{3j} = [\,0\ 0\ -b_j\ 0\ \dots\ 0\ b_3\ 0\ \dots\ 0]$$

and the only matrices in $\mathcal{S}_B$ with nonzero entries in the third column are $BF_{13} = [\,-b_3\ 0\ b_1\ 0\ \dots\ 0\,]$ and $BF_{23} = [\,0\ -b_3\ b_2\ 0\ \dots\ 0\,]$; thus, $b_3 = 0$ and b_j is a linear combination of b_1 and b_2. Hence, rank A = rank $B = 2$ if $n > 3$. Note that if $n = 2$ or $m = 2$ and if $A \in M_{m,n}$ with rank $A \geq 2$, then rank $A = 2$.

LEMMA 2.8. *Let $A \in M_{m,n}$ be nonzero. Then*

$$\mathcal{T}_A = \{XA + AY : X \in M_m, Y \in M_n \text{ and } X + X^T = 0, Y + Y^T = 0\}$$

and

(a) dim $\mathcal{T}_A = m + n - 2$ *if* $A \in \{E_{11}, E_{11} + iE_{12}, E_{11} + iE_{21}\}$;
(b) dim $\mathcal{T}_A = m + n - 3$ *if* $A = E_{11} - E_{22} + iE_{12} + iE_{21}$;
(c) dim $\mathcal{T}_A \geq 2n - 3$ *if* rank $A \geq 2$. *Moreover, if* $n > 3$, rank $A \geq 2$, *and* dim $\mathcal{T}_A = 2n - 3$, *then* rank $A = 2$ *and there exists a permutation matrix* $Q \in M_n$ *such that* $\mathcal{T}_{AQ} = \text{span } \mathcal{S}_{AQ}$;
(d) dim $\mathcal{T}_A \geq 3n - 6$ *if* rank $A \geq 3$.
(e) *If* rank $A = 2$ *and* dim $\mathcal{T}_A = 2n - 3$, *then there exists a permutation matrix* $Q \in M_n$ *such that* dim span $\mathcal{S}_{AQ} = 2n - 3$.

Proof. The asserted form of $\mathcal{T}_A$, as well as assertions (c) and (d), have been verified. Assertions (a) and (b) follow from direct computations. We consider in detail only the case in which $A = E_{11} + iE_{12}$; the other cases can be dealt with similarly. Let $X = [x_{ij}] \in M_m$ and $Y = [y_{ij}] \in M_n$ be skew-symmetric. Then

$$XA = \begin{bmatrix} 0 & 0 & 0 \cdots 0 \\ x_{21} & ix_{21} & 0 \cdots 0 \\ \vdots & \vdots & \vdots \ddots \vdots \\ x_{m1} & ix_{m1} & 0 \cdots 0 \end{bmatrix}$$

and

$$AY = \begin{bmatrix} iy_{21} & y_{12} & y_{13} + iy_{23} & \cdots & y_{1n} + iy_{2n} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Since $y_{21} = -y_{12}$, dim $\mathcal{T}_A = m + n - 2$. □

3. A rank-preserving property of T^{-1}.

PROPOSITION 3.1. *Let T be a nonsingular linear operator on $M_{m,n}$ that preserves orthogonal equivalence. Then $T^{-1}(E)$ has rank 1 whenever E has rank 1.*

We have organized the proof of Proposition 3.1 into a sequence of ten lemmata.

LEMMA 3.2. *Let T be a nonsingular linear operator on $M_{m,n}$ that preserves orthogonal equivalence. If $E \in M_{m,n}$ and* rank $E = 1$, *then* dim $\mathcal{T}_{T^{-1}(E)} \leq m + n - 2$.

Proof. Lemma 2.1 shows that dim $\mathcal{T}_{T^{-1}(E)} \leq$ dim $\mathcal{T}_E$, while Lemma 2.6 and Lemma 2.8 (a), (b) give dim $\mathcal{T}_E \leq m + n - 2$. □

LEMMA 3.3. *Proposition 3.1 holds if $m = 1$ or $m + 1 < n$.*

Proof. If $m = 1$, then the nonsingularity of T implies that $T^{-1}(E) \neq 0$ whenever $E \neq 0$, and this is equivalent to the asserted rank property in this case. Let $E \in M_{m,n}$

be given with $3 \leq m+1 < n$. If rank $E = 1$, then dim $\mathcal{T}_{T^{-1}(E)} \leq n+m-2 < 2n-3$. The first inequality is from Lemma 3.2 and the strict inequality is from our assumption that $3 \leq m+1 < n$. Lemma 2.8(c) now shows that $T^{-1}(E)$ must have rank 1. □

LEMMA 3.4. *Let T be a nonsingular linear operator on $M_{m,n}$ that preserves orthogonal equivalence. If $m+1 = n > 3$ or $m = n > 4$, and if $E \in M_{m,n}$ and* rank $E = 1$, *then* rank $T^{-1}(E) \leq 2$.

Proof. If rank $E = 1$, then Lemma 3.2 guarantees that dim $\mathcal{T}_{T^{-1}(E)} \leq m+n-2$. If $m+1 = n > 3$, then $m+n-2 = 2n-3 < 2n-2 \leq 3n-6$. If $m = n > 4$, then $m+n-2 = 2n-2 < 3n-6$. Thus, under the stated hypotheses we have dim $\mathcal{T}_{T^{-1}(E)} < 3n-6$ and hence rank $T^{-1}(E) \leq 2$ by Lemma 2.8(d). □

Let rank $A = 2$ and suppose that $A = [\, a_1\ a_2\ \ldots\ a_n\,] \in M_{m,n}$. There are two possibilities: at least one a_i is not isotropic, in which case we may suppose $a_1^T a_1 \neq 0$, or all of them are isotropic. Moreover, we may assume that $\{a_1, a_2\}$ is linearly independent.

Case 1. $a_1^T a_1 \neq 0$. Let $\alpha = \sqrt{a_1^T a_1}$, so $\alpha \neq 0$. Lemma 2.5 ensures that $Qa_1 = [\, \alpha\ 0\ \ldots\ 0\,]^T$ for some orthogonal $Q \in M_m$, so

$$QA = \begin{bmatrix} \alpha & * \\ 0 & A_1 \end{bmatrix}.$$

If we write $A_1 \equiv [b_2\ b_3\ \ldots\ b_n]$, then $b_2 \neq 0$ since a_1 and a_2 are linearly independent. There are two possibilities: b_2 is isotropic or is not isotropic.

(i) Suppose $b_2^T b_2 \neq 0$. After applying the preceding argument to A_1, we see that there exists an orthogonal $Q_1 \in M_m$ such that

$$B \equiv Q_1 A = \begin{bmatrix} \alpha & \delta & * \\ 0 & \beta & * \\ 0 & 0 & A_2 \end{bmatrix},\ \alpha, \beta, \delta \in \mathbb{C}, \quad \text{with } \alpha\beta \neq 0,$$

but $A_2 = 0$ since rank $A = 2$. For $n \geq m \geq 3$, and referring to the discussion after Definition 2.7, we see that that $\{G_{13}B, G_{23}B\} \cup \mathcal{S}_B$ is a linearly independent subset of $\mathcal{T}_B$. Hence, dim $\mathcal{T}_A$ = dim $\mathcal{T}_B \geq 2n-1 > m+n-2$.

(ii) If $b_2^T b_2 = 0$, a similar argument shows that there is an orthogonal $Q_1 \in M_m$ such that

$$B \equiv Q_1 A = \begin{bmatrix} \alpha & \delta & * \\ 0 & \beta & z \\ 0 & i\beta & iz \\ 0 & 0 & 0 \end{bmatrix},$$

where $z \in M_{1,n-2}$ and $\beta \neq 0$. Let $X = [\, x_1 \ldots\ x_n\,] \in \mathcal{S}_B$. Then the columns of X have the form

$$x_j = \begin{bmatrix} * \\ a_j \\ ia_j \\ 0 \end{bmatrix}, \qquad j = 1, \ldots, n$$

for some $a_j \in \mathbb{C}$. Hence, for $n \geq m \geq 3$, $\{G_{12}B, G_{13}B\} \cup \mathcal{S}_B$ is a linearly independent subset of $\mathcal{T}_B$. Hence, dim $\mathcal{T}_A$ = dim $\mathcal{T}_B \geq 2n-1 > m+n-2$.

Case 2. $a_1^T a_1 = a_2^T a_2 = 0$. Because $a_1^T a_1 = 0$, there exists an orthogonal $Q \in M_m$ and $\alpha \neq 0$ such that

$$QA = \begin{bmatrix} \alpha & \delta & * \\ i\alpha & \beta & * \\ 0 & b_2 & * \end{bmatrix}.$$

Note that the second column of QA is isotropic and independent of the first column.

(i) If $b_2 = 0$, then there is $\delta \neq 0$ such that

$$B \equiv QA = \begin{bmatrix} \alpha & \delta & * \\ i\alpha & -i\delta & * \\ 0 & 0 & 0 \end{bmatrix}.$$

For $n \geq m \geq 3$, $\{G_{13}B, G_{23}B\} \cup \mathcal{S}_B$ is a linearly independent subset of $\mathcal{T}_B$. Hence, $\dim \mathcal{T}_A \geq 2n - 1 > m + n - 2$.

(ii) If $b_2 \neq 0$ and $b_2^T b_2 \neq 0$, there exists an orthogonal $Q_1 \in M_m$ and $\lambda \neq 0$ such that

$$B \equiv Q_1 A = \begin{bmatrix} \alpha & \delta & * \\ i\alpha & \beta & * \\ 0 & \lambda & * \\ 0 & 0 & 0 \end{bmatrix}.$$

For $n \geq m \geq 4$, $\{G_{14}B, G_{34}B\} \cup \mathcal{S}_B$ is a linearly independent subset of $\mathcal{T}_A$, and $\dim \mathcal{T}_A \geq 2n - 1 > m + n - 2$.

(iii) If $b_2 \neq 0$ and $b_2^T b_2 = 0$, there exists an orthogonal $Q_1 \in M_m$ and $\lambda \neq 0$ such that

$$B \equiv Q_1 A = \begin{bmatrix} \alpha & \delta & * \\ i\alpha & \beta & * \\ 0 & \lambda & * \\ 0 & i\lambda & * \\ 0 & 0 & 0 \end{bmatrix}.$$

Just as in Case 1(ii), $\{G_{13}B, G_{23}B\} \cup \mathcal{S}_B \subset \mathcal{T}_B$ is linearly independent. Hence, for $n \geq m \geq 4$, $\dim \mathcal{T}_A = \dim \mathcal{T}_B \geq 2n - 1 > m + n - 2$.

Therefore, if $A \in M_{m,n}$, $n \geq m \geq 4$, and rank $A = 2$, then $\dim \mathcal{T}_A \geq 2n - 1 > m + n - 2 \geq \dim \mathcal{T}_E$ for any $E \in M_{m,n}$ with rank $E = 1$ by Lemma 2.8(a), (b). Combining this result with Lemma 3.4 proves the following lemma.

LEMMA 3.5. *Proposition* 3.1 *holds if* $n > 4$ *and* $m = n$ *or* $m = n - 1$.

Let $\psi = \{(2,2), (2,3), (3,3), (3,4), (4,4)\}$. If $E \in M_{m,n}$ with $n \geq m$ and $(m,n) \notin \psi$, we have shown that rank $T^{-1}(E) = 1$ whenever rank $E = 1$. We treat the five special cases $(m,n) \in \psi$ separately. We use the following two results.

LEMMA 3.6. *Let* $A \in M_n$. *Then* XA *is skew-symmetric for every skew-symmetric* $X \in M_n$ *if and only if* $A = \alpha I$ *for some* $\alpha \in \mathbb{C}$.

Proof. If $A = \alpha I$, then $XA = \alpha X$ is evidently skew-symmetric. Conversely, if X and XA are skew-symmetric, then $XA = -(XA)^T = -(A^T X^T) = A^T X$. Let $A = [a_{ij}]$ and consider the skew-symmetric matrices $F_{1k} = E_{1k} - E_{k1}$ for $k = 1, \ldots n$.

Then

$$F_{1k}A = \begin{bmatrix} a_{k1} & a_{k2} & \cdots & a_{kk} & \cdots & a_{kn} \\ 0 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ -a_{11} & -a_{12} & \cdots & -a_{1k} & \cdots & -a_{1n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}$$

and

$$A^T F_{1k} = \begin{bmatrix} -a_{k1} & 0 & \cdots & a_{11} & \cdots & 0 \\ -a_{k2} & 0 & \cdots & a_{12} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ -a_{kk} & 0 & \cdots & a_{1k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ -a_{nk} & 0 & \cdots & a_{1n} & \cdots & 0 \end{bmatrix}.$$

Hence, $a_{kk} = a_{11}$ and $a_{ki} = 0$ for all $i \neq k$. Thus, $A = a_{11}I$. □

LEMMA 3.7. *Let $A \in M_{m,n}$. If* dim $\mathcal{T}_A$ = dim span $\mathcal{S}_{AQ}$ *for some orthogonal $Q \in M_n$, then $AA^T = \alpha I_m$ for some $\alpha \in \mathbb{C}$.*

Proof. Suppose dim $\mathcal{T}_A$ = dim span $\mathcal{S}_{AQ}$. Since dim $\mathcal{T}_A$ = dim $\mathcal{T}_{AQ}$, we have $\mathcal{T}_{AQ}$ = span $\mathcal{S}_{AQ}$. Hence, for each skew-symmetric $Y \in M_m$ there exists a skew-symmetric $X \in M_n$ such that $YAQ = AQX$. Thus $YAA^T = YAQ^T(AQ^T)^T = (AQ)X(AQ)^T$ is skew-symmetric for every skew-symmetric Y, and hence $AA^T = \alpha I_m$ by Lemma 3.6. □

LEMMA 3.8. *Proposition* 3.1 *holds if* $(m, n) = (2, 3)$.

Proof. Suppose $E \in M_{2,3}$ has rank 1. Since rank $T^{-1}(E) \leq m = 2$, there are only two possibilities: rank $T^{-1}(E) = 1$ (which is the assertion of Proposition 3.1), or rank $T^{-1}(E) = 2$. We wish to exclude the latter possibility.

We look at

$$\mathcal{D} \equiv \{Y \in M_{2,3} : \text{ dim } \mathcal{T}_Y \leq 3\}.$$

Since dim $\mathcal{T}_{T^{-1}(Y)} \leq$ dim $\mathcal{T}_Y$, it must be the case that $T^{-1}(Y) \in \mathcal{D}$ whenever $Y \in \mathcal{D}$. If E has rank 1, then Lemma 2.8(a), (b) show that $E \in \mathcal{D}$. Suppose $X \in \mathcal{D}$ and rank $X = 2$. Then Lemma 2.8(c) ensures that dim $\mathcal{T}_X \geq 3$, so that dim $\mathcal{T}_X = 3$ for this case. Hence, Lemma 2.8(e) ensures that there exists an orthogonal $Q \in M_3$ such that dim span $\mathcal{S}_{XQ} = 3$. Lemma 3.7 now guarantees that $XX^T = \beta^2 I_2$ for some $\beta \in \mathbb{C}$. Moreover, Lemma (4.4) of [5] shows that $\beta \neq 0$. It follows from Lemma 2.5 that there exists an orthogonal $Q_1 \in M_3$ such that

$$X = \begin{bmatrix} \beta & 0 & 0 \\ 0 & \beta & 0 \end{bmatrix} Q_1. \tag{3.2}$$

We will show that $T^{-1}(X) \in \mathcal{O}(\alpha X)$ for some $\alpha \neq 0$. Since $X \in \mathcal{D}$, it suffices to show that $T(E) \notin \mathcal{O}(X)$ for each $E \in \mathcal{E} \equiv \{E_{11}, E_{11} + iE_{12}, E_{11} + iE_{21}, E_{11} - E_{22} + iE_{12} + iE_{21}\}$.

Let

$$A \equiv \begin{bmatrix} \beta & 0 & 0 \\ 0 & \beta & 0 \end{bmatrix},$$

so that $\mathcal{O}(A) = \mathcal{O}(X)$. One checks that

$$\mathcal{T}_A = \left\{ \begin{bmatrix} 0 & x & y \\ -x & 0 & z \end{bmatrix} : x, y, z \in \mathbb{C} \right\}.$$

Let $B \in \mathcal{O}(A) \cap \mathcal{T}_A$. Then

$$\begin{bmatrix} x^2 + y^2 & yz \\ yz & x^2 + z^2 \end{bmatrix} = BB^T = \beta^2 I_2.$$

Hence, $y = z = 0$ and $x^2 = \beta^2$. Thus, $\mathcal{O}(A) \cap \mathcal{T}_A$ consists of exactly two vectors.

Let $E \equiv E_{11} - E_{22} + iE_{12} + iE_{21}$. Then $\dim \mathcal{T}_{T^{-1}(E)} \leq \dim \mathcal{T}_E = 2 < \dim \mathcal{T}_B$ for any $B \notin \{0\} \cup \mathcal{O}(E)$. It follows that $T^{-1}(E) \in \mathcal{O}(E)$ and hence, $T(E) \in \mathcal{O}(E)$. Thus, $T(E) \notin \mathcal{O}(A)$.

Suppose that $T(E_{11}) \in \mathcal{O}(A)$, say $T(E_{11}) = Q_2 A Q_3$. Let $T_1 \equiv Q_2^T T Q_3^T$. Then, $A = Q_2^T T(E_{11}) Q_3^T = T_1(E_{11})$. Note that

$$\mathcal{T}_{E_{11}} = \left\{ \begin{bmatrix} 0 & x & y \\ z & 0 & 0 \end{bmatrix} : x, y, z \in \mathbb{C} \right\}.$$

Thus, $\{E_{12}, E_{13}, E_{21}\} \subset \mathcal{O}(E_{11}) \cap \mathcal{T}_{E_{11}}$ so that $\mathcal{O}(E_{11}) \cap \mathcal{T}_{E_{11}}$ contains at least three vectors. Moreover, $\chi \equiv \{T_1(E_{12}), T_1(E_{13}), T_1(E_{21})\}$ contains three vectors since T_1 is nonsingular. However, $\chi \subset T_1(\mathcal{O}(E_{11}) \cap \mathcal{T}_{E_{11}}) \subset \mathcal{O}(T_1(E_{11})) \cap \mathcal{T}_{T_1(E_{11})} = \mathcal{O}(A) \cap \mathcal{T}_A$, which contains exactly two vectors. This contradiction shows that $T(E_{11}) \notin \mathcal{O}(A)$.

Now let $E \equiv E_{11} + iE_{12}$. Then $E = E(-iF_{12}) \in \mathcal{T}_E$. Since $A \notin \mathcal{T}_A$,we have $T(E) \notin \mathcal{O}(A)$ by Lemma 2.2. Similarly, if $E \equiv E_{11} + iE_{21} = (-iG_{12})E$, then $T(E) \notin \mathcal{O}(A)$.

Thus, $T^{-1}(A)$ cannot have rank 1, and hence it has rank 2 and $T^{-1}(A) \in \mathcal{O}(\alpha A)$ for some $\alpha \neq 0$. It follows that $T(\mathcal{O}(A)) \subset \mathcal{O}(\frac{1}{\alpha} A)$ and hence, $T^{-1}(E) \notin \mathcal{O}(\alpha A)$ for all rank-1 $E \in M_{2,3}$, and all $\alpha \neq 0$. Combining this result with the observation that $T^{-1}(Y) \in \mathcal{D}$ for any $Y \in \mathcal{D}$, we see that rank $T^{-1}(E) = 1$ whenever rank $E = 1$. □

LEMMA 3.9. *Proposition* 3.1 *holds if* $(m, n) = (3, 4)$.

Proof. Let $E \in M_{3,4}$ have rank 1 and let $A \equiv T^{-1}(E)$. Lemma 3.4 shows that rank A is either 1 or 2. Suppose rank $A = 2$. We apply the analysis of the proof of Lemma 3.5 to A. Note that Cases 1(i), (ii) and 2(i), (iii) are not possible here. Thus, A has the form covered in Case 2(ii):

$$A = \begin{bmatrix} \alpha & \delta & x_1 & x_4 \\ i\alpha & \beta & x_2 & x_5 \\ 0 & \lambda & x_3 & x_6 \end{bmatrix}$$

with $\lambda \neq 0$ and all the columns of A are isotropic vectors. It now follows from Lemma 2.8(a), (c) and Lemma 2.1 that $\dim \mathcal{T}_A = 5$. Moreover, Lemma 2.8(e) ensures that $\dim \operatorname{span} \mathcal{S}_{AP} = 5$ for some permutation matrix $P \in M_4$. Hence, Lemma 3.7 guarantees that there exists $a \in \mathbb{C}$ such that $AA^T = aI_3$. Since it is always the case that rank $A \geq$ rank AA^T, we must have $a = 0$, that is, the rows of A are also isotropic vectors. Thus, there exists an orthogonal $Q \in M_4$ such that the first row of AQ is $[\ \alpha\ i\alpha\ 0\ 0\ ]$. Since $AA^T = 0$, the third row of AQ has the form $[\ 0\ 0\ c\ d\ ]$, where $c^2 + d^2 = 0$ and $d \neq 0$. It follows that

$$AQ = \begin{bmatrix} \alpha & i\alpha & 0 & 0 \\ i\alpha & -\alpha & ebd & ed \\ 0 & 0 & bd & d \end{bmatrix}$$

with $e^2 = b^2 = -1$ and $\alpha d \neq 0$. By considering all the possible cases, one checks that $G_{12}AQ \notin \text{span } \mathcal{S}_{AQ}$. We show one such case: $b = e = i$. Suppose that $G_{12}AQ = AQ(a_1F_{12} + a_2F_{13} + a_3F_{14} + a_4F_{23} + a_5F_{24})$. Examining the $(1,1), (2,1), (3,1)$, and $(2,4)$ entries, we get $a_1 = -1, a_2 = a_3 = a_5 = 0$. This is a contradiction since for any a_4, $G_{12}AQ \neq -AQF_{12} + a_4AQF_{23}$. Hence, $\dim \mathcal{T}_A = \dim \mathcal{T}_{AQ} \geq 6$, again a contradiction. Since we can exclude all the possibilities associated with rank $A = 2$, it follows that rank $A = 1$. □

LEMMA 3.10. *Proposition* 3.1 *holds if* $(m,n) = (4,4)$.

Proof. If $A \in M_4$ has rank 2, then $\dim \mathcal{T}_A \geq 7$ as shown on the proof of Lemma 3.5. Let $B \equiv E_{11} - E_{22} + iE_{12} + iE_{21}$. Then $\dim \mathcal{T}_{T^{-1}(B)} \leq \dim \mathcal{T}_B \leq \dim \mathcal{T}_C$ for any $C \notin \{0\} \cup \mathcal{O}(B)$ and hence, $T^{-1}(B) \in \mathcal{O}(B)$ has rank 1. Let $D \equiv \text{diag}\,(1, -1, 1, 1)$. Then $E_{11} + iE_{12} = \frac{1}{2}(B + DB)$ and $T^{-1}(E_{11} + iE_{12}) = \frac{1}{2}(T^{-1}(B) + T^{-1}(DB))$, a sum of two rank 1 matrices. Hence, $T^{-1}(E_{11} + iE_{12})$ has rank at most 2. But since $\dim \mathcal{T}_C \leq 6 < 7 \leq \dim \mathcal{T}_A$ for all $A, C \in M_4$ such that rank $C = 1$ and rank $A = 2$, we must have rank $T^{-1}(C) \neq 2$ so that rank $T^{-1}(E_{11} + iE_{12}) = 1$. Similarly, since $E_{11} + iE_{21} = \frac{1}{2}(B + BD)$ and $E_{11} = \frac{1}{2}((E_{11} + iE_{12}) + (E_{11} + iE_{12})D)$, both $T^{-1}(E_{11} + iE_{21})$ and $T^{-1}(E_{11})$ have rank 1. Thus, if $E \in M_4$ has rank 1, then $T^{-1}(E)$ must also have rank 1. □

Let T be a nonsingular linear orthogonal equivalence preserver on $M_{m,n}$, with $m \leq n$ and $(m,n) \notin \{(2,2),(3,3)\}$. Our arguments up to this point show that T^{-1} preserves rank 1 matrices, that is, $T^{-1}(E)$ has rank 1 whenever $E \in M_{m,n}$ has rank 1. Theorem 1 of [8] guarantees that there exist nonsingular $X \in M_m$ and $Y \in M_n$ such that either $T^{-1}(A) = XAY$ for all $A \in M_{m,n}$, or $m = n$ and $T^{-1}(A) = XA^TY$ for all $A \in M_n$. Hence, either $T(A) = MAN$ for all $A \in M_{m,n}$, or $m = n$ and $T(A) = MA^TN$ for all $A \in M_n$, where $M \equiv X^{-1}$ and $N \equiv Y^{-1}$. We will now show that the same conclusion can be drawn for the two remaining cases $(m,n) = (2,2), (3,3)$, from which Proposition 3.1 follows in these two cases.

LEMMA 3.11. *Let* T *be a nonsingular linear operator on* M_n *that preserves orthogonal equivalence. If* $n = 2$ *or* $n = 3$, *then there exists a scalar* $\alpha \neq 0$ *such that* $T^{-1}(I_n) \in \mathcal{O}(\alpha I_n)$.

Proof. First note that $X_n \equiv E_{11} - E_{22} + iE_{12} + iE_{21} = X_n(iF_{12}) \in \mathcal{T}_{X_n}$ for all $n = 2, 3, \ldots$, where $E_{ij}, X_n \in M_n$. Note also that, $\mathcal{T}_{I_n} = \{X \in M_n : X + X^T = 0\}$ is the set of all skew symmetric matrices in M_n. Hence, $\dim \mathcal{T}_{I_n} = \frac{n(n-1)}{2}$. Moreover, $I_n \notin \mathcal{T}_{I_n}$ for each n.

Let $n = 2$. Then $\dim \mathcal{T}_{I_2} = 1$. It follows from Lemma 2.8(a)–(c) that either $T^{-1}(I_2)$ is nonsingular or $T^{-1}(I_2) \in \mathcal{O}(X_2)$. However, $T^{-1}(I_2) \notin \mathcal{O}(X_2)$ by Lemma 2.2. Thus, $T^{-1}(I_2)$ is nonsingular and Lemma 2.8(e) implies that $\dim \mathcal{T}_{T^{-1}(I_2)} = 1 = \dim \text{span } \mathcal{S}_{I_2Q}$ for some permutation matrix $Q \in M_2$. Lemma 3.7 guarantees that $T^{-1}(I_2) \in \mathcal{O}(\beta I_2)$ for some $\beta \neq 0$.

Let $n = 3$. A similar argument shows that either

$$\text{rank } T^{-1}(I_3) \geq 2 \quad \text{or} \quad T^{-1}(I_3) \in \mathcal{O}(X_3),$$

and that the latter possibility is excluded. Hence, rank $T^{-1}(I_3) \geq 2$. Let $A \equiv T^{-1}(I_3)$. Then $\dim \mathcal{T}_A = 3 = \dim \text{span } \mathcal{S}_{AQ}$ for some permutation matrix $Q \in M_3$. Hence, $AA^T = \alpha I_3$ by Lemma 3.7, and Lemma 4.4 of [5] again guarantees that $\alpha \neq 0$. Therefore, $T^{-1}(I_3) \in \mathcal{O}(\beta I_3)$ with $\beta^2 = \alpha \neq 0$. □

Suppose $n = 2$ or 3. Then Lemma 3.11 ensures that $T^{-1}(I_n) \in \mathcal{O}(\alpha I_n)$. Hence, $T_1 \equiv \alpha T$ satisfies $T_1^{-1}(I_n) \in \mathcal{O}(I_n)$. It follows that $T_1(\mathcal{O}(I_n)) \subset \mathcal{O}(I_n)$. Lemma 1 of [4] guarantees that $T_1(\mathcal{O}(I_n)) = \mathcal{O}(I_n)$. Thus, Lemma 6 of [2] guarantees that there

exist nonsingular $M, N \in M_n$ such that either $T_1(A) = MAN$ or $T_1(A) = MA^TN$ for all $A \in M_n$. It follows that Proposition 3.1 holds for these two cases as well.

The preceding ten lemmata constitute a proof of all cases of Proposition 3.1 in which $m \leq n$. The remaining cases follow from considering $T_1(X) \equiv T(X^T)$ and applying the known cases to $T_1 : M_{n,m} \to M_{n,m}$.

The following proposition summarizes the main conclusions of this section.

PROPOSITION 3.12. *Let T be a nonsingular linear operator on $M_{m,n}$ that preserves orthogonal equivalence. Then there exist nonsingular $M \in M_m$ and $N \in M_n$ such that either*

(1) *$T(A) = MAN$ for all $A \in M_{m,n}$, or*

(2) *$m = n$ and $T(A) = MA^TN$ for all $A \in M_{m,n}$.*

4. Proof of the main theorem. Let T be a given linear operator on $M_{m,n}$. Suppose T preserves orthogonal equivalence. Then Lemma 2.4 guarantees that either $T = 0$ or T is nonsingular. If $T = 0$, then Theorem 1.1 holds with $\alpha = 0$. If $T \neq 0$, we will use the following to show that Theorem 1.1 still holds.

PROPOSITION 4.1. *Let $A \in M_n$ be nonsingular. Suppose that*

$$x^TA^TAx = x^TP^TA^TAPx$$

for all orthogonal $P \in M_n$ and all $x \in \mathbb{C}^n$. Then there exist an orthogonal $Q \in M_n$ and a scalar $\alpha \neq 0$ such that $A = \alpha Q$.

Proof. An easy polarization argument shows that if $C \in M_n$ is symmetric and $x^TCx = 0$ for all $x \in \mathbb{C}^n$, then $C = 0$. Since $x^TA^TAx = x^TP^TA^TAPx$ for all $x \in \mathbb{C}^n$, it follows that $A^TA = P^TA^TAP$ for all orthogonal $P \in M_n$. Hence,

$$A^TA = \begin{bmatrix} \alpha & \beta & \cdots & \beta \\ \beta & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta \\ \beta & \cdots & \beta & \alpha \end{bmatrix}.$$

Let $x \equiv [\ 1\ 1\ 0\ \ldots\ 0\ ]^T$ and $y \equiv [\ \sqrt{2}\ 0\ 0\ \ldots\ 0\ ]^T \in \mathbb{C}^n$. Then $x^Tx = y^Ty$ and hence there exists an orthogonal $Q_1 \in M_n$ such that $y = Q_1x$. Now, $2\alpha + 2\beta = x^TA^TAx = x^TQ_1^TA^TAQ_1x = y^TA^TAy = 2\alpha$. Hence, $\beta = 0$ and $A^TA = \alpha I$ with $\alpha \neq 0$ since A is nonsingular. Thus, $Q \equiv \frac{1}{\sqrt{\alpha}}A$ is orthogonal and $A = \sqrt{\alpha}Q$. □

LEMMA 4.2. *Let T be a given nonsingular linear operator on $M_{m,n}$. Then T preserves orthogonal equivalence if and only if there exist orthogonal matrices $Q_1 \in M_m$, $Q_2 \in M_n$, and a scalar $\beta \neq 0$ such that either*

(1) *$T(A) = \beta Q_1AQ_2$ for all $A \in M_{m,n}$, or*

(2) *$m = n$ and $T(A) = \beta Q_1A^TQ_2$ for all $A \in M_n$.*

Proof. Under the stated assumptions, Proposition 3.12 ensures that there exist nonsingular $M \in M_m$ and $N \in M_n$ such that either $T(A) = MAN$ for all $A \in M_{m,n}$, or $m = n$ and $T(A) = MA^TN$ for all $A \in M_{m,n}$. We consider only the case $T(A) = MAN$; the case $T(A) = MA^TN$ can be dealt with similarly. Let an orthogonal $P \in M_n$ be given. Since T preserves orthogonal equivalence, for each $A \in M_{m,n}$ there exist orthogonal matrices $Q \in M_m$ and $Z \in M_n$ (which depend on A and P) such that $T(A) = QT(AP)Z$. Hence,

$$\begin{aligned} MAN(MAN)^T &= T(A)T(A)^T \\ &= QT(AP)Z(QT(AP)Z)^T \\ &= QMAPN(MAPN)^TQ^T. \end{aligned} \tag{4.1}$$

Choose

$$A \equiv M^{-1} \begin{bmatrix} x^T \\ 0 \end{bmatrix},$$

where $x \in \mathbb{C}^n$. Then (4.1) becomes

$$\begin{bmatrix} x^T NN^T x & 0 \\ 0 & 0 \end{bmatrix} = Q \begin{bmatrix} x^T PNN^T P^T x & 0 \\ 0 & 0 \end{bmatrix} Q^T.$$

Since $Q^T = Q^{-1}$, taking the trace of both sides shows that $x^T NN^T x = x^T PNN^T P^T x$. Since this identity holds for all $x \in \mathbb{C}^n$ and all orthogonal $P \in M_n$, Proposition 4.1 ensures that $N = \alpha_2 Q_2$ for some orthogonal $Q_2 \in M_n$ and some scalar $\alpha_2 \neq 0$. A similar analysis of $T(A)^T T(A)$ shows that $M = \alpha_1 Q_1$ for some orthogonal $Q_1 \in M_m$ and some scalar $\alpha_1 \neq 0$. □

This completes the proof of the forward implication of Theorem 1.1. The converse can be easily verified.

Acknowledgment. It is a pleasure to acknowledge valuable advice from Professor Steven Pierce in the preparation of this paper.

REFERENCES

[1] W. M. Boothby, *An Introduction to Differential Manifolds and Riemannian Geometry*, Academic Press, New York, 1975.

[2] E. P. Botta and S. Pierce, *The preservers of any orthogonal group*, Pacific J. Math., 70 (1977), pp. 37–49.

[3] D. Choudhury and R. A. Horn, *A complex orthogonal-symmetric analog of the polar decomposition*, SIAM J. Alg. Disc. Meth. 8 (1987), pp. 219–225.

[4] J. D. Dixon, *Rigid embedding of simple groups in the general linear group*, Canad. J. Math., XXIX (1977), pp. 384–391.

[5] Y. P. Hong, R. A. Horn, and C. K. Li, *Linear operators preserving t–congruence on matrices*, Linear Algebra Appl., to appear.

[6] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[7] R. A. Horn, C. K. Li, and N. K. Tsing, *Linear operators preserving certain equivalence relations on matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 195–204.

[8] M. Marcus and B. N. Moyls, *Transformations on tensor product spaces*, Pacific J. Math., 9 (1959), pp. 1215–1221.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 530–548, March 1994

012

A PARALLEL ALGORITHM FOR COMPUTING THE SINGULAR VALUE DECOMPOSITION OF A MATRIX*

E. R. JESSUP† AND D. C. SORENSEN‡

Abstract. A parallel algorithm for computing the singular value decomposition of a matrix is presented. The algorithm uses a divide and conquer procedure based on a rank one modification of a bidiagonal matrix. Numerical difficulties associated with forming the product of a matrix with its transpose are avoided, and numerically stable formulae for obtaining the left singular vectors after computing updated right singular vectors are derived. A deflation technique is described that, together with a robust root finding method, assures computation of the singular values to full accuracy in the residual and also assures orthogonality of the singular vectors.

Key words. bidiagonal matrix, singular value decomposition, divide and conquer algorithm

AMS subject classifications. 15A18, 65Y05

1. Introduction.

The singular value decomposition (SVD) of a real $m \times n$ matrix A can be written

$$A = U\Sigma V^T,$$

where U and V are both orthogonal matrices and Σ is a diagonal matrix with nonnegative diagonal elements. The columns of U and V are, respectively, the left and right singular vectors of A; the diagonal elements of Σ are its singular values. A standard algorithm for computing the SVD involves first reducing a matrix A to upper bidiagonal form B using elementary orthogonal transformations [12], [13] as follows:

$$A = \hat{U}B\hat{V}^T$$

and then computing the SVD of $B = \hat{Y}\Sigma\hat{X}^T$. Combining the two results gives

$$A = \hat{U}(\hat{Y}\Sigma\hat{X}^T)\hat{V}^T = U\Sigma V^T,$$

where $U = \hat{U}\hat{Y}$ and $V = \hat{V}\hat{X}$.

This paper focuses on the computation of the SVD of the bidiagonal matrix B by divide and conquer mechanisms based on rank one tearing of the bidiagonal matrix B. Algorithms founded on this technique have proven accurate and efficient for both serial and parallel computation of eigensystems of symmetric tridiagonal matrices [6], [10]. The notable speed and accuracy of the rank one updating process for that problem motivate application of rank one updating techniques to the SVD.

The presentation of the updating technique begins in §2 with a review of the rank one updating techniques used for the symmetric tridiagonal eigenproblem. Section 3 continues with a discussion of some difficulties arising in the design of an SVD

* Received by the editors July 17, 1991; accepted for publication (in revised form) August 13, 1992.

† Department of Computer Science, University of Colorado, Boulder, Colorado 80309-0430 (jessup@cs.colorado.edu). This author was funded by Department of Energy contracts W-31-109-Eng-38 (at Argonne National Laboratory), DE-AC05-84OR21400 (at Oak Ridge National Laboratory), and DE-FG02-92ER25122, and by National Science Foundation grant CCR-9109785.

‡ Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251-1892 (sorensen@rice.edu). This author was funded by Department of Energy contracts W-31-109-Eng-38 (at Argonne National Laboratory) and DE-FG0f-91ER25103 and by National Science Foundation cooperative agreement CCR-9120008.

algorithm that is both accurate and efficient. Section 4 describes a basic divide and conquer step for the SVD equivalent to a rank one tearing of a symmetric tridiagonal matrix.

Sections 5 and 6 are devoted to finite precision deflation rules and the orthogonality of the computed singular vectors. Section 7 covers implementation of the divide and conquer algorithm PSVD and the results of numerical experiments.

In all sections, we consider only the case $m = n$. If $m > n$, the initial reduction may be preceded by computing a QR factorization of A and using the $n \times n$ triangular matrix R in place of A. A similar procedure is appropriate for the case $m < n$.

Throughout this paper, unless otherwise specified, capital Roman letters represent matrices, lower case Roman letters represent column vectors, and lower case Greek letters represent scalars. A superscript T denotes transpose. All matrices and vectors are real.

2. Divide and conquer for the symmetric tridiagonal eigenproblem. In [6], Cuppen presents a divide and conquer technique for finding the eigenvalues and eigenvectors of a symmetric tridiagonal matrix. Rank one tearing is applied to divide the tridiagonal matrix T of order n into

$$T = \begin{pmatrix} \tilde{T}_1 & \beta e_k e_1^T \\ \beta e_1 e_k^T & \tilde{T}_2 \end{pmatrix} = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + \beta \begin{pmatrix} e_k \\ e_1 \end{pmatrix} (e_k^T, \quad e_1^T),$$

where $1 \leq k \leq n$, and e_j represents the jth canonical vector of appropriate length. If the eigensystems of the two submatrices are $T_1 = Q_1 D_1 Q_1^T$ and $T_2 = Q_2 D_2 Q_2^T$, then

$$T = Q \left[D + \beta \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} (z_1^T, \quad z_2^T) \right] Q^T = Q \left[D + \rho z z^T \right] Q^T,$$

where

$$Q = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix}, \qquad D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix},$$

z_1^T is the last row of Q_1, z_2^T is the first row of Q_2, and ρ is chosen so that $\| z \|_2 = 1$. The problem is then reduced to finding the eigensystem of a diagonal matrix plus a rank one change. The eigenvalues of T are equal to the eigenvalues of $D + \rho z z^T$; the eigenvectors of T are the eigenvectors of $D + \rho z z^T$ premultiplied by the matrix Q.

An updating technique described in [5], [6], and [11] is employed to determine the eigensystem of $D + \rho z z^T$. When the diagonal elements of D are distinct and the elements of z are nonzero, the eigenvalues of $D + \rho z z^T$ are equal to the roots of

$$w(\lambda) = 1 + \rho z^T (D - \lambda I)^{-1} z^T$$

and can be determined efficiently by a rational interpolation scheme developed in [5]. The eigenvector corresponding to the ith eigenvalue λ_i is found directly from

$$u_i = (D - \lambda_i I)^{-1} z.$$

When the diagonal elements of $D = \mathrm{diag}(\delta_1, \ldots, \delta_n)$ are not distinct (i.e., $\delta_k = \delta_{k+1} = \cdots = \delta_{k+l}$), the eigenvector basis is rotated to zero out the components $\zeta_{k+1}, \ldots, \zeta_{k+l}$ of z corresponding to the repeated diagonal elements of D [5]. When the jth element of z is zero, the element δ_j is an eigenvalue of $D + \rho z z^T$, and the jth unit vector e_j is its corresponding eigenvector.

Multiple diagonal elements of D and zero elements of z result in significant reduction in the work required to compute the eigensystem of $D+\rho zz^T$. This phenomenon called *deflation* has been refined for use in finite precision arithmetic where *nearly equal* diagonal elements of D and *small* elements of z are deflated [10]. As shown in [6], [10], and [15], substantial deflation and resulting savings in computation time occurs for a wide variety of symmetric tridiagonal eigenproblems. The computed eigensystem is obtained to high accuracy, and the computed eigenvectors are orthogonal [10], [14], and [20].

An experimental comparison in [16] finds the implementation TREEQL [10] of the divide and conquer method and bisection with inverse iteration to be the fastest serial techniques for solving the symmetric tridiagonal eigenproblem. TREEQL is generally fastest when deflation is significant. The QL method (implemented as EISPACK's TQL2 [18]) is generally slowest. All three demonstrate comparable high accuracy in practice, although only TREEQL, TQL2, and bisection can be proven to be backward stable [3], [7]. TREEQL is also fast on shared-memory multiprocessors [10], but is less efficient on statically scheduled distributed-memory multiprocessors [14].

3. Background. The bidiagonal SVD is closely related to the symmetric tridiagonal eigenproblem. For example, the matrix products $\hat{T}_1 = B^TB$ and $\hat{T}_2 = BB^T$ are symmetric tridiagonal matrices of order n having as eigenvalues the squares of the singular values of B and having as eigenvectors, respectively, the left and right singular vectors of B. Thus, one way to determine the SVD of B is to compute the eigendecompositions $\hat{T}_1 = X\Sigma^2X^T$ and $\hat{T}_2 = Y\Sigma^2Y^T$. This approach, however, can be both inefficient and inaccurate. In this section, we review the drawbacks of using eigensolvers to compute $B = Y\Sigma X^T$.

First, finding the eigendecomposition of $\hat{T}_1$ gives only the singular values and the left singular vectors of B. Computing the eigendecomposition of $\hat{T}_2$ gives the left vectors Y but requires redundant computation of the singular value matrix Σ. Because X and Y are computed independently, it is also generally impossible to correctly pair the left and right singular vectors associated with equal or nearly equal singular values. It is preferable to compute each right singular vector using suitable relationships to its corresponding left singular vector.

The vector pairing problem can be overcome by computing the matrix of right singular vectors directly from $X = B^TY\Sigma^{-1}$. This approach fails, however, when Σ has a zero diagonal element. Moreover, numerical experiments have shown an increased residual and degraded orthogonality of right singular vectors computed this way for matrices with large condition numbers. This is particularly disturbing as the SVD is often called upon when a matrix has a large condition number. Attempts to avoid conditioning problems through a combination of the two equations such as

$$(B+\sigma I)x = (B+\sigma I)^T y \tag{1}$$

can fail when there are a significant number of small singular values. However, Arbenz and Golub [2] suggest an iterative procedure using a modified Lanczos process that essentially corrects initial numerical errors made in (1).

Inaccuracies in the small singular values can also result from multiplication of B and its transpose in finite precision arithmetic [13]. For example, suppose that $\mathrm{fl}(1+\epsilon^2) = 1$ in finite precision arithmetic. If

$$B = \begin{pmatrix} 1 & 0 \\ 1 & \epsilon \end{pmatrix},$$

then the computed product $T^T = BB^T$ is

$$\text{fl}\left[\begin{pmatrix} 1 & 0 \\ 1 & \epsilon \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & \epsilon \end{pmatrix}\right] = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

with exact eigenvalues 0 and 2. The computed singular values of B are then 0 and $\sqrt{2}$, while the true singular values are

$$\left(\frac{2\epsilon^2}{2+\epsilon^2+\sqrt{4+\epsilon^4}}\right)^{1/2}$$

and

$$\left(\frac{2+\epsilon^2+\sqrt{4+\epsilon^4}}{2}\right)^{1/2}.$$

For this matrix, the relative error in the smallest computed singular value is 1.

Some existing techniques for the solution of the singular value problem bypass these numerical problems by operating on the matrix B and implicitly forming the product BB^T. The Golub–Reinsch QL method [12], [13] for computing the SVD, for example, has been implemented as the LINPACK routine DSVDC. When using rank one updating techniques, however, it is not convenient to represent the torn matrix as the product of bidiagonal matrices in this way; it is necessary to devise a different way to work with the matrix product implicitly.

A final alternative that permits computation of a correct SVD is to embed the order n bidiagonal matrix in an order $2n$ symmetric banded matrix: the eigenvalues of the $2n \times 2n$ matrix

$$M_1 = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$$

are the singular values of B and their negatives. To compute the SVD of B, the columns and rows of M_1 are permuted to the order $1, n+1, 2, n+2, \ldots, n, 2n$ to form the $2n \times 2n$ tridiagonal matrix M_2 with a zero diagonal. The eigenvector u_i of M_2 corresponding to eigenvalue $\lambda_i = \sigma_i$ has as its odd-numbered components, the components of the ith left singular vector $y_i = (\nu_{1i}, \ldots, \nu_{ni})^T$ and has as its even-numbered components, the components of the ith right singular vector $x_i = (\mu_{1i}, \ldots, \mu_{ni})^T$ [12]:

$$M_2 \begin{pmatrix} \nu_{1i} \\ \mu_{1i} \\ \vdots \\ \nu_{ni} \\ \mu_{ni} \end{pmatrix} = \sigma_i \begin{pmatrix} \nu_{1i} \\ \mu_{1i} \\ \vdots \\ \nu_{ni} \\ \mu_{ni} \end{pmatrix}.$$

Methods for the symmetric tridiagonal eigenproblem are then applied directly to the matrix M_2. This approach is efficient for methods that can take advantage of the zero structure of M_2 and that can compute the first n eigenpairs independently of the second n eigenpairs. Bisection with inverse iteration, for example, falls into this category; the divide and conquer method of [6] and [10] described in §2 does not [15]. A divide and conquer strategy that maintains the zero diagonal in the torn submatrices is described in [1].

The remainder of this paper discusses a method for computing the SVD that is both efficient and stable. It implicitly formulates the matrix product BB^T in a way that avoids cancellation in finite precision arithmetic, and it computes each right singular vector from its corresponding left singular vector. The formulation presented here is the most accurate of several alternatives considered in [17].

4. Divide and conquer for the bidiagonal SVD. This section presents a divide and conquer technique designed for use with the matrix B. It is an efficient alternative to the divide and conquer eigensolver applied to a $2n \times 2n$ tridiagonal matrix. It avoids the numerical difficulties associated with explicit formation of BB^T or B^TB by reformulating the product to prevent cancellation. The algorithm relies on rank one tearing. Specifically, the rank one modification of the matrix B

$$B = \begin{pmatrix} B_1 & \beta e_k e_1^T \\ 0 & B_2 \end{pmatrix} = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} + \beta \begin{pmatrix} e_k \\ 0 \end{pmatrix} (0, e_1^T), \tag{2}$$

where $\beta = \beta_k$ allows implicit formation of BB^T as follows:

$$\begin{aligned} BB^T &= \begin{pmatrix} B_1 & \beta e_k e_1^T \\ 0 & B_2 \end{pmatrix} \begin{pmatrix} B_1^T & 0 \\ \beta e_1 e_k^T & B_2^T \end{pmatrix} \\ &= \begin{pmatrix} B_1 B_1^T & 0 \\ 0 & B_2(I - e_1 e_1^T) B_2^T \end{pmatrix} + \begin{pmatrix} \beta e_k \\ B_2 e_1 \end{pmatrix} (\beta e_k^T, e_1^T B_2^T) \\ &= \begin{pmatrix} B_1 B_1^T & 0 \\ 0 & \hat{B}_2 \hat{B}_2^T \end{pmatrix} + \begin{pmatrix} \beta e_k \\ \alpha e_1 \end{pmatrix} (\beta e_k^T, \alpha e_1^T), \end{aligned} \tag{3}$$

where the matrix

$$\hat{B}_2 = B_2 \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}$$

is the bidiagonal matrix B_2 with its first column replaced by the zero vector and $\alpha e_1 = B_2 e_1$. This splitting may be considered a special case of the general rank one updates to the SVD described in [4].

The SVDs $B_1 = U_1 \Sigma_1 V_1^T$ and $\hat{B}_2 = \hat{U}_2 \hat{\Sigma}_2 \hat{V}_2^T$ can be computed independently and used with (4) to produce

$$\begin{aligned} BB^T &= \begin{pmatrix} U_1 \Sigma_1^2 U_1^T & 0 \\ 0 & \hat{U}_2 \hat{\Sigma}_2^2 \hat{U}_2^T \end{pmatrix} + \begin{pmatrix} \beta e_k \\ \alpha e_1 \end{pmatrix} (\beta e_k^T, \alpha e_1^T), \\ &= \begin{pmatrix} U_1 & 0 \\ 0 & \hat{U}_2 \end{pmatrix} \left[\begin{pmatrix} \Sigma_1^2 & 0 \\ 0 & \hat{\Sigma}_2^2 \end{pmatrix} + \begin{pmatrix} u_1 \\ \hat{u}_2 \end{pmatrix} (u_1^T, \hat{u}_2^T) \right] \begin{pmatrix} U_1^T & 0 \\ 0 & \hat{U}_2^T \end{pmatrix}, \end{aligned} \tag{4}$$

where $u_1 = \beta U_1^T e_k$ and $\hat{u}_2 = \alpha \hat{U}_2^T e_1$. The eigendecomposition of the diagonal plus rank one matrix can be found via the updating techniques derived in [5], [10], and [11] and summarized in §2.

This computation requires that the diagonal elements of the matrix

$$\begin{pmatrix} \Sigma_1^2 & 0 \\ 0 & \hat{\Sigma}_2^2 \end{pmatrix}$$

be distinct and that the elements of $(u_1^T, \hat{u}_2^T)$ be nonzero. When these assumptions do not hold, the problem deflates. However, because the squares of the singular

values less than one are not as well separated as the singular values themselves, the deflation rules of [10] concerning nearly equal diagonal values are not appropriate. To develop deflation rules for the SVD, it is necessary to reformulate the basic step of the updating process and to provide rules based on the original data rather than on the squared data appearing in (4).

To this end, let the $(n-k) \times (n-k-1)$ matrix $\tilde{B}_2$ be defined by

$$(0, \tilde{B}_2) \equiv \hat{B}_2 \equiv B_2 \left(I - e_1 e_1^T\right).$$

Now consider the SVD of $\tilde{B}_2 = \tilde{U}_2 \tilde{\Sigma}_2 \tilde{V}_2^T$, and note that

$$\hat{B}_2 = \begin{pmatrix} \tilde{U}_2 & \tilde{u} \end{pmatrix} \begin{pmatrix} \tilde{\Sigma}_2 \\ 0^T \end{pmatrix} \tilde{V}_2^T,$$

where $\tilde{u}$ is a unit vector orthogonal to the columns of $\tilde{U}_2$. When $B_1 = U_1 \Sigma_1 V_1^T$,

$$B = \begin{pmatrix} U_1 & 0 & 0 \\ 0 & \tilde{U}_2 & \tilde{u} \end{pmatrix} \begin{pmatrix} \Sigma_1 & u_1 & 0 \\ 0 & u_2 & \tilde{\Sigma}_2 \\ 0 & \mu & 0 \end{pmatrix} \begin{pmatrix} V_1^T & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tilde{V}_2^T \end{pmatrix}. \tag{5}$$

For notational convenience, we permute (5) to obtain

$$B = \begin{pmatrix} U_1 & 0 & 0 \\ 0 & \tilde{U}_2 & \tilde{u} \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 & u_1 \\ 0 & \tilde{\Sigma}_2 & u_2 \\ 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} V_1^T & 0 & 0 \\ 0 & 0 & \tilde{V}_2^T \\ 0 & 1 & 0 \end{pmatrix}.$$

Deflation rules are then needed for the interior matrix

$$\bar{M} \equiv \begin{pmatrix} \bar{\Sigma} & \bar{u} \\ 0 & \mu \end{pmatrix} \equiv \begin{pmatrix} \Sigma_1 & 0 & u_1 \\ 0 & \tilde{\Sigma}_2 & u_2 \\ 0 & 0 & \mu \end{pmatrix}, \tag{6}$$

where $\bar{\Sigma} = \operatorname{diag}(\bar{\sigma}_1, \ldots, \bar{\sigma}_{n-1})$ and $\bar{u} = (\bar{\mu}_1, \ldots, \bar{\mu}_{n-1})$. (The matrices of (5) are not explicitly permuted in the implementation of PSVD described in §7.)

The deflation procedure for $\bar{M}$ resembles that for tridiagonal matrices. In exact arithmetic, the problem deflates whenever any of the following cases occur:

1. An element of $\bar{u}$ is zero: $\bar{\mu}_j = 0$.
2. Diagonal elements of $\bar{\Sigma}$ are equal: $\bar{\sigma}_i = \bar{\sigma}_j$, $i \neq j$.
3. A diagonal element of $\bar{\Sigma}$ is zero: $\bar{\sigma}_j = 0$.

It is easily verified that if $\bar{\mu}_j = 0$, $\bar{\sigma}_j$ is a singular value of $\bar{M}$ with left and right singular vectors equal to the jth canonical vector e_j providing the deflation for case 1. The other two cases may be reduced to case 1 using appropriate plane rotations to introduce a zero component in the vector $\bar{u}$.

When $\bar{\sigma}_i = \bar{\sigma}_j$, two-sided rotations are applied as in the tridiagonal case. A plane rotation G_1 in the (i,j)-plane is constructed and applied to $\bar{M}$ (and to the other matrix factors in (6) as well) so that

$$\bar{M} \leftarrow \begin{pmatrix} G_1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \bar{\Sigma} & \bar{u} \\ 0 & \mu \end{pmatrix} \begin{pmatrix} G_1^T & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G_1 \bar{\Sigma} G_1^T & G_1 \bar{u} \\ 0 & \mu \end{pmatrix}.$$

A 2×3 submatrix of $\bar{M}$ is affected as follows:

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \sigma & 0 & \bar{\mu}_i \\ 0 & \sigma & \bar{\mu}_j \end{pmatrix} \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \sigma & 0 & 0 \\ 0 & \sigma & \tau \end{pmatrix},$$

where $\sigma = \bar{\sigma}_i = \bar{\sigma}_j$, with $\tau^2 = (\bar{\mu}_i)^2 + (\bar{\mu}_j)^2$, $c = \bar{\mu}_j/\tau$, and $s = \bar{\mu}_i/\tau$.

When $\bar{\sigma}_i = 0$, a one-sided rotation G_2 in the (i,n)-plane is applied from the left using μ to zero out $\bar{\mu}_i$: In this case

$$\bar{M} \leftarrow G_2 \bar{M}$$

and a 2×2 submatrix is affected as follows:

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \bar{\sigma}_i & \bar{\mu}_i \\ 0 & \mu \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & \tau \end{pmatrix}$$

because $\bar{\sigma}_i = 0$.

In practice, these deflation rules must be modified to accommodate the limitations of finite precision arithmetic. Finite precision rules are given in §7 that apply when $\bar{\Sigma}$ or $\bar{u}$ has small elements or when $\bar{\Sigma}$ has close elements.

Permuting so that all zero elements in the last column are grouped together, the result of deflation is a matrix of the form

$$\bar{M}' = PH\bar{M}G^T P^T = \begin{pmatrix} \hat{\Sigma}_1 & 0 & 0 \\ 0 & \tilde{\Sigma} & u \\ 0 & 0 & \mu \end{pmatrix},$$

where $\tilde{\Sigma}$ has distinct, positive elements, and the vector u has only nonzero elements. P is the appropriate permutation matrix, and G and H are matrices consisting of accumulated products of the rotations constructed at each of the deflation steps.

After deflation, one only needs to compute the SVD of

$$M \equiv \begin{pmatrix} \tilde{\Sigma} & u \\ 0 & \mu \end{pmatrix} \equiv Y \Sigma X^T. \tag{7}$$

The diagonal elements of $\hat{\Sigma}_1$ are taken as singular values of $\bar{M}$ with appropriate canonical vectors as singular vectors. The squares of the singular values of M and its left singular vectors are given by the eigendecomposition

$$Y\Sigma^2 Y^T \equiv MM^T \equiv \begin{pmatrix} \tilde{\Sigma}^2 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} u \\ \mu \end{pmatrix} (u^T, \quad \mu).$$

An eigenvalue σ^2 of MM^T is a root of the secular equation

$$f(\sigma^2) = 1 + u^T(\tilde{\Sigma}^2 - \sigma^2 I)^{-1} u - \left(\frac{\mu}{\sigma}\right)^2 \tag{8}$$

and can be computed using the root-finder from [5]. If the sorted diagonal elements of $\mathrm{diag}(\tilde{\Sigma}^2, 0)$ are $0 = \tilde{\sigma}_1^2 < \tilde{\sigma}_2^2 < \cdots < \tilde{\sigma}_n^2$, the jth eigenvalue σ_j^2 of MM^T lies in the interval $(\tilde{\sigma}_j^2, \tilde{\sigma}_{j+1}^2)$ [5], and all eigenvalues are positive. The jth singular value of B is σ_j, and the left singular vector of B associated with σ_j for $j = 1, \ldots, n$ is

$$y_j = \begin{pmatrix} (\tilde{\Sigma}^2 - \sigma_j^2 I)^{-1} u \\ -\mu/\sigma_j^2 \end{pmatrix} \theta,$$

where θ is a normalization factor. The corresponding right singular vector is

$$x_j = \frac{M^T y_j}{\| M^T y_j \|_2}.$$

That is, a vector in the direction of the right singular vector x_j is given by

$$M^T y_j = \begin{pmatrix} \tilde{\Sigma}^2 & 0 \\ u^T & \mu \end{pmatrix} \begin{pmatrix} (\tilde{\Sigma}^2 - \sigma_j^2 I)^{-1} u \\ -\mu/\sigma_j^2 \end{pmatrix} \theta$$

$$= \begin{pmatrix} \tilde{\Sigma}^2 (\tilde{\Sigma}^2 - \sigma_j^2 I)^{-1} u \\ u^T (\tilde{\Sigma}^2 - \sigma_j^2 I)^{-1} u - \mu/\sigma_j^2 \end{pmatrix} \theta.$$

Recall from (8) that σ_j satisfies

$$u^T (\tilde{\Sigma}^2 - \sigma_j^2 I)^{-1} u + \left(\frac{\mu}{\sigma_j}\right)^2 = -1.$$

Thus, the quantities σ_j, x_j, and y_j can be computed as follows

PROCEDURE 4.1 (SOLUTION OF THE DEFLATED UPDATING PROBLEM).

1. Solve (8) for σ_j.
2. $y = \begin{pmatrix} v_j \\ \eta_j \end{pmatrix} = \begin{pmatrix} (\tilde{\Sigma}^2 - \sigma_j^2 I)^{-1} u \\ -\mu/\sigma_j^2 \end{pmatrix}$.
3. $x = \begin{pmatrix} \tilde{\Sigma}^2 v_j \\ -1 \end{pmatrix}$.
4. $y_j = \frac{y}{\|y\|_2}$, $\quad x_j = \frac{x}{\|x\|_2}$.

The orthogonality of the singular vectors computed according to this procedure is examined in §6.

The singular values of $B = \hat{Y}\Sigma\hat{X}^T$ are those of $\bar{M}$ and its singular vectors are derived from those of $\bar{M}$. Specifically,

$$\hat{\Sigma} = \begin{pmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \Sigma \end{pmatrix},$$

$$\hat{Y} = \begin{pmatrix} U_1 & 0 & 0 \\ 0 & \tilde{U}_2 & \tilde{u} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & Y \end{pmatrix},$$

and

$$\begin{pmatrix} V_1^T & 0 & 0 \\ 0 & 0 & \tilde{V}_2^T \\ 0 & 1 & 0 \end{pmatrix}^T \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix},$$

where I is the identity matrix of the same order as $\hat{\Sigma}_1$.

5. Deflation rules for finite precision arithmetic. Section 4 gives rules for deflating the problem when the matrix $\bar{M} = \begin{pmatrix} \bar{\Sigma} & \bar{u} \\ 0 & \mu \end{pmatrix}$ from (7) when $(\bar{\Sigma} \quad \bar{u})$ has equal diagonal elements, $\bar{\sigma}_i = \bar{\sigma}_j$, $i \neq j$, or zeros in the last column, $\bar{\mu}_j = 0$. As in the tridiagonal case, these rules can be extended to deflate the problem when $\bar{M}$ has *close* diagonal elements, $\bar{\sigma}_i \approx \bar{\sigma}_j$, or *small* elements $|\bar{\mu}_j| < \epsilon$ for some small positive value of ϵ. In this section, we present deflation rules for finite precision arithmetic followed by an analysis to show that the errors imposed by deflation are small.

The finite precision rules are summarized in Procedure 5.1 below. They follow the same three basic steps as in exact arithmetic, but small elements of $\bar{u}$ or $\bar{\Sigma}$ are approximated as zero and close diagonal elements of $\bar{\Sigma}$ are approximated as equal. To keep track of deflation, two lists are used: *deflate_list* holds the indices of all small elements in the last column of the matrix, and *solve_list* holds the remaining indices. The list *solve_list* is initialized with the indices $1, \ldots, n-1$ of all diagonal elements in $\bar{\Sigma}$. When deflation is complete, the list *deflate_list* holds the indices of the diagonal elements of the transformed matrix to be accepted as singular values of $\bar{M}$, and *solve_list* holds the indices of the rows of $\bar{M}$ retained in the deflated matrix.

PROCEDURE 5.1 (DEFLATION IN FINITE PRECISION ARITHMETIC).

1. *For all* $k \in$ *solve_list,*
   if $|\bar{\mu}_k| < \epsilon$, *move* k *from solve_list to deflate_list.*
2. *Permute the indices so the elements of solve_list are increasing adjacent integers and* $\bar{\sigma}_k \leq \bar{\sigma}_{k+1}$. *(That is, replace* M *by* $P^T M P$ *such that* $Pe_n = e_n$ *with* $k' \in$ *deflate_list for* $1 \leq k' \leq |$*deflate_list*$|$ *and* $k' \in$ *solve_list for* $|$*deflate_list*$| + 1 \leq k' \leq n-1$.)
3. *For all but the last* $k \in$ *solve_list,*
   if $|\bar{\sigma}_k - \bar{\sigma}_{k+1}|$ *is small, apply a two-sided plane rotation so that* $\mu_k \leftarrow 0$, *and move* k *from solve_list to deflate_list.*
   {*In PSVD, the two-sided rotation is applied as follows:*
   $\tau^2 = (\bar{\mu}_k^2 + \bar{\mu}_{k+1}^2)$
   if $|(\bar{\sigma}_k - \bar{\sigma}_{k+1})\bar{\mu}_k\bar{\mu}_{k+1}| < \epsilon\tau^2$ *then*
   $c = \bar{\mu}_{k+1}/\tau$ *and* $s = \bar{\mu}_k/\tau$
   $\bar{\sigma}_k \leftarrow c^2\bar{\sigma}_k + s^2\bar{\sigma}_{k+1}$
   $\bar{\sigma}_{k+1} \leftarrow s^2\bar{\sigma}_k + c^2\bar{\sigma}_{k+1}$
   $\bar{\mu}_{k+1} \leftarrow \tau$
   $\bar{\mu}_k \leftarrow 0$ }
4. *For* $k \in$ *solve_list,*
   if $|\bar{\sigma}_k|$ *is small, apply a one-sided plane rotation so that* $\mu_k \leftarrow 0$, *and move* k *from solve_list to deflate_list.*
   {*In PSVD, the one-sided rotation is applied as follows:*
   $\tau^2 = (\bar{\mu}_k^2 + \mu^2)$
   if $|\bar{\sigma}_k\mu_k| < \epsilon\tau^2$ *then*
   $c = \bar{\mu}_k/\tau$ *and* $s = \frac{\mu}{\tau}$
   $\bar{\sigma}_k \leftarrow c\bar{\sigma}_k$
   $\mu \leftarrow \tau$
   $\bar{\mu}_k \leftarrow 0$ }

We now examine the errors introduced by Procedure 5.1. The first source of error is the transformation of the matrix at steps 2 and 3. The exact result of one of the two-sided rotations would be

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \bar{\sigma}_k & 0 & \bar{\mu}_k \\ 0 & \bar{\sigma}_{k+1} & \bar{\mu}_{k+1} \end{pmatrix} \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \hat{\sigma}_k & 0 & 0 \\ 0 & \hat{\sigma}_{k+1} & \tau \end{pmatrix} + cs(\hat{\sigma}_k - \hat{\sigma}_{k+1}) \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \tag{9}$$

where $\hat{\sigma}_k = c^2\bar{\sigma}_k^2 + s^2\sigma_{k+1}$ and $\hat{\sigma}_{k+1} = s^2\bar{\sigma}_k^2 + c^2\sigma_{k+1}$. In step 2, the second term in the matrix sum in (9) is set to zero. Thus, with each rotation an error is imposed of the form

$$E_k^{(2)} = \epsilon_k \left(e_{k+1}{e_k}^T + e_k{e_{k+1}}^T\right),$$

where $\epsilon_k = (\bar{\sigma}_k - \bar{\sigma}_{k+1})\bar{\mu}_k\bar{\mu}_{k+1}/(\bar{\mu}_k^2 + \bar{\mu}_{k+1}^2)$. The test to decide when rotations should be performed guarantees that $|\epsilon_k| < \epsilon$ for all k. Similarly, the exact result of one of the one-sided rotations would be

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix}\begin{pmatrix} \bar{\sigma}_k & \bar{\mu}_k \\ 0 & \mu \end{pmatrix} = \begin{pmatrix} \hat{\sigma}_k & 0 \\ 0 & \tau \end{pmatrix} - s\bar{\sigma}_k \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

so that each rotation in step 3 causes an error of the form

$$E_k^{(3)} = \epsilon_k \left(e_n{e_k}^T\right),$$

with $|\epsilon_k| < \epsilon$ for all k.

As in the exact case, the resulting matrix may then be permuted to form

$$\begin{pmatrix} \hat{\Sigma}_1 & & \bar{u}_1 \\ 0 & \tilde{\Sigma} & u \\ 0 & 0 & \mu \end{pmatrix} + E_1 = \begin{pmatrix} \hat{\Sigma}_1 & & 0 \\ 0 & \tilde{\Sigma} & u \\ 0 & 0 & \mu \end{pmatrix} + E, \tag{10}$$

where

$$E = \begin{pmatrix} 0 & 0 & \bar{u}_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + E_1 .$$

In these equations, E_1 is a matrix with elements bounded by ϵ, $\bar{u}_1 = \{\bar{\mu}_k | k \in$ *deflate_list*$\}$, and $u = \{\bar{\mu}_k | k \in$ *solve_list*$\}$. Thus, $|e_i^T\bar{u}_1| < \epsilon$ for all i and $|e_i^T u| \geq \epsilon$ for all i.

We approximate the vector $\bar{u}_1$ as zero, accept the diagonal elements of $\hat{\Sigma}$ as singular values of $\bar{M}$, and compute the SVD of the deflated submatrix

$$\begin{pmatrix} \tilde{\Sigma} & u \\ 0 & \mu \end{pmatrix}$$

by the procedure outlined in §4. Because the constituent errors are small, the SVD of $\bar{M}$ computed in this way is the exact SVD of a matrix close to $\bar{M}$.

The magnitude of the total error E depends on the deflation tolerance ϵ. A reasonable choice for ϵ is *macheps* $\times\ \tilde{\sigma}_{\max}$ where *macheps* is machine precision and $\tilde{\sigma}_{\max}$ is the largest diagonal element of $\tilde{\Sigma}$ in the undeflated matrix $\bar{M}$. Because the accurate determination of the small singular values may be important, however, it is also possible to vary the tolerance at each step of deflation according to the size of the singular value at that deflation step.

6. Orthogonality of the singular vectors. Let us now consider the possible limitations on orthogonality of singular vectors due to nearly equal singular values. Many of the results concerning this issue for the symmetric eigenvalue problem apply directly. In particular, the recent results of Sorensen and Tang [20] concerning the numerical orthogonality of the computed eigenvectors will apply. Our first result in

this section is a perturbation lemma that demonstrates the inherent difficulty with nearly equal roots.

For purposes of this discussion, we denote the diagonal elements of $\tilde{\Sigma}$ of the deflated matrix M by $\tilde{\sigma}_j$ and the corresponding components of the vector u by μ_j for $j = 1, 2, \ldots, n-1$ so that the secular equation (8) becomes

$$f(\sigma^2) = 1 + \sum_{j=1}^{n-1} \frac{\mu_j^2}{\tilde{\sigma}_j^2 - \sigma^2} - \frac{\mu^2}{\sigma^2}. \tag{11}$$

Let

$$y_\sigma^T = \left(\frac{\mu_1}{\tilde{\sigma}_1^2 - \sigma^2}, \frac{\mu_2}{\tilde{\sigma}_2^2 - \sigma^2}, \ldots, \frac{\mu_n - 1}{\tilde{\sigma}_{n-1}^2 - \sigma^2}, \frac{\mu}{-\sigma^2} \right) \left[\frac{1}{f'(\sigma^2)} \right]^{1/2}, \tag{12}$$

and let

$$x_\sigma^T = \left(\frac{\tilde{\sigma}_1 \mu_1}{\tilde{\sigma}_1^2 - \sigma^2}, \frac{\tilde{\sigma}_1 \mu_2}{\tilde{\sigma}_2^2 - \sigma^2}, \ldots, \frac{\tilde{\sigma}_1 \mu_{n-1}}{\tilde{\sigma}_{n-1}^2 - \sigma^2}, -1 \right) \left[\frac{1}{1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \sigma^2)^2}} \right]^{1/2}. \tag{13}$$

Note that the squares of the singular values of M are roots of f and that the unit vectors x_σ and y_σ are just permuted left and right singular vectors of M corresponding to σ when σ^2 is a root of f.

LEMMA 6.1. *Let y_σ and x_σ be given by* (12) *and* (13), *respectively. Then for any* $\sigma, \gamma \notin \{\tilde{\sigma}_i : i = 1, \ldots, n-1\} \cup \{0\}$,

$$|y_\sigma^T y_\gamma| = \frac{1}{|\sigma^2 - \gamma^2|} \frac{|f(\sigma^2) - f(\gamma^2)|}{(f'(\sigma^2) f'(\gamma^2))^{1/2}}, \tag{14}$$

and

$$|x_\sigma^T x_\gamma| = \left| \frac{\sigma^2 (f(\sigma^2) - f(\gamma^2))}{(\sigma^2 - \gamma^2)} + f(\gamma^2) \right| \times \left[\left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \sigma^2)^2} \right) \left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \gamma^2)^2} \right) \right]^{-1/2}. \tag{15}$$

Proof. Equation (14) is just the result implied by Lemma 4.2 in [10]. To derive (15), we note that

$$\begin{aligned} & 1 + \sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \\ &= 1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j^2 - \sigma^2)\mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} - \frac{\mu^2}{\gamma^2} + \frac{\sigma^2 \mu^2}{\sigma^2 \gamma^2} + \sigma^2 \sum_{j=1}^{n-1} \frac{\mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \\ &= 1 + \sum_{j=1}^{n-1} \frac{\mu_j^2}{\tilde{\sigma}_j^2 - \gamma^2} - \frac{\mu^2}{\gamma^2} + \sigma^2 \left(\frac{\mu^2}{\sigma^2 \gamma^2} + \sum_{j=1}^{n-1} \frac{\mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \right) \\ &= f(\gamma^2) + y_\sigma^T y_\gamma \left(f'(\sigma^2) f'(\gamma^2) \right)^{1/2}. \end{aligned} \tag{16}$$

Equation (15) follows from (14) applied to the second term in (16). □

Note that in (12) and (13), y_σ and x_σ are always vectors of unit length and that the set of n vectors selected by setting σ^2 equal to the roots of (11) provides the set of left and right singular vectors for the deflated matrix M in (7). Moreover, (14) shows that the set of left vectors are mutually orthogonal, and (15) shows that the set of right vectors are mutually orthogonal whenever σ^2 and γ^2 are set to distinct roots of f. Finally, the term $|\sigma^2 - \gamma^2|$ appearing in the denominator of (14) warns that it may be difficult to attain orthogonal singular vectors when the roots σ and γ are close.

We now wish to examine the situation of close roots. With an argument similar to that given in Lemma 4.6 of [10], one can show that $\tilde{\sigma}_i - \sigma$ must be bounded away from zero due to the deflation process. Because of this, we can expect to compute the differences $\tilde{\sigma}_j - \sigma$ to high relative accuracy. This is quite important with regard to orthogonality of the computed singular vectors as the following lemma shows.

LEMMA 6.2. *Suppose that $\hat{\sigma}^2$ and $\hat{\gamma}^2$ are numerical approximations to exact roots σ^2 and γ^2 of f. Assume that these roots are distinct and let the relative errors for the quantities $\tilde{\sigma}_i - \sigma$ and $\tilde{\sigma}_i - \gamma$ be denoted by θ_i and η_i, respectively. That is, the computed differences are*

$$\tilde{\sigma}_i^2 - \hat{\sigma}^2 = (\tilde{\sigma}_i^2 - \sigma^2)(1+\theta_i) \quad \text{and} \quad \tilde{\sigma}_i^2 - \hat{\gamma}^2 = (\tilde{\sigma}_i{}^2 - \gamma^2)(1+\eta_i), \tag{17}$$

for $i = 1, 2, \ldots, n$. Let $y_{\hat{\sigma}}$ and $y_{\hat{\gamma}}$ be defined according to (12), *and let $x_{\hat{\sigma}}$ and $x_{\hat{\gamma}}$ be defined according to* (13) *using the computed quantities given in* (17). *If $|\theta_i|, |\eta_i| \leq \epsilon << 1$, then*

$$|y_{\hat{\sigma}}^T y_{\hat{\gamma}}| \leq \epsilon(2+\epsilon)\left(\frac{1+\epsilon}{1-\epsilon}\right)^2$$

and

$$|x_{\hat{\sigma}}^T x_{\hat{\gamma}}| \leq \frac{\epsilon(2+\epsilon)}{(1-\epsilon)^2}.$$

Proof. A proof of the bound on the inner product of the left vectors is given in Lemma 4.7 of [10]. For the right vectors, note that

$$\begin{aligned}
&1 + \sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(1+\theta_j)(\tilde{\sigma}_j^2 - \gamma^2)(1+\eta_j)} \\
&= -\sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} + \sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(1+\theta_j)(\tilde{\sigma}_j^2 - \gamma^2)(1+\eta_j)} \\
&= -\sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \left(\frac{\theta_j + \eta_j + \theta_j \eta_j}{(1+\theta_j)(1+\eta_j)}\right).
\end{aligned}$$

Thus,

$$|x_{\hat{\sigma}}^T x_{\hat{\gamma}}| = \left| \sum_{j=1}^{n-1} \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \left(\frac{\theta_j + \eta_j + \theta_j \eta_j}{(1+\theta_j)(1+\eta_j)}\right) \right.$$

$$\cdot \left[\left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \sigma^2)^2} \right) \left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \gamma^2)^2} \right) \right]^{-1/2} \Bigg|$$

$$\leq \left| \left(\sum_{j=1}^{n-1} \left| \frac{\tilde{\sigma}_j^2 \mu_j^2}{(\tilde{\sigma}_j^2 - \sigma^2)(\tilde{\sigma}_j^2 - \gamma^2)} \right| \right) \left(\frac{\epsilon(2+\epsilon)}{(1-\epsilon)^2} \right) \right.$$

$$\cdot \left[\left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \sigma^2)^2} \right) \left(1 + \sum_{j=1}^{n-1} \frac{(\tilde{\sigma}_j \mu_j)^2}{(\tilde{\sigma}_j^2 - \gamma^2)^2} \right) \right]^{-1/2} \Bigg|$$

$$\leq \left(\frac{\epsilon(2+\epsilon)}{(1-\epsilon)^2} \right),$$

and the lemma is proved. □

This lemma shows that orthogonality can be assured whenever it is possible to provide small relative errors in the computed differences $\tilde{\sigma}_j - \sigma$. The results of [20] indicate that this condition may be achieved in practice.

We have applied the root-finder used in [10] directly to (11) and have taken the square root of σ^2 to get a singular value without any apparent difficulty even when very difficult problems were solved. Nevertheless, it is certainly conceivable that problems could arise. Namely, it may be necessary to further refine the root-finding process to prevent loss of accuracy in terms of the form

$$\frac{\mu_j^2}{\tilde{\sigma}_j^2 - \sigma^2},$$

when $\tilde{\sigma}_j$ is small and σ is near to σ_j.

Both of the following two modifications to the root-finder remove the dependence on σ^2 from all terms other than the term $\frac{\mu}{\sigma^2}$. First, one could use

$$\frac{\mu_j^2}{\tilde{\sigma}_j^2 - \sigma^2} = \frac{\mu_j^2}{(\tilde{\sigma}_j + \sigma)(\tilde{\sigma}_j - \sigma)}$$

and update both $\tilde{\sigma}_j - \sigma$ and $\tilde{\sigma}_j + \sigma$ to avoid unnecessary cancellation caused by the squaring. Alternatively, one could use

$$\frac{\mu_j^2}{\tilde{\sigma}_j^2 - \sigma^2} = \frac{\mu_j^2}{2\tilde{\sigma}_j} \left(\frac{1}{(\tilde{\sigma}_j + \sigma)} + \frac{1}{(\tilde{\sigma}_j - \sigma)} \right),$$

and the results of [20] would apply directly to the evaluation of f in this form.

If either of these last two schemes are employed, then it is possible to show that the hypothesis (17) of Lemma 6.2 will be satisfied when the differences $\tilde{\sigma}_j - \sigma$ are computed to high relative accuracy.

7. Experimental results. In this section, we present computational results from the implementation PSVD of the divide and conquer method developed in §§4–5. PSVD splits the matrix B recursively into submatrices of order 8 and solves the subproblems using DSVDC if B is of order 16 or greater and calls DSVDC without matrix splitting otherwise. DSVDC is the fastest serial method for solving the problems of very small order [15]. The results reported here concern only the computation of the

SVD of a bidiagonal matrix B and not reduction of a general matrix to bidiagonal form. In all cases, the full sets of left and right singular vectors were computed along with the singular values.

The first set of experiments include serial timings and accuracy tests. These were carried out in double precision on a single Sequent Symmetry S81 processor using the Weitek floating point accelerator. On this machine, *macheps*$= 2.22 \times 10^{-16}$. We compare the results from PSVD to those from the LINPACK code DSVDC [8] and the implementation of bisection and inverse iteration B/III developed in [16].

The five bidiagonal matrices tested are introduced below. A pair of computed eigenvalues $\hat{\lambda}_i$, $\hat{\lambda}_{i+1}$ belong to a *cluster* if $\hat{\lambda}_i - \hat{\lambda}_{i+1} \leq 10^{-14}|\hat{\lambda}|_{\max}$.

1. Matrix [2,1]: All diagonal elements are 2, and all off-diagonal elements are 1. All singular values lie within the interval [1,3]. For all tested matrix orders, the singular values are computationally distinct.

2. Random: These matrices have uniformly distributed random entries between -1 and 1 generated by the uniform pseudorandom number generator RAND available from NETLIB on both diagonal and off-diagonal elements. The matrices tested turn out to have singular values with minimum magnitude $O(10^{-5})$.

3. B_W: Inspired by the Wilkinson matrix W^+ [21], this matrix of even order has diagonal elements $\frac{n}{2}, \ldots, 1, 1, \ldots, \frac{n}{2}$ and all off-diagonal elements equal to one. Its smallest singular value is $O(10^{-3})$, and, in finite precision, its largest singular values have multiplicity two for matrix orders of about ten and larger.

4. Matrix $[2, u]/n$: The matrix $[2, u]/n$ of order n has the value $2/n$ in each off-diagonal position and the value $\frac{i}{n}$ in the ith diagonal position. This matrix is ill conditioned and has one singular value less than 10^{-14} for orders greater than eighty.

5. Modified matrix [2,1]: This matrix is formed from matrix [2,1] by setting the sixth through ninth diagonal elements $\alpha_6, \ldots, \alpha_9$ and fifth through eighth off-diagonal elements $\beta_5, \ldots, \beta_8$ equal to 10^{-14}. This matrix is severely ill conditioned, having between four and eight singular values less than 10^{-8} and between two and four singular values less than 10^{-14} for all tested orders.

Let $\hat{X}\hat{\Sigma}\hat{Y}^T$ denote the computed SVD of B. To determine the accuracy of the result, we measure the residual error and the deviation from orthogonality of both sets of singular vectors:

$$\mathcal{R} = \frac{1}{\hat{\sigma}_1} \max_i \| B\hat{x}_i - \hat{\sigma}_i \hat{y}_i \|_2,$$
$$\mathcal{O}_X = \| \hat{X}^T\hat{X} - I \|_\infty,$$
$$\mathcal{O}_Y = \| \hat{Y}^T\hat{Y} - I \|_\infty.$$

When all of these quantities are small, the computed SVD of B is nearly the SVD of a matrix near B [15]. In other words, $(\hat{X} + \delta\hat{X})\hat{\Sigma}(\hat{Y} + \delta\hat{Y})^T$ is the exact SVD of a matrix $B + E$ with small $\delta\hat{X}$, $\delta\hat{Y}$, and E.

Table 1 shows the greatest residual and deviation from orthogonality measured for the five test problems solved by B/III, PSVD, and DSVDC for matrix orders 32, 100, and 200. Even for ill-conditioned matrices, PSVD attains full orthogonality of singular vectors and a small residual. Although only PSVD, DSVDC, and bisection (B) are provably stable methods, all three methods tested achieve similarly good results.

The run times for computing the full SVDs of the five matrices are compared in Figs. 1–4 for problem [2,1] and B_W. (These are representative samples from the

TABLE 1
Maximum residual and orthogonalities of SVDs *computed by* B/III, PSVD, *and* DSVDC *for the five test matrices.*

Matrix order	Method	Maximum residual $\mathcal{R}$	Maximum orthogonality (left) $\mathcal{O}_Y$	Maximum orthogonality (right) $\mathcal{O}_X$
n = 32	PSVD	1.66d–14	7.65d–15	7.54d–15
	DSVDC	1.79d–15	1.13d–14	1.13d–14
	B/III	9.77d–16	9.76d–15	1.02d–14
n = 100	PSVD	9.39d–14	2.56d–14	2.37d–14
	DSVDC	4.22d–15	2.68d–14	2.86d–14
	B/III	2.38d–15	1.90d–14	1.87d–14
n = 200	PSVD	4.09d–15	1.13d–14	1.64d–14
	DSVDC	7.60d–15	8.13d–14	8.14d–14
	B/III	5.99d–15	5.42d–13	5.53d–14

TABLE 2
The number of roots computed by PSVD *for five bidiagonal matrices.*

Matrix	Order	PSVD: Fraction of roots computed
[2, 1]	32	1.0
	100	0.9
	200	0.8
B_W	32	0.6
	100	0.6
	200	0.5

full set presented in [15].) For small order problems, PSVD is consistently the fastest method except when PSVD simply calls DSVDC, and B/III is the slowest. For larger order problems (greater than about 50), performance depends more strongly on the matrix characteristics. Namely, when significant deflation occurs, PSVD is the fastest of the three methods. Table 2 shows the fraction of singular values actually computed (as opposed to deflated) by PSVD for matrices [2,1] and B_W. The greater degree of deflation for B_W accounts for the lower runtime of PSVD relative to B/III for B_W.

Similar tests were performed on an Alliant FX/8. The operating system was Concentrix 3.0 and the optimization level for the subroutines comprising the units of computation were options -Ogv. Parallelism was invoked and controlled explicitly through the use of the SCHEDULE package [19]. Implementation details are quite similar to the those for the symmetric eigenvalue routine TREEQL [8]. The recursive matrix splitting leads to a hierarchy of subproblems with a data dependency graph

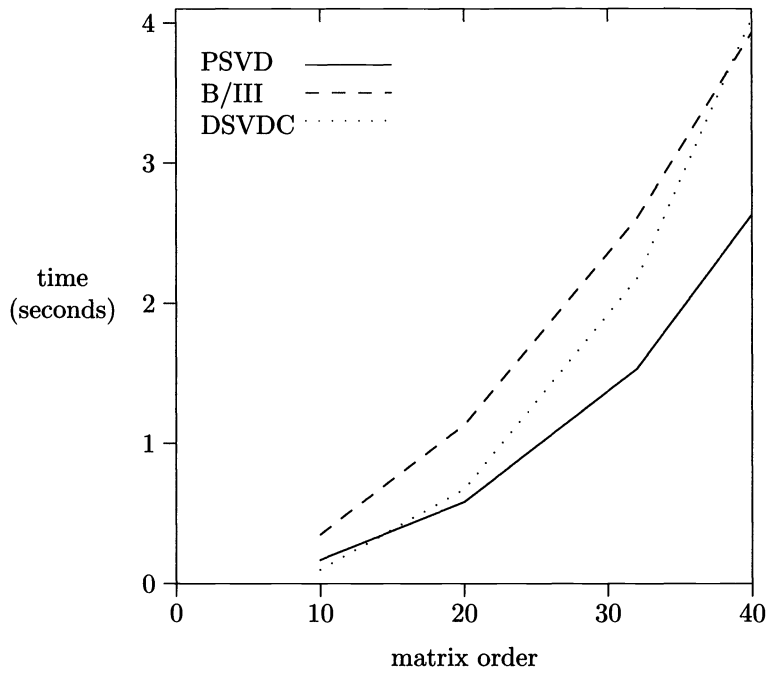


FIG. 1. *Times for computation of the* SVD *by* B/III, PSVD, *and* DSVDC *versus matrix order for matrix* [2,1].

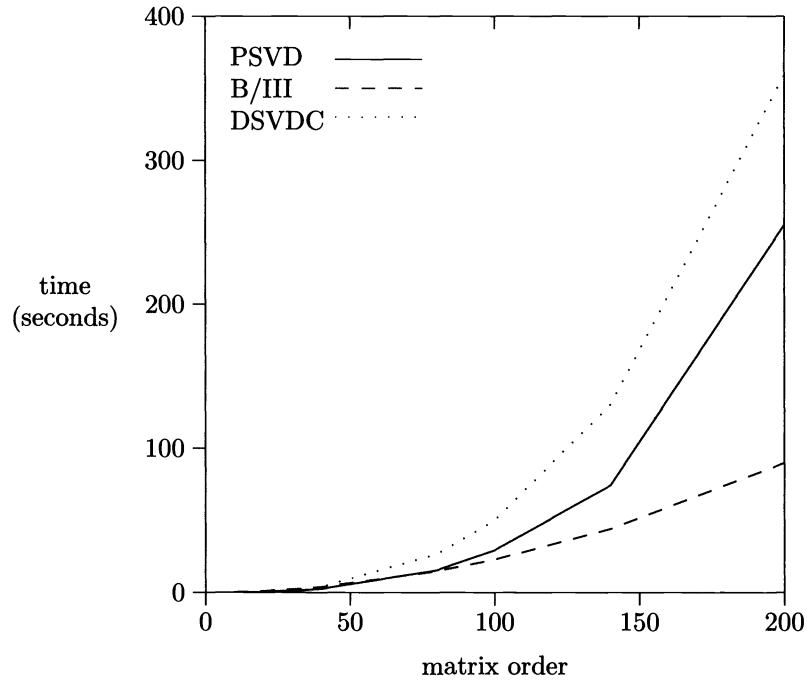


FIG. 2. *Times for computation of the* SVD *by* B/III, PSVD, *and* DSVDC *versus matrix order for matrix* [2,1].

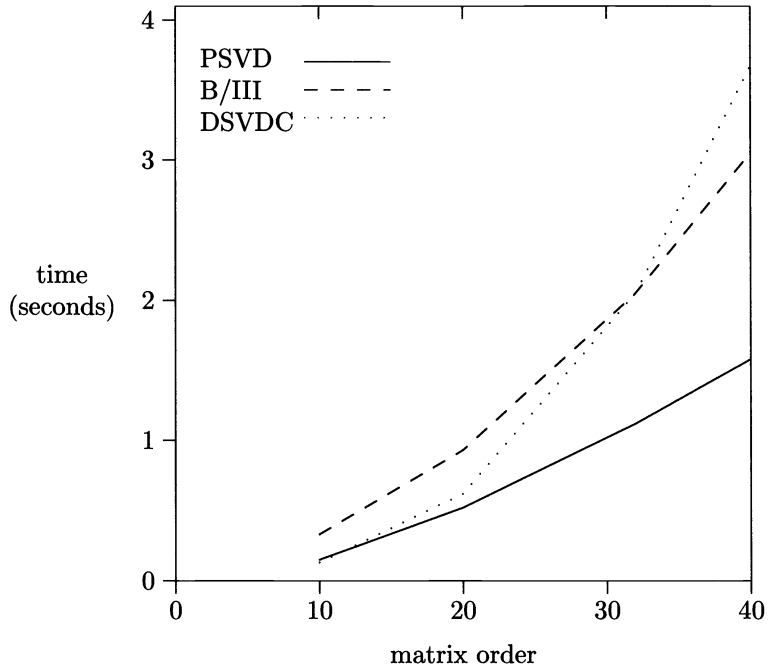


FIG. 3. *Times for computation of the* SVD *by* B/III, PSVD, *and* DSVDC *versus matrix order for matrix* B_W.

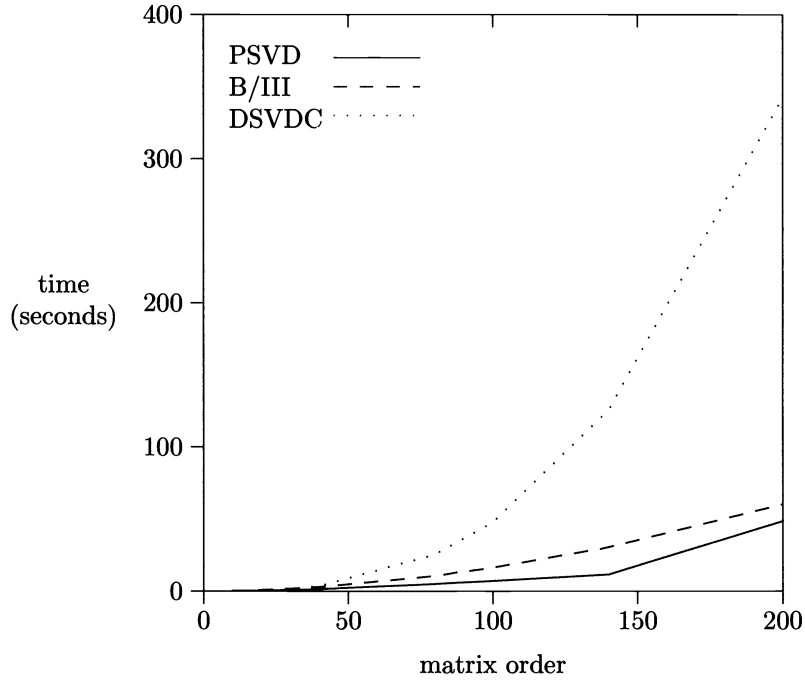


FIG. 4. *Times for computation of the* SVD *by* B/III, PSVD, *and* DSVDC *versus matrix order for matrix* B_W.

TABLE 3
Speedups for PSVD *on the Alliant* FX/8.

Order	Ratio of time DSVDC (1 proc) / PSVD (8 procs)	Ratio of time PSVD (1 proc) / PSVD (8 procs)
100	13.4	5.0
150	14.8	5.3
300	15.7	4.7
350	13.9	4.6

in the form of a binary tree of height h. The smallest subproblems are of order $n/2^h$ and lie at the leaves of the tree (tree level 0). At level 0, the subproblems are solved independently in parallel, one problem per processor. At level l, $1 \leq l \leq h$, each problem of order $n/2^{h-l}$ is solved by updating the solutions to a pair of order $n/2^{h-l+1}$ subproblems from level $l-1$. At these levels, parallelism is achieved by dynamically assigning root-finding and singular vector computation tasks to processors [17]. As in the symmetric case, this parallel algorithm can be pipelined with block reduction of a matrix to bidiagonal form [9, 10, 17]. (An implementation of PSVD for distributed-memory machines without dynamically scheduled processes is described in [15]. Experiments with a similar implementation of the symmetric eigensolver suggest that PSVD would not be efficient on statically scheduled multiprocessors [14], [15].)

Table 3 shows the speedup of PSVD run on eight processors relative to the PSVD and DSVDC run on one processor for matrix [2,1]. As can be seen, the performance of the parallel algorithm compared to DSVDC on a single processor is quite impressive. The somewhat disappointing results in the second column of this table are not yet understood. We attribute it to two aspects of the implementation. First, the deflation step within each SVD update step is done serially and must be completed before any dynamic allocation is done for root-finding. This limits the expected speedup. Second, there is potential for cache conflict when explicit parallel processing is done on the Alliant FX/8. We have not quantified either of these phenomena, however. Speedup of PSVD on eight processors as compared to one processor is limited to about 5. When the same comparison is done with four processors, speedup is limited to about 3. In all cases, the accuracies of DSVDC and PSVD were comparable.

The results in this section suggest that, as for the symmetric tridiagonal eigenproblem, PSVD provides a fast and accurate serial alternative to DSVDC and B/III. More care is needed in the parallel implementation to increase the speedup observed when PSVD is compared to itself using one processor and eight processors. Moreover, a careful study of the effects of deflation is in order. A cursory examination of the results seemed to indicate that deflation is not nearly as prevalent in this setting as it has been in the symmetric tridiagonal case.

Acknowledgments. We are grateful to Peter Arbenz and Gene Golub for several enlightening discussions. In particular, their paper [2] motivated us to consider the formulation developed in §3. We also wish to acknowledge a stimulating discussion during a special session on divide and conquer methods during the Gatlinburg X meeting held at Fairfield Glade in October 1987. A question from Gene Wachspress prompted us to reconsider the computation of right singular vectors so that we ultimately discovered the method described in §6.

REFERENCES

[1] P. ARBENZ, *Divide-and-conquer algorithms for the computation of the SVD of bidiagonal matrices*, in Vector and Parallel Computing, J. Dongarra, I. Duff, P.Gaffney, and S. McKee, eds., Ellis Horwood, Chichester, England, 1989, pp. 1–10.

[2] P. ARBENZ AND G. GOLUB, *On the spectral decomposition of Hermitian matrices subjected to indefinite low rank perturbations*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[3] J. BARLOW, *Error analysis of update methods for the symmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 598–618.

[4] J. BUNCH AND C. NIELSEN, *Updating the singular value decomposition*, Numer. Math., 31 (1978), pp. 111–129.

[5] J. BUNCH, C. NIELSEN, AND D. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[6] J. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

[7] J. DEMMEL AND W. KAHAN, *LAPACK Working Note #3: Computing Small Singular Values of Bidiagonal Matrices with Guaranteed Relative Accuracy*, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1988.

[8] J. DONGARRA, J. BUNCH, C. MOLER, AND G. STEWART, *LINPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[9] J. DONGARRA, S. HAMMARLING, AND D. SORENSEN, *Block reduction of matrices to condensed form for eigenvalue computations*, J. Comput. Appl. Math., 27 (1989), pp. 215–227.

[10] J. DONGARRA AND D. SORENSEN, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. s139–s154.

[11] G. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.

[12] G. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[13] G. GOLUB AND C. REINSCH, *Singular value decomposition and least squares solutions*, in Handbook for Automatic Computation: Linear Algebra, Springer-Verlag, Berlin, New York, 1971, pp. 134–151.

[14] I. IPSEN AND E. JESSUP, *Solving the symmetric tridiagonal eigenvalue problem on the hypercube*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 203–229.

[15] E. JESSUP, *Parallel Solution of the Symmetric Tridiagonal Eigenproblem*, Ph.D. thesis, Dept. of Computer Science, Yale University, New Haven, CT, 1989.

[16] E. JESSUP AND I. IPSEN, *Improving the accuracy of inverse iteration*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 550–571.

[17] E. JESSUP AND D. SORENSEN, *A Parallel Algorithm for Computing the Singular Value Decomposition of a Matrix*, Tech. Report ANL/MCS-TM-102, Argonne National Laboratory, Argonne, IL, 1987.

[18] B. SMITH, J. BOYLE, J. DONGARRA, B. GARBOW, Y. IKEBE, V. KLEMA, AND C. MOLER, *Matrix Eigensystem Routines–EISPACK Guide, Lecture Notes in Computer Science, Vol. 6, 2nd ed.*, Springer-Verlag, New York, 1976.

[19] D. SORENSEN AND J. DONGARRA, *SCHEDULE: Tools for developing and analyzing parallel Fortran programs*, in The Characteristics of Parallel Algorithms, D. G. L. H. Jamieson and R. Douglass, eds., MIT Press, Cambridge, MA, 1987.

[20] D. SORENSEN AND P. TANG, *On the orthogonality of eigenvectors computed by divide and conquer techniques*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.

[21] J. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 549–568, April 1994

013

ACCURATE DOWNDATING OF LEAST SQUARES SOLUTIONS*

Å. BJÖRCK†, H. PARK‡, AND L. ELDÉN†

Abstract. Solutions to a sequence of modified least squares problems, where either a new observation is added (updating) or an old observation is deleted (downdating), are required in many applications. Stable algorithms for downdating can be constructed if the complete QR factorization of the data matrix is available. Algorithms that only downdate R and do not store Q require less operations. However, they do not give good accuracy and may not recover accuracy after an ill-conditioned problem has occurred. The authors describe a new algorithm for accurate downdating of least squares solutions and compare it to existing algorithms. Numerical test results are also presented using the sliding window method, where a number of updatings and downdatings occur repeatedly.

Key words. downdating, iterative refinement, least squares, seminormal equations

AMS subject classifications. 65F05, 65C20, 15A04, 65M12

1. Introduction.

Many problems in signal processing can be formulated as a least squares problem

$$\min_w \|Xw - s\|_2, \quad X \in \mathbf{R}^{p\times n}, \quad p > n. \tag{1.1}$$

If rank $(X) = n$ and the QR decomposition of the data matrix $(X\ s)$ is

$$Q^T(X\ s) = \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{p\times(n+1)}, \tag{1.2}$$

where $Q \in \mathbf{R}^{p\times p}$, then the least squares solution w is obtained from

$$Rw = u, \tag{1.3}$$

and the residual vector r and its norm satisfy

$$r = s - Xw, \qquad \|r\|_2 = |\rho|.$$

(Throughout this paper, we assume that X and s are scaled to have norms $O(1)$.)

Frequently, we know the factorization in (1.2) and wish to find the solution to a modified problem

$$\min_w \|\tilde{X}w - \tilde{s}\|_2,$$

where a new observation $(y^T\ \eta)$ is added (updating):

$$\tilde{X} = \begin{pmatrix} X \\ y^T \end{pmatrix}, \qquad \tilde{s} = \begin{pmatrix} s \\ \eta \end{pmatrix},$$

* Received by the editors April 7, 1992; accepted for publication (in revised form) September 10, 1992.

† Department of Mathematics, Linköping University, S-581 83, Linköping, Sweden (akbjo@math.liu.se, laeld@math.liu.se). This work was supported in part by the Institute for Mathematics and its Applications, University of Minnesota, with funds provided by the National Science Foundation.

‡ Computer Science Department, University of Minnesota, Minneapolis, Minnesota 55455 (hpark@cs.umn.edu). The work of this author was supported in part by the National Science Foundation grant CCR-8813493.

or an old observation $(z^T\ \sigma)$ is removed (downdating):

$$X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix}, \qquad s = \begin{pmatrix} \sigma \\ \tilde{s} \end{pmatrix}.$$

Often the modified problem involves both an updating and a downdating. From (1.3), we see that the solution to the modified problem can be obtained by modifying the R factor of the corresponding augmented matrix $(\tilde{X}\ \tilde{s})$. If R and $\tilde{R}$ are the R factors of X and $\tilde{X}$, respectively, then we have for updating

$$\tilde{R}^T\tilde{R} = R^TR + yy^T,$$

and for downdating

$$\tilde{R}^T\tilde{R} = R^TR - zz^T.$$

Throughout this paper, we will assume that the data matrices X and $\tilde{X}$ have full column rank. Hence, the problem of modifying the R factor of X after a row is added or deleted is mathematically, but not numerically, equivalent to that of updating or downdating a Cholesky factorization under a rank-one perturbation. Thus, when we deal with the solution of the least squares problem (1.1), it is essential to consider the matrix X as the data rather than the upper triangular factor R. Note that an algorithm that is stable merely for the problem of downdating R^TR, may not be stable for downdating least squares solutions.

From the relation $\sigma_i^2(A) = \lambda_i(A^TA)$ and classical perturbation theory for eigenvalues [8], it follows that the singular values $\tilde{\sigma}_i = \sigma_i(\tilde{R})$ interleave with $\sigma_i = \sigma_i(R)$, where for downdating

$$\sigma_1 \geq \tilde{\sigma}_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq \tilde{\sigma}_n \geq 0.$$

In downdating, the smallest singular value may decrease and we can have $\tilde{\sigma}_n \approx 0$, even when R has full column rank. Moreover, any singular value may decrease by a considerable amount, which indicates that downdating can be a sensitive problem [15]. On the other hand, updating R will increase all its singular values.

Important applications where the recursive least squares problems arise include speech echo cancellation, speech coding, and adaptive radar signal processing. The following issues are critical for these applications [1].

1. The modification should be performed with *as few operations* (*real time applications*) *and as little storage requirement as possible.* Recomputing the QR factorization is too costly since it requires $O(pn^2)$ operations, and, thus, a modification technique must be used.

2. The solution should be accurate up to the limitations of data and conditioning of the problem, i.e., a *stable numerical method* must be used. It should be possible to use a computer with *short word length.* This rules out the use of the method of normal equations, which requires twice the word length as methods based on the QR decomposition.

The purpose of this paper is to discuss accurate and efficient algorithms for downdating least squares solutions. We consider the LINPACK algorithm and indicate that it does not give an accurate solution when the downdating problem is ill conditioned. Then we discuss more accurate algorithms: the downdating algorithm based on Gram–Schmidt orthogonalization and an algorithm based on corrected seminormal equations.

The paper is organized as follows. In §2, we review the algorithms for updating and downdating the QR decomposition when both Q and R factors are available. In §3, the downdating algorithm based on the Gram–Schmidt (GS) orthogonalization method is summarized. In §§4 and 5, the LINPACK algorithm for downdating the Cholesky factor and its stability properties are presented. A downdating algorithm based on the corrected seminormal equation method is described in §6. In §7, we describe the application of the methods of this paper to the problem of downdating R^{-1}. Finally, we compare the algorithms discussed in this paper and present the results of numerical tests in §8.

2. Updating and downdating the QR decomposition. Assume that we have computed the QR decomposition of $(X\ s)$ as in (1.2). Then we have

$$\begin{pmatrix} Q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X & s \\ y^T & \eta \end{pmatrix} = \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \\ y^T & \eta \end{pmatrix}. \tag{2.1}$$

The row $(y^T\ \eta)$ can now be annihilated and the updated factor is computed by a sequence of plane rotations $U = G_1 \cdots G_{n+1}$, where G_k is a rotation in the plane $(k, p+1)$. We obtain

$$U^T \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \\ y^T & \eta \end{pmatrix} = \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ 0 & 0 \end{pmatrix}. \tag{2.2}$$

It then follows that

$$\tilde{Q} = \begin{pmatrix} Q & 0 \\ 0 & 1 \end{pmatrix} U \tag{2.3}$$

is the updated factor $\tilde{Q}$. *Note that Q is not needed for the construction of U and of the updated factor $\tilde{R}$.* This algorithm for updating is backward stable [9]. Indeed, if we construct the upper triangular factor by a sequence of such modifications, the resulting algorithm is equivalent to the sequential row orthogonalization method for computing the QR decomposition.

Assume that we have the QR decomposition

$$(X \quad s) = \begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = Q \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} \tag{2.4}$$

and want to *remove* the first row $(z^T\ \sigma)$. We now show that this is equivalent to updating the QR factorization when a special column $e_1 = (1, 0, \ldots, 0)^T$ is *added* to the left of $(X\ s)$,

$$(e_1 \quad X \quad s) = \begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{X} & \tilde{s} \end{pmatrix}.$$

Using (2.4) it follows that

$$Q^T (e_1 \quad X \quad s) = \begin{pmatrix} q_1 & R & u \\ \psi & 0 & \rho \\ q_2 & 0 & 0 \end{pmatrix},$$

where $q^T \equiv (q_1^T \ \psi \ q_2^T)$ is the first row of Q. We can now determine a sequence of plane rotations J_k, $k = p-1, p-2, \ldots, 1$, in the plane $(k, k+1)$ such that

$$U^T \begin{pmatrix} q_1 & R & u \\ \psi & 0 & \rho \\ q_2 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & v^T & \tau \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \\ 0 & 0 & 0 \end{pmatrix}, \qquad U^T = J_1 \cdots J_{p-2} J_{p-1}. \tag{2.5}$$

Here J_k is chosen to annihilate the $(k+1)$st component in q. Then we have

$$\hat{Q}^T \begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} 1 & v^T & \tau \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \\ 0 & 0 & 0 \end{pmatrix},$$

where $\hat{Q} = QU$. Note that by an extra reflection we could ensure that $\tilde{\rho} \geq 0$, but we do not assume this in the following. Equating the first columns on both sides we see that $\hat{Q}^T e_1 = e_1$, so the first row in $\hat{Q}$ equals e_1. Hence, $\hat{Q}$ must have the form

$$\begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q} \end{pmatrix},$$

and it follows that $(v^T \ \tau) = (z^T \ \sigma)$. Dropping the first row and column gives the downdated QR decomposition of

$$(\tilde{X} \ \tilde{s}) = \tilde{Q} \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ 0 & 0 \end{pmatrix}.$$

Note the important fact that in the downdating case, we need the first row of the square orthogonal factor Q to construct the matrix U. Paige [9] has proved that this downdating algorithm is *mixed stable*, i.e., the computed $\tilde{R}, \tilde{u}$, and $\tilde{\rho}$ are close to the corresponding quantities in the exact factor of

$$(\tilde{X} + E, \tilde{s} + f), \quad \|E\|_2 = c_1 \mu, \quad \|f\|_2 = c_2 \mu,$$

where c_1 and c_2 are constants depending on the dimension of X, and μ is the round-off unit.

3. Modifying the GS factorization. In many applications, especially if $p \gg n$, *it is too costly to save and modify the full QR decomposition.* When we use the GS QR factorization, the storage requirement for the Q factor is reduced to pn from p^2, which is for the full QR decomposition. In [5], stable algorithms are derived for modifying the GS QR factorization of a matrix A when A is changed by a matrix of rank one, or when a row or column is added or deleted. A principal tool of the algorithms is the GS process *with reorthogonalization.* A slightly simplified algorithm given in Reichel and Gragg [13] relies on the fact that in the full-rank case, one reorthogonalization is always enough; see Parlett [12].

The algorithm given in §2 for adding a row also applies with trivial modifications to the GS factorization. Assume now that we have the QR factorization

$$(X \quad s) = \begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} q_1^T & \psi \\ Q_1 & y \end{pmatrix} \begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}, \tag{3.1}$$

and want to delete the first row $(z^T \quad \sigma)$. Note that (3.1) can be written as

$$\begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} q_1^T & \psi & 1 \\ Q_1 & y & 0 \end{pmatrix} \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix}. \tag{3.2}$$

Following [5], we first apply the GS process (with reorthogonalization) so that $e_1 = (1, 0 \cdots, 0)^T$ is orthogonalized to

$$\hat{Q}_1 \equiv \begin{pmatrix} q_1^T & \psi \\ Q_1 & y \end{pmatrix} \in \mathbf{R}^{p \times (n+1)}.$$

Because of the special form of the appended column, the result has the form

$$\begin{pmatrix} q_1^T & \psi & 1 \\ Q_1 & y & 0 \end{pmatrix} = \begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} \begin{pmatrix} I & 0 & q_1 \\ 0 & 1 & \psi \\ 0 & 0 & \hat{\gamma} \end{pmatrix} \tag{3.3}$$

for some $h \in \mathbf{R}^{(p-1)\times 1}$, $\hat{\gamma}$, and $\bar{\gamma} \in \mathbf{R}$. Here $q_1^T q_1 + \psi^2 + \hat{\gamma}^2 = \|e_1\|_2^2 = 1$, and equating the first element in the last column in (3.3) $q_1^T q_1 + \psi^2 + \bar{\gamma}\hat{\gamma} = 1$. Hence we have $\hat{\gamma} = \bar{\gamma}$. If e_1 is linearly dependent on the columns of $\hat{Q}_1$, then we get $\bar{\gamma} = 0$, $h = 0$, and the orthogonalization will fail. In this case we can take a random vector in $\mathbf{R}^{(p-1)\times 1}$ and reorthogonalize to find a unit vector h that is orthogonal to $(Q_1 \; y)$; see [5].

We now write using (3.2) and (3.3)

$$\begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix},$$

and determine a sequence of plane rotations J_k, $k = n+1, n, \ldots, 1$, in the plane $(k, n+2)$ such that

$$\begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} U = \begin{pmatrix} 0 & 0 & \tau \\ \tilde{Q}_1 & \tilde{y} & \tilde{h} \end{pmatrix}, \qquad U = J_{n+1} J_n \cdots J_1,$$

where J_k is chosen to annihilate the kth component in $(q_1^T \; \psi \; \bar{\gamma})$. Since orthogonal transformations preserve length we can make $\tau = 1$. The transformed matrix has orthonormal columns and so $\tilde{h} = 0$. It follows that

$$\begin{pmatrix} z^T & \sigma \\ \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ \tilde{Q}_1 & \tilde{y} & 0 \end{pmatrix} \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ z^T & \sigma \end{pmatrix},$$

where

$$U^T \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ z^T & \sigma \end{pmatrix}$$

with $\tilde{R}$ upper triangular, and the downdated QR decomposition becomes

$$(\tilde{X} \quad \tilde{s}) = (\tilde{Q}_1 \; \tilde{y}) \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \end{pmatrix}. \tag{3.4}$$

Summarizing the above, we get the following algorithm.

GS downdating algorithm. Given

$$\hat{Q}_1 = \begin{pmatrix} q_1^T & \psi \\ Q_1 & y \end{pmatrix} \in \mathbf{R}^{p\times(n+1)} \quad \text{and} \quad \begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix} \in \mathbf{R}^{(n+1)\times(n+1)},$$

the following algorithm computes the downdated quantities $\tilde{Q}_1, \tilde{R}, \tilde{u}, \tilde{\rho}$ and $\tilde{w}$.

1. Orthogonalize e_1 to $\hat{Q}_1$ by the GS process with one reorthogonalization step:
   (a) $s := (q_1^T \ \psi)^T$
   (b) $v := e_1 - \hat{Q}_1 s$
   (c) if $\|v\|_2 \geq 1/\sqrt{2}$, $v := v/\|v\|_2$
   (d) else
      i. $s' := \hat{Q}_1^T v$; $v' := v - \hat{Q}_1 s'$
      ii. if $\|v'\|_2 \leq \|v\|_2/\sqrt{2}$, $\bar{\gamma} = 0$;
      determine h with unit length orthogonal to $(Q_1 \ y)$
      iii. else $v := v'/\|v'\|_2$
   (e) $\bar{\gamma} := v(1)$; $h := v(2:p)$.

2. Determine an orthogonal matrix U as a product of Givens rotations such that

$$\begin{pmatrix} 0 & 0 & 1 \\ \tilde{Q}_1 & \tilde{y} & \tilde{h} \end{pmatrix} := \begin{pmatrix} q_1^T & \psi & \bar{\gamma} \\ Q_1 & y & h \end{pmatrix} U. \tag{3.5}$$

3. Update the R factor by U^T:

$$\begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ z^T & \sigma \end{pmatrix} := U^T \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix}. \tag{3.6}$$

4. Compute the new solution $\tilde{w}$ from $\tilde{R}\tilde{w} = \tilde{u}$ and take $\tilde{\rho}$ as the new residual norm.

With one reorthogonalization process, the GS downdating algorithm requires about $7pn + 2.5n^2$ flops. This can be reduced to $5pn + 1.5n^2$ flops when fast scaled rotations [2], [8] are used in (3.5) and (3.6). Note that the data matrix X is never needed: to delete the first row of X, the R factor and the corresponding row in $\hat{Q}_1$ are needed. Thus, the storage requirement is about $pn + 0.5n^2$ for $\hat{Q}_1$ and R.

4. Downdating the Cholesky factor. There are several algorithms for *downdating the Cholesky factor* of A^TA, which is mathematically the same as downdating the R factor of the QR decomposition of A. These algorithms have the property that the Q factor is never used. One important algorithm of this type, which uses hyperbolic rotations, has been analyzed by Alexander, Pan, and Plemmons [1]. Another standard algorithm is the LINPACK algorithm due to Saunders [14].

To derive the LINPACK algorithm for downdating R, note that downdating the ith row of the data matrix $(X \quad s)$ by the method in §2 requires the ith row of the orthogonal factor Q. Also note that the transformations $J_{n+2}, \ldots, J_{p-1}$ in (2.5) do not affect $\left(\begin{smallmatrix} R & u \\ 0 & \rho \end{smallmatrix}\right)$, but the vector q_2, which is replaced by $\bar{\gamma}e_1$, $\bar{\gamma} = \|q_2\|_2$. Thus, mathematically it suffices to know the first $n+1$ components $(q_1^T \ \psi)$ of the ith row

of Q and $\bar{\gamma} \equiv \|q_2\|_2 = \sqrt{1-(\|q_1\|^2+\psi^2)}$ to delete the ith row of $(X \quad s)$. From the first row of the QR decomposition (2.4), we have

$$(z^T \quad \sigma) = (q_1^T \quad \psi \quad q_2^T)\begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} = (q_1^T \quad \psi)\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}.$$

It follows that q_1 and ψ can be computed by solving the triangular system

$$\begin{pmatrix} R^T & 0 \\ u^T & \rho \end{pmatrix}\begin{pmatrix} q_1 \\ \psi \end{pmatrix} = \begin{pmatrix} z \\ \sigma \end{pmatrix}.$$

Using the relation $u^T q_1 = u^T R^{-T} z = w^T z$, we obtain

$$q_1 = R^{-T} z, \quad \psi = (\sigma - z^T w)/\rho \qquad (\rho \neq 0). \tag{4.1}$$

Next we should determine a product of plane rotations U such that

$$U^T \begin{pmatrix} q_1 & R & u \\ \psi & 0 & \rho \\ \bar{\gamma} & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \end{pmatrix}. \tag{4.2}$$

The first rotation in the $(n+1, n+2)$ plane only affects the 2×2 matrix

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} \psi & \rho \\ \bar{\gamma} & 0 \end{pmatrix} = \begin{pmatrix} \gamma & \hat{\rho} \\ 0 & \tilde{\rho} \end{pmatrix},$$

and a short calculation gives

$$\gamma^2 = \bar{\gamma}^2 + \psi^2 = 1 - \|q_1\|_2^2, \quad \hat{\rho} = (\sigma - z^T w)/\gamma, \quad \tilde{\rho} = (\rho^2 - \hat{\rho}^2)^{1/2}. \tag{4.3}$$

(Note that ψ in (4.1) need not be computed so the assumption that $\rho \neq 0$ is not needed.)

Collecting these results we get the following algorithm

LINPACK downdating algorithm. Given R, u, ρ, w, and $(z^T \sigma)$, the following algorithm computes the downdated quantities $\tilde{R}, \tilde{u}, \tilde{\rho}$, and $\tilde{w}$.

1. Compute q_1, γ, and $\hat{\rho}$ from

$$R^T q_1 = z, \quad \gamma := (1 - \|q_1\|_2^2)^{1/2}, \quad \hat{\rho} := (\sigma - z^T w)/\gamma. \tag{4.4}$$

2. Determine an orthogonal matrix $\hat{U}$ as a product of Givens rotations such that

$$\begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{R} & \tilde{u} \end{pmatrix} := \hat{U}^T \begin{pmatrix} q_1 & R & u \\ \gamma & 0 & \hat{\rho} \end{pmatrix}. \tag{4.5}$$

3. Compute the new solution $\tilde{w}$ and the residual norm from

$$\tilde{R}\tilde{w} = \tilde{u}, \qquad \tilde{\rho} := (\rho^2 - \hat{\rho}^2)^{1/2}.$$

This algorithm requires about $3n^2$ flops for two triangular solves and updating in (4.5), which can be reduced to $2n^2$ flops when fast rotations are used in (4.5). Unlike the GS downdating algorithm, the LINPACK algorithm does not require the Q factor. However, the row $(z^T\ \sigma)$ to be deleted should be known to recover the necessary elements in the Q factor. Thus, for solving recursive least squares problems with the sliding window method, all the rows in the data matrix X need to be stored for future downdating. Accordingly, the storage requirement is $pn + 0.5n^2$. This means that *the storage requirement for the LINPACK downdating algorithm can be about as large as that for the* GS*-based downdating algorithm for recursive least squares problems* contrary to a widespread misconception. In special cases such as when X is sparse or its elements can be generated by a formula, the cost of storing X can be much smaller than that of storing Q.

5. Stability properties of the LINPACK algorithm. It is well known that downdating the Cholesky factor can be very ill conditioned and can fail. We first note that

$$R^TR - zz^T = R^T(I - q_1q_1^T)R = \tilde{R}^T\tilde{R}. \tag{5.1}$$

If we put $I - q_1q_1^T = LL^T$, then $\tilde{R} = L^TR$, where

$$\kappa(L) = \frac{1}{\gamma}, \qquad \gamma = \sqrt{1 - \|q_1\|^2}. \tag{5.2}$$

Stewart [15] considered the effect of a perturbation δz in z on the downdated factor $\tilde{R}$. He showed that if

$$\|\delta z\|_2 \le \mu\sigma_1, \qquad \sigma_1 = \|R\|_2,$$

where μ is the round-off unit, then, neglecting higher-order terms

$$|\delta\tilde{\sigma}_i| \le 2\mu\sigma_1(\sigma_1/\tilde{\sigma}_i), \tag{5.3}$$

where $\tilde{\sigma}_i = \sigma_i(\tilde{R})$. This shows that the method can break down if $\tilde{\sigma}_i/\sigma_1 \approx \mu^{1/2}$, i.e., if we downdate to an ill-conditioned matrix $\tilde{R}$. The analysis in [15] also shows that the downdating problem is ill conditioned if any singular value is reduced significantly (not necessarily becoming small). This happens, for example, if the row to be downdated contains an *outlier*, i.e., an erroneous and large element.

Pan [10] has given a detailed perturbation analysis of the downdating problem for the Cholesky factor and has proved the following result.

THEOREM 5.1. *Let $\alpha > 0$ be small enough so that the factorization*

$$\tilde{R}(\epsilon)^T\tilde{R}(\epsilon) = (R + \epsilon E)^T(R + \epsilon E) - (z + \epsilon f)^T(z + \epsilon f) \tag{5.4}$$

exists for all $\epsilon \in (-\alpha, \alpha)$, where E is an upper triangular matrix. Then we have the bound

$$\begin{aligned}\frac{\|\tilde{R}(\epsilon) - \tilde{R}\|_2}{\|\tilde{R}\|_2} \le |\epsilon|\kappa^2(R)C\Bigg[&2n(n^{1/2}C + 1)\frac{\|f\|_2}{\|z\|_2} \\ &+ (2n^{3/2}C + 2n + 1)\frac{\|E\|_2}{\|R\|_2}\Bigg] + O(\epsilon^2),\end{aligned} \tag{5.5}$$

where $\kappa(R)$ is the condition number of R, and $C = \|q_1\|^2/\gamma$.

The above perturbation analysis shows that *using R to form the downdating transformations may be a much more ill-conditioned problem than downdating the original matrix X.* This is because the original row in X is not perturbed in the same way as the vector $(q_1^T \gamma)$, which is computed by solving a triangular system to determine the downdating transformation in the LINPACK algorithm. Hence, any method that uses R alone to recover the necessary elements of Q cannot be *backward stable in the same sense as the downdating algorithm that uses Q directly.*

We now illustrate the perturbation result, and the possible failure of the LINPACK algorithm using a simple example.

Example 1. Let $X = \binom{\tau}{1}$, where $\tau = 1/\sqrt{\mu}$, and let $s = \binom{1}{1}$. We may think of the first row of X as an outlier. Then the QR decomposition of X, correctly rounded to single precision, is

$$X = \begin{pmatrix} \tau \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & -\epsilon \\ \epsilon & 1 \end{pmatrix} \begin{pmatrix} \tau \\ 0 \end{pmatrix},$$

where $\epsilon = 1/\tau$. The LINPACK algorithm will compute

$$q_1 = \tau/\tau = 1, \quad \gamma^2 = 1 - 1 = 0, \quad J_1 = I,$$

and we obtain the downdated factor $\tilde{R} = 0$, and the least squares solution is not defined. It is easily verified that if we downdate using Q, we get the correct result $\tilde{R} = 1$ and $\tilde{w} = 1$.

The information from the second row in X is not present in R, only in Q. Therefore *no method working only from R can hope to do better.*

6. Downdating using seminormal equations. We now consider a downdating algorithm in which the method of iterative refinement is incorporated. The method is based on the *seminormal equations* (SNE)

$$R^T R w = X^T s,$$

for solving a least squares problem $\min_w \|Xw - s\|_2$. This method is generally no more accurate than the method of normal equations. We instead consider the method of *corrected seminormal equations* (CSNE)

$$\begin{aligned} R^T R w &= X^T s, & r &= s - Xw, \\ R^T R \delta w &= X^T r, & w_c &= w + \delta w. \end{aligned} \tag{6.1}$$

Here a corrected solution w_c is computed by performing one step of iterative refinement on the solution computed from the SNE. Note that we assume that *all computations* are performed in *single precision.*

Assume that the computed $R = \bar{R}$ is such that there exists an exactly orthogonal matrix $\hat{Q}$ such that

$$X + E = \hat{Q}\bar{R}, \qquad \|E\|_F \le c_1 \mu \|X\|_F.$$

Then in Björck [4] the following error bound is given for the solution $\bar{w}_c$ computed by CSNE.

THEOREM 6.1. *Let $\bar{w}_c$ be the least squares solution computed by the* CSNE *method, and assume that $\rho = c_1 n^{1/2} \mu \kappa < 1$, where κ is the spectral condition number*

of X. Then the following error estimate holds up to terms of higher order in $\mu\kappa$:

$$\begin{aligned}\|w-\bar{w}_c\|_2 \le \sigma\mu\kappa\Big(c_2\|w\|_2 + n^{1/2}p\frac{\|s\|_2}{\|X\|_2}\Big)\\ + n^{1/2}\mu\kappa\Big(n\|w\|_2 + p\kappa\frac{\|r\|_2}{\|X\|_2}\Big) + n^{1/2}\mu\|w\|_2,\end{aligned} \tag{6.2}$$

where

$$\sigma = c_3\mu\kappa^2, \quad c_2 = 2n^{1/2}(c_1+n), \quad c_3 \le 2n^{1/2}(c_1+2n+p/2). \tag{6.3}$$

Hence, provided that $c_3\mu\kappa^2 < 1$, the forward error will not be worse than for a backward stable method.

In [4], it was shown how the CSNE method can be used to update the R factor when a new column is added. This can be adopted to reconstruct the vector $(q_1^T\ \gamma)$ as follows.

THEOREM 6.2. *Let v be the solution to*

$$\min_v \|e_1 - Xv\|_2,$$

and R the R factor of X. Then the R factor of $(X\ e_1)$ is

$$\begin{pmatrix} R & q_1 \\ 0 & \gamma \end{pmatrix}, \quad q_1 = Rv, \quad |\gamma| = \|e_1 - Xv\|_2. \tag{6.4}$$

The downdated R factor is then obtained by applying orthogonal transformations to transform the last column into the vector e_1. We now apply this result to downdate the *augmented* R factor by solving the least squares problem

$$\min_{x,\phi} \left\| e_1 - (X\ s)\begin{pmatrix} x \\ \phi \end{pmatrix} \right\|_2$$

using the CSNE.

The first step is similar to the LINPACK algorithm. Assuming $\rho \neq 0$, from

$$\begin{pmatrix} R^T & 0 \\ u^T & \rho \end{pmatrix}\begin{pmatrix} q_1 \\ \psi \end{pmatrix} = \begin{pmatrix} z \\ \sigma \end{pmatrix},$$

we get

$$q_1 = R^{-T}z, \qquad \psi = (\sigma - z^T w)/\rho.$$

Next we solve

$$\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}\begin{pmatrix} x \\ \phi \end{pmatrix} = \begin{pmatrix} q_1 \\ \psi \end{pmatrix},$$

which gives

$$\phi = \psi/\rho, \qquad x = R^{-1}(q_1 - u\phi) = v - \phi w.$$

We have $|\gamma| = \|e_1 - Xv\|_2$, and ϕ and x are only needed to compute $|\bar{\gamma}| = \|t\|_2$, where

$$t = e_1 - (X\ s)\begin{pmatrix} x \\ \phi \end{pmatrix} = e_1 - Xv - \psi\hat{r}, \qquad \hat{r} = (s - Xw)/\rho. \tag{6.5}$$

In exact computing it holds that

$$\rho^2 = \|Xw - s\|_2^2 = \|\tilde{X}w - \tilde{s}\|_2^2 + (\sigma - z^T w)^2,$$

and $\hat{r}$ is the residual normalized to unit length with first component equal to ψ. Hence $\rho = 0$ implies that $\hat{\rho} = 0$, and $\tilde{\rho} = \|\tilde{X}\tilde{w} - \tilde{s}\|_2 = 0$. In floating point computation the division $\bar{r}/\rho$, where $\bar{r} = fl(s - Xw)$ is the *computed* residual, may not be safe to perform if $\rho \neq 0$ but very small. If we recompute $\rho = fl(\|\bar{r}\|_2)$, then if $fl(1 + \rho) = 1$ the true residual norm certainly is of the order of machine precision. In this case we put $\psi\hat{r} = 0$ in (6.5) and take $\hat{\rho} = \tilde{\rho} = 0$. Otherwise, ρ will equal $\|\bar{r}\|_2$ with small relative error, and the division in computing $\hat{r}$ can be carried out safely. Following (4.3), we then compute

$$\hat{\rho} = \rho\psi/\gamma, \qquad \tilde{\rho} = \rho\bar{\gamma}/\gamma,$$

and continue as in the LINPACK downdating algorithm.

We remark that because the condition number of the augmented R factor is large when ρ is small, it is important to refine q_1 before ψ is computed, i.e., to perform the algorithm in Gauss–Seidel rather than Jacobi fashion!

CSNE downdating algorithm. Given R, u, w, the data $(X \ s)$ the following algorithm deletes the first row $(z^T \ \sigma)$ and computes the downdated quantities $\tilde{R}, \tilde{u}, \tilde{\rho}$, and $\tilde{w}$.

1. Compute q_1, v and t from

$$R^T q_1 = z, \quad Rv = q_1, \quad t := e_1 - Xv.$$

2. Update q_1, v and compute γ:

$$\begin{aligned} R^T \delta q_1 &= X^T t, \qquad q_1 := q_1 + \delta q_1, \\ R\delta v &= \delta q_1, \quad t := t - X\delta v, \quad \gamma := \|t\|_2. \end{aligned}$$

3. Set $\tilde{\rho} := \hat{\rho} := 0$ and compute the residual:

$$r := s - Xw, \qquad \rho = \|r\|_2.$$

If $fl(1 + \rho) \neq 1$,
   (a) normalize the residual: $\hat{r} = r/\rho$,
   (b) modify t: $\psi := e_1^T \hat{r}, \quad t := t - \psi\hat{r}$,
   (c) update ψ and t: $\delta\psi := \hat{r}^T t, \quad \psi := \psi + \delta\psi, \quad t := t - \delta\psi\hat{r}$,
   (d) compute $\bar{\gamma} = \|t\|_2, \quad \hat{\rho} = \psi\rho/\gamma, \quad \tilde{\rho} = \rho\bar{\gamma}/\gamma$.

4. Determine an orthogonal matrix U^T as a product of Givens rotations such that

$$\begin{pmatrix} 1 & z^T & \sigma \\ 0 & \tilde{R} & \tilde{u} \end{pmatrix} := U^T \begin{pmatrix} q_1 & R & u \\ \gamma & 0 & \hat{\rho} \end{pmatrix}.$$

5. Compute the new solution $\tilde{w}$ from

$$\tilde{R}\tilde{w} = \tilde{u}.$$

Example 2. Let X be as in Example 1. In the method of seminormal equations, we compute

$$q_1 = 1, \quad v = \frac{1}{\tau}, \quad \gamma = \left\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} \tau \\ 1 \end{pmatrix} \tau^{-1} \right\|_2 = \epsilon = \frac{1}{\tau}.$$

There is no need for the refinement steps 2 and 3 here, and we get

$$U^T \begin{pmatrix} 1 & \tau \\ \epsilon & 0 \end{pmatrix} = \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}, \qquad \tilde{R} = 1,$$

which is the correct result.

Algorithm CSNE requires three more triangular solves than the LINPACK method, if the iterative refinement is carried out. Also, four extra matrix times vector multiplications with X and X^T are required. Hence, the added computational complexity is quite high, $\approx 4pn + 1.5n^2$ flops. However, sometimes X may have a special structure and the matrix-vector multiplications can be performed by fast algorithms. This is the case, for example, when X is a sparse or Toeplitz matrix. The storage requirement is the same as that for the LINPACK algorithm.

When the CSNE algorithm is too expensive to use in every step, we suggest a *hybrid algorithm* [6], where the CSNE algorithm is used if the downdating is ill conditioned and the LINPACK algorithm is used otherwise. As a measure of conditioning of the downdating problem for R augmented by the right-hand side u and residual ρ, we have used the quantity (cf. [15])

$$\gamma^2 = 1 - \|q_1\|_2^2. \tag{6.6}$$

If γ^2 is less than a user-specified constant tol, then the downdating is performed with CSNE. Our numerical experiments indicate the hybrid method with tol in the range $[0.25, 0.5]$ produces much more accurate results than the LINPACK algorithm.

7. Updating and downdating the inverse of R. The problem of updating and downdating the inverse of the matrix R in the QR decomposition has been studied in [11] for methods based on orthogonal as well as hyperbolic rotations. One motivation for working with R^{-1} instead of R itself is that such an algorithm can be parallelized more easily. Furthermore, there are applications (see [11]) where the elements of the inverse are needed.

The downdating methods described in this paper can be modified to downdate the inverse of R. We first describe how the inverse and the solution vector w can be downdated recursively.

Consider the transformation

$$U^T \begin{pmatrix} q_1 & R \\ \gamma & 0 \end{pmatrix} = \begin{pmatrix} 1 & z^T \\ 0 & \tilde{R} \end{pmatrix}, \qquad U^T \begin{pmatrix} u \\ \hat{\rho} \end{pmatrix} = \begin{pmatrix} \sigma \\ \tilde{u} \end{pmatrix}. \tag{7.1}$$

By simply inverting the matrices in the first equation in (7.1) we get the following formula for downdating the inverse $S = R^{-1}$:

$$\begin{pmatrix} 0 & \frac{1}{\gamma} \\ S & -\frac{1}{\gamma} v \end{pmatrix} U = \begin{pmatrix} 1 & -z^T \tilde{S} \\ 0 & \tilde{S} \end{pmatrix}, \tag{7.2}$$

where $v = R^{-1} q_1 = S q_1$. Hence the same orthogonal transformation U downdates the inverse. U can be determined as a product of plane rotations that zeros the diagonal in S from bottom to top

$$U = J_{n,n+1} \cdots J_{23} J_{12}.$$

This determines $\tilde{S} = \tilde{R}^{-1}$. Using the second equation in (7.1), we can determine $\tilde{u}$

$$(u^T \quad \hat{\rho})\, U = (\sigma \quad \tilde{u}^T), \tag{7.3}$$

and the downdated solution is obtained from $\tilde{w} = \tilde{S}\tilde{u}$.

A different downdating formula for $\tilde{w}$ is obtained as follows. Combining (7.2) and the second equation in (7.1), we see that

$$\begin{pmatrix} \sigma - z^T\tilde{S}\tilde{u} \\ \tilde{S}\tilde{u} \end{pmatrix} = \begin{pmatrix} 1 & -z^T\tilde{S} \\ 0 & \tilde{S} \end{pmatrix} \begin{pmatrix} \sigma \\ \tilde{u} \end{pmatrix} = \begin{pmatrix} 1 & -z^T\tilde{S} \\ 0 & \tilde{S} \end{pmatrix} U^T \begin{pmatrix} u \\ \hat{\rho} \end{pmatrix}$$
$$= \begin{pmatrix} 0 & \frac{1}{\gamma} \\ S & -\frac{1}{\gamma}v \end{pmatrix} \begin{pmatrix} u \\ \hat{\rho} \end{pmatrix} = \begin{pmatrix} \frac{\hat{\rho}}{\gamma} \\ Su - \frac{\hat{\rho}}{\gamma}v \end{pmatrix},$$

and hence

$$\tilde{w} = w - \frac{\hat{\rho}}{\gamma}v. \tag{7.4}$$

Since we assume that $\operatorname{rank}(\tilde{X}) = n$, we have $\gamma \neq 0$. The LINPACK, the CSNE, and the hybrid algorithms only differ in the way the vector $(q_1^T \ \gamma)$ is computed. For determining this vector, triangular systems with the matrices R and R^T need to be solved. This can be replaced by the corresponding multiplication by the inverse. Therefore these methods can also be used in connection with inverse downdating.

For updating the inverse factor, we consider the relation

$$U^T \begin{pmatrix} y^T & \eta \\ R & u \\ 0 & \rho \end{pmatrix} = \begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ 0 & 0 \end{pmatrix}. \tag{7.5}$$

By adding a column e_1, we write the first part of this equation as

$$U^T \begin{pmatrix} 1 & y^T \\ 0 & R \end{pmatrix} = \begin{pmatrix} p_1 & \tilde{R} \\ \xi & 0 \end{pmatrix},$$

where $(p_1 \quad \xi)$ is the first row of U. We then invert this equation to get

$$\begin{pmatrix} 1 & -y^T S \\ 0 & S \end{pmatrix} U = \begin{pmatrix} 0 & \frac{1}{\xi} \\ \tilde{S} & -\frac{1}{\xi}\tilde{S}p_1 \end{pmatrix}. \tag{7.6}$$

Here we first choose $U = \hat{U}$ as a product of plane rotations

$$\hat{U} = J_{12}J_{23}\cdots J_{n,n+1},$$

where $J_{i,i+1}$, $i = 1, \ldots, n$ are chosen to zero elements in the first row from left to right. The same transformation $\hat{U}$ is then applied to

$$(\eta \quad u^T \quad \rho)\,\hat{U} = (\tilde{u}^T \quad \bar{\rho} \quad \rho). \tag{7.7}$$

(Compare the last column of (7.5), and note that $\hat{U}$ does not affect column $n+2$.) Finally we choose $J_{n+1,n+2}$ to zero the element ρ. This will not affect $\tilde{u}$ and $\tilde{S}$ and determines

$$\tilde{\rho} = (\bar{\rho}^2 + \rho^2)^{1/2}. \tag{7.8}$$

Inverse updating algorithm. Given $S = R^{-1}, u, \rho, w$, the data $(X \ s)$, the following algorithm adds a row $(y^T \ \eta)$ and computes the updated quantities $\tilde{S}, \tilde{u}, \tilde{\rho}$, and $\tilde{w}$.

1. Determine $\tilde{u}$ and $\tilde{S}$

$$\begin{pmatrix} 0 & \frac{1}{\xi} \\ \tilde{u}^T & \bar{\rho} \\ \tilde{S} & -\frac{1}{\xi}\tilde{S}p_1 \end{pmatrix} := \begin{pmatrix} 1 & -y^T S \\ \eta & u^T \\ 0 & S \end{pmatrix} \hat{U}, \tag{7.9}$$

where $\hat{U}$ is a product of Givens rotations

$$\hat{U} = J_{12}J_{23}\cdots J_{n,n+1}.$$

2. Compute $\tilde{\rho}$ and the updated solution $\tilde{w}$ from

$$\tilde{\rho} = (\bar{\rho}^2 + \rho^2)^{1/2}, \qquad \tilde{w} = \tilde{S}\tilde{u}.$$

LINPACK inverse downdating algorithm. Given $S = R^{-1}, u, \rho, w$, the data $(X\ s)$, the following algorithm deletes the first row $(z^T\ \sigma)$ and computes the downdated quantities $\tilde{S}$, $\tilde{u}$, $\tilde{\rho}$, and $\tilde{w}$:

1. Compute q_1, v, and γ from

$$q_1 = S^T z, \quad v = Sq_1, \quad \gamma = \left(1 - \|q_1\|_2^2\right)^{1/2}.$$

2. Compute: $\hat{\rho} = (\sigma - z^T w)/\gamma, \quad \tilde{\rho} = (\rho^2 - \hat{\rho}^2)^{1/2}.$
3. Determine $\tilde{u}$ and $\tilde{S}$ from

$$\begin{pmatrix} \sigma & \tilde{u}^T \\ 1 & -z^T\tilde{S} \\ 0 & \tilde{S} \end{pmatrix} := \begin{pmatrix} u^T & \hat{\rho} \\ 0 & \frac{1}{\gamma} \\ S & -\frac{1}{\gamma}v \end{pmatrix} U,$$

where U is a product of Givens rotations

$$U = J_{n,n+1}\cdots J_{23}J_{12}.$$

4. Compute the new solution $\tilde{w}$ from

$$\tilde{w} = \tilde{S}\tilde{u} = w - \frac{1}{\gamma}\hat{\rho}v.$$

CSNE inverse downdating algorithm. Given $S = R^{-1}, u, \rho, w$, the data $(X\ s)$, the following algorithm deletes the first row $(z^T\ \sigma)$ and computes the downdated quantities $\tilde{S}, \tilde{u}, \tilde{\rho}$, and $\tilde{w}$:

1. Compute q_1, v, and t from

$$q_1 = S^T z, \quad v = Sq_1, \quad t := e_1 - Xv.$$

2. Update v and compute γ:

$$\begin{aligned} \delta q_1 &= S^T X^T t, \quad q_1 := q_1 + \delta q_1, \quad \delta v = S\delta q_1, \\ v &:= v + \delta v, \quad t := t - X\delta v, \quad \gamma := \|t\|_2. \end{aligned}$$

3. Set $\psi := \tilde{\rho} := \hat{\rho} := 0$ and compute the residual:

$$r := s - Xw, \qquad \rho = \|r\|_2.$$

If $fl(1+\rho) \neq 1$,

(a) normalize the residual: $\hat{r} = r/\rho$,
(b) modify t: $\psi := e_1^T \hat{r}, \quad t := t - \psi \hat{r}$,
(c) update ψ and t: $\delta\psi := \hat{r}^T t, \quad \psi := \psi + \delta\psi, \quad t := t - \delta\psi\hat{r}$,
(d) compute: $\tilde{\rho} = \|t\|_2 \rho/\gamma, \quad \hat{\rho} = \psi\rho/\gamma$.

4. Determine $\tilde{u}$ and $\tilde{S}$ from

$$\begin{pmatrix} \sigma & \tilde{u}^T \\ 1 & -z^T\tilde{S} \\ 0 & \tilde{S} \end{pmatrix} := \begin{pmatrix} u^T & \hat{\rho} \\ 0 & \frac{1}{\gamma} \\ S & -\frac{1}{\gamma}v \end{pmatrix} U,$$

where U is a product of Givens rotations

$$U = J_{n,n+1} \cdots J_{23} J_{12}.$$

5. Compute the new solution $\tilde{w}$ from

$$\tilde{w} = \tilde{S}\tilde{u} = w - \frac{1}{\gamma}\hat{\rho}v.$$

8. Numerical experiments. In a sliding window method, a least squares solution is computed based on the p latest rows of an observation matrix A, where p is the number of rows in the window matrix [1]. In step k, the new row of observation, $A(k,:)$, is updated into the QR decomposition, and the existing row $A(k-p,:)$ of the data matrix is downdated from the decomposition. If an outlier occurs at step j, then in exact arithmetic its influence will not be seen after step $j+p$. However, the downdating problem is very ill conditioned in the step when the outlier is to be removed, and any algorithm that does not explicitly use Q or the original data X, e.g., the LINPACK algorithm or a hyperbolic rotation-based algorithm [1], is likely to introduce a large error into the decomposition.

In the sliding window context, the storage requirements of the four algorithms presented in the previous sections are the same in the general case, $pn + 0.5n^2$. The GS algorithm requires the orthogonal factor but not the data matrix in storage, while the other algorithms require the data matrix but not the orthogonal factor. However, when X has a special structure, the storage requirement for X can be much smaller than that for Q.

The computational complexities of the four algorithms for each downdating are compared in Table 8.1. We give the operation counts (1 flop = 1 addition and 1 multiplication) for standard and fast [2], [8] Givens rotations. For the hybrid algorithm, the computational complexity is either the same as that for the LINPACK algorithm or the CSNE algorithm. The LINPACK algorithm has a clear advantage in computational complexity. In a sliding window context, the GS algorithm is more expensive also in the updating stage, since the Q factor needs to be modified. For the CSNE algorithm the updating stage is the same as for the LINPACK method. Hence for a complete up/downdating step using standard Givens rotations the GS algorithm requires $11pn + 4.5n^2$, but the CSNE algorithm requires only $4pn + 6.5n^2$ flops. Finally, note that if, for example, X is Toeplitz the term $4pn$ in the operation count for the CSNE algorithm can be reduced by using a fast algorithm for the matrix-vector products with X and X^T.

One application area where downdating is used in connection with the sliding window method is adaptive filtering. In [3] it is noted that in certain situations this method does not perform well as rounding errors accumulate and eventually destroy

TABLE 8.1
Computational complexity of downdating algorithms (flops).

Algorithm	Standard Givens rotations	Fast Givens rotations
GS	$7pn + 2.5n^2$	$5pn + 1.5n^2$
LINPACK	$3n^2$	$2n^2$
CSNE	$4pn + 4.5n^2$	$4pn + 3.5n^2$
Hybrid	$3n^2$ or $4pn + 4.5n^2$	$2n^2$ or $4pn + 3.5n^2$

the solution. Our numerical tests indicate that these difficulties are not to be ascribed to the sliding window method itself, but rather to the downdating algorithm used.

Numerical tests using the sliding window method have been performed in Pro-Matlab with IEEE double precision floating point arithmetic to compare the accuracy of the four downdating algorithms. The solution obtained from the QR decomposition of the window matrix was used as a reference and a window of size 8 was used throughout. In each figure, we present the relative error in Euclidean norm in the downdated solution vector by the LINPACK, CSNE, hybrid, and GS algorithms. The spectral condition number κ_X of the window matrix to be downdated and $1/\gamma^2$, which is a measure of the conditioning of the downdating problem (6.6), are also shown. The values of $1/\gamma^2$ are from the hybrid method. A plus (+) sign in the plot shows where iterative refinement is made in the hybrid method. We have used the following criterion: if $\gamma^2 < 0.25$, then downdating is performed with the CSNE method.

The following test problems are similar to those in [6]. They were also used in the context of adaptive condition number estimation in [7]. Tests I and II illustrate the downdating of R and Test III illustrates the downdating of R^{-1}.

Test I. A random matrix $A \in \mathbf{R}^{50\times 5}$ was constructed with elements taken from a uniform distribution in $(0, 1)$. An outlier equal to $r \cdot 10^3$, where r is random number from the same distribution, was added in position (18,3). The right-hand side vector b was taken to be $b = Ax_0 + b_r$, where b_r has random elements uniformly distributed in $(0, 10^{-6})$, and x_0 is 5×1 vector with ones as its components.

The results are shown in Fig. 8.1. It is seen that the relative error in the solution using the LINPACK algorithm is considerably magnified in the ill-conditioned downdating step and that it remains on that high level even if the subsequent downdating steps are well conditioned. The other algorithms are much less affected by the ill-conditioned downdating and the errors remain on a low level throughout.

Test II. A 50×5 matrix was constructed by taking a 25×5 Hilbert matrix as the first 25 rows, and the same rows in reversed order as the 25 last rows. Then a perturbation from a uniform distribution in $(0, \delta)$ was added to each matrix element. Two different cases were studied, with $\delta = 10^{-5}$ and 10^{-9}, respectively. The right-hand side was constructed as in Test I, but here with a random perturbation in $(0, 10^{-6})$.

In Fig. 8.2 we show the results obtained with $\delta = 10^{-5}$. Throughout this test the downdating problem is rather ill conditioned, so iterative refinement is performed in most steps in the hybrid algorithm. It is remarkable that the LINPACK algorithm performs so much worse than the others. This is probably due to the fact that the window matrix is very ill conditioned, which leads to large errors in the computed approximations of q_1. In the CSNE method this vector is refined and much better accuracy is attained.

In Fig. 8.3 we show the results obtained with $\delta = 10^{-9}$. Here the window matrix

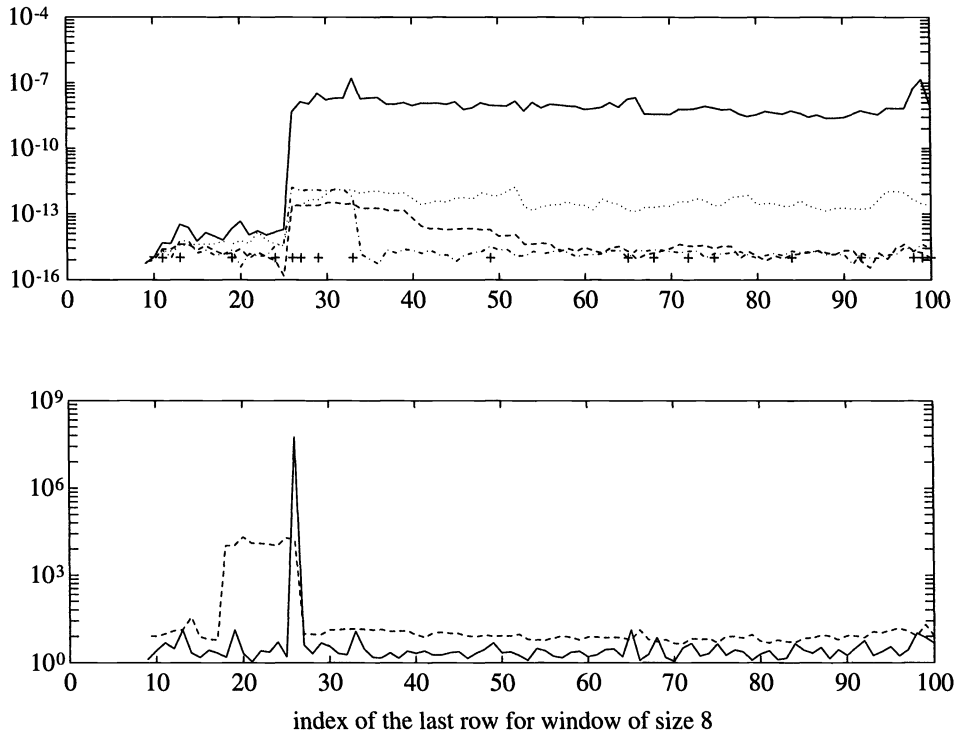


FIG. 8.1. *Test* I. *The upper graph shows the relative error in Euclidean norm in the downdated solution vector by the* LINPACK (*solid line*), CSNE (*dashed*), *hybrid* (*dotted*), *and* GS (*dot-dashed*) *algorithms. A plus* (+) *sign in the plot shows where iterative refinement is made in the hybrid method. The lower graph shows the condition number* κ_X *of the window matrix to be downdated* (*dotted*) *and* $1/\gamma^2$ (*solid line*).

is even more ill conditioned, and after some steps the LINPACK algorithm breaks down because a computed q_1 has norm larger than 1.

Test III. Two problems were solved. The first is the same as in Test I, and the second is the same as in Test II with $\delta = 10^{-5}$. The results are for LINPACK, CSNE, and hybrid *inverse* downdating and are shown in Fig. 8.4.

9. Concluding remarks. We have studied two standard methods for downdating least squares solutions: the LINPACK and the GS algorithms; and two new methods, the CSNE algorithm and a hybrid algorithm CSNE/LINPACK. In terms of storage requirements the four algorithms are all the same in the general case. However, when X has a special structure, the storage requirement for the LINPACK and CSNE algorithms can be much smaller than that for the GS algorithm.

The algorithms differ considerably in efficiency and accuracy. The LINPACK algorithm is the fastest, but the analysis and the tests show that it can be much less accurate or even fail. It is clear that the CSNE algorithm is more accurate but considerably slower than the LINPACK algorithm. However, if X has some special structure, the difference in efficiency may be less pronounced. The hybrid algorithm has almost as good accuracy as the CSNE algorithm and it can be much more efficient. The GS algorithm is comparable in accuracy to the CSNE algorithm, but it is also

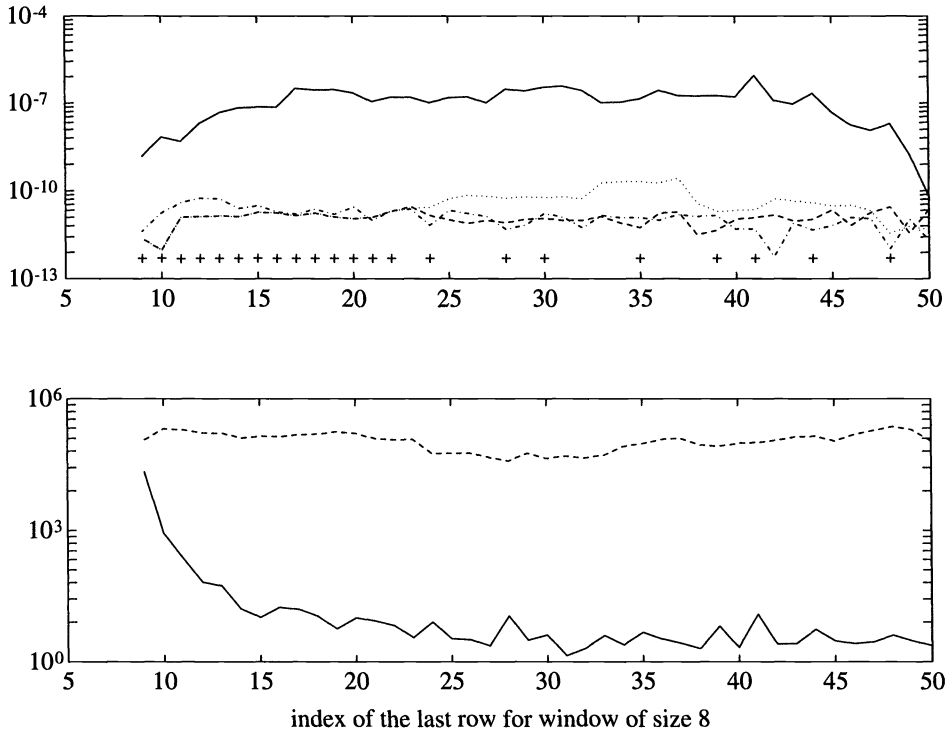


FIG. 8.2. *Test* IIa. *Modified Hilbert matrix with perturbations from a uniform distribution in* $(0, 10^{-5})$.

the slowest of the four algorithms.

The reason why the LINPACK algorithm is inferior in terms of accuracy to the three others is that it uses less information, i.e., only the R factor. The others use both R and either the Q factor (GS) or the original data matrix X (CSNE and hybrid). Note that other methods which only use R (e.g., methods based on hyperbolic transformations) will show a similar loss of accuracy as the LINPACK algorithm.

Our results indicate that in cases where, for example, outliers occur, the CSNE and the GS algorithms are the safest choices. If accuracy and efficiency are both important, then the hybrid method may be a better alternative than the LINPACK algorithm. Further study is needed in deciding how to choose the tolerance used in the hybrid algorithm to switch between the CSNE and LINPACK algorithms.

We have also shown that the new methods for downdating R can be extended to downdate R^{-1}. More numerical experiments are needed in this area.

Acknowledgments. A large part of this work was performed when Å. Björck and L. Eldén visited the Institute for Mathematics and its Applications at the University of Minnesota, Minneapolis, during the Applied Linear Algebra year. We are grateful for the pleasant working conditions and for the stimulating environment.

We are also grateful to R. J. Plemmons for suggesting the application of the algorithms to the problem of downdating the inverse of R.

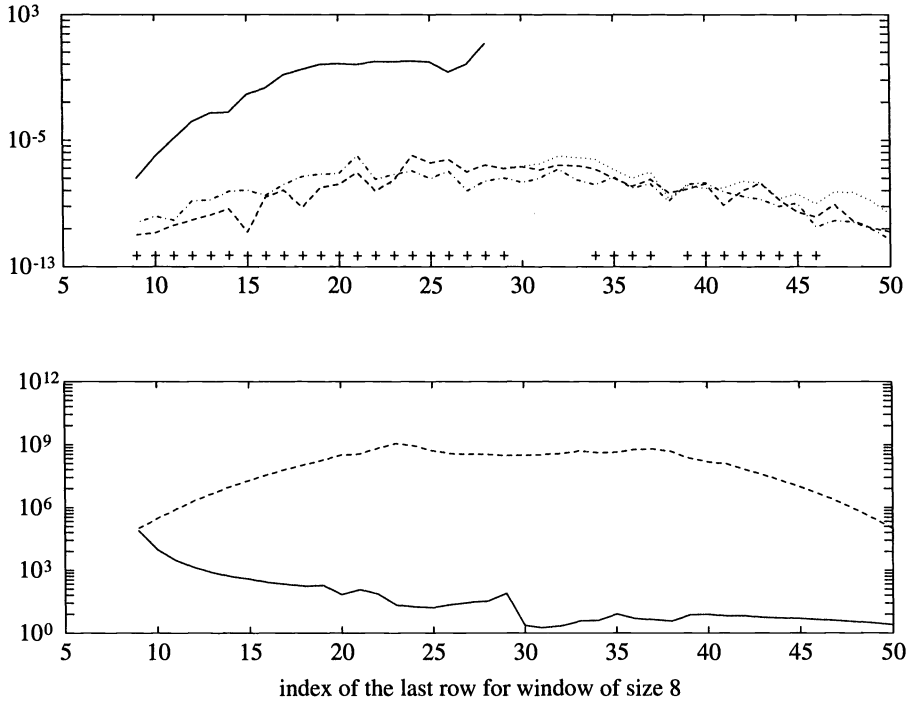


FIG. 8.3. *Test* IIb. *Modified Hilbert matrix with perturbations from a uniform distribution in* $(0, 10^{-9})$.

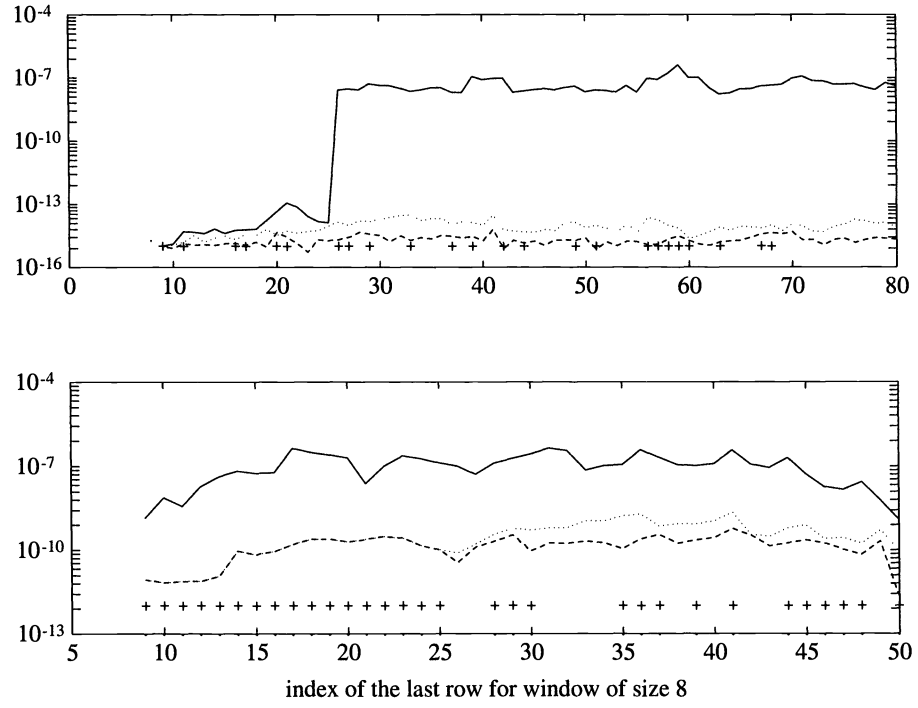


FIG. 8.4. *Test* III. *The inverse* R^{-1} *is downdated. The upper and lower graphs show results for the problems of Test* I *and Test* IIa, *respectively.*

REFERENCES

[1] S. T. ALEXANDER, C.-T. PAN, AND R. J. PLEMMONS, *Analysis of a recursive least squares hyperbolic rotation algorithm for signal processing*, Linear Algebra Appl., 98 (1988), pp. 3–40.

[2] A. A. ANDA AND H. PARK, *Fast plane rotations with dynamic scaling*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 162–174.

[3] M. G. BELLANGER, *The family of fast least squares algorithms for adaptive filtering*, in Mathematics in Signal Processing, J. McWhirter, ed., Clarendon Press, Oxford, 1990, pp. 415–434.

[4] A. BJÖRCK, *Stability analysis of the method of semi-normal equations for least squares problems*, Linear Algebra Appl., 88/89 (1987), pp. 31–48.

[5] J. DANIEL, W. B. GRAGG, L. KAUFMAN, AND G. W. STEWART, *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[6] L. ELDÉN, *Downdating QR decompositions*, in Mathematics in Signal Processing, J. McWhirter, ed., Clarendon Press, Oxford, 1990, pp. 561–574.

[7] W. R. FERNG, G. H. GOLUB, AND R. J. PLEMMONS, *Adaptive Lanczos methods for recursive condition estimation*, J. Numer. Algebra, 1 (1991), pp. 1–20.

[8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[9] C. C. PAIGE, *Error analysis of some techniques for updating orthogonal decompositions*, Math. Comp., 34 (1980), pp. 465–471.

[10] C.-T. PAN, *A perturbation analysis of the problem of downdating a Cholesky factorization*, Linear Algebra Appl., 183 (1993), pp. 103–116.

[11] C.-T. PAN AND R. J. PLEMMONS, *Least squares modifications with inverse factorizations: parallel implications*, J. Comput. Appl. Math., 27 (1989), pp. 109–127.

[12] B. N. PARLETT, *The symmetric eigenvalue problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[13] L. REICHEL AND W. B. GRAGG, *FORTRAN subroutines for updating the QR decomposition*, ACM Trans. Math. Software, 16 (1990), pp. 369–377.

[14] M. A. SAUNDERS, *Large-scale linear programming using the Cholesky factorization*, Tech. Report CS252, Computer Science Dept., Stanford University, Stanford, CA, 1972.

[15] G. W. STEWART, *The effects of rounding error on an algorithm for downdating a Cholesky factorization*, J. Inst. Math. Appl., 23 (1979), pp. 203–213.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 569–577, April 1994

014

ITERATIVE CONSISTENCY: A CONCEPT FOR THE SOLUTION OF SINGULAR SYSTEMS OF LINEAR EQUATIONS*

M. HANKE†

Abstract. This paper deals with the computation of generalized solutions of singular linear systems of equations by *semi-iterative methods*. A new concept called *iterative consistency* is introduced to characterize linear fixed-point equations which lead to certain prescribed generalized solutions of the original problem. Several properties of this concept are discussed.

Perturbations of such iteratively consistent fixed-point equations give rise to the computation of perturbed limit points. These approximants can still be interpreted as appropriate generalized solutions of the original system. Error expressions and first order derivatives are derived. The results are illustrated by the successive overrelaxation (SOR) and by the symmetric SOR (SSOR) method.

Key words. singular linear systems, iterative methods, SOR method, generalized inverses

AMS subject classifications. 65F10, 65F20

1. Introduction and definition.

Consider the singular linear system of equations

$$A\mathbf{x} = \mathbf{b} + \epsilon =: \hat{\mathbf{b}}, \tag{1.1}$$
$$A \in \mathbb{C}^{m\times n}, \qquad \mathbf{b}, \hat{\mathbf{b}}, \epsilon \in \mathbb{C}^m.$$

Here, $\mathbf{b}$ denotes some available approximation of the correct right-hand side $\hat{\mathbf{b}}$, $\epsilon = \hat{\mathbf{b}} - \mathbf{b}$ the corresponding error, and $\mathbf{x} \in \mathbb{C}^n$ the unknown solution. In engineering applications it is typically assumed that ϵ has certain statistical properties, e.g., that ϵ is a random variable with zero mean (cf. Björck [3]).

Appropriate estimates for $\mathbf{x}$ in (1.1) are

$$\mathbf{x} = A^-\mathbf{b},$$

where here and further on, the term A^- is used to denote some arbitrarily specified *generalized inverse* of A. The reader should be familiar with the basic terminology of this subject. The notations used here agree with those used in the standard reference book by Ben-Israel and Greville [1].

According to the Gauss–Markov theory of linear estimation, it is optimal in a certain sense to select for A^- a *weighted generalized inverse* $A^{(1,2)}_{(W,U)}$ ([1, §III.3]), where W^{-1} is the Hermitian, positive definite covariance matrix associated with ϵ. For $\mathbf{x} = A^{(1,2)}_{(W,U)}\mathbf{b}$, $\mathbf{x}' \in \mathbb{C}^n \setminus \{\mathbf{x}\}$ arbitrary, it is

$$\|\mathbf{b} - A\mathbf{x}\|_W^2 := (\mathbf{b} - A\mathbf{x})^* W(\mathbf{b} - A\mathbf{x}) \le \|\mathbf{b} - A\mathbf{x}'\|_W^2$$

and

$$\|\mathbf{b} - A\mathbf{x}\|_W = \|\mathbf{b} - A\mathbf{x}'\|_W \;\Rightarrow\; \|\mathbf{x}\|_U^2 := \mathbf{x}^* U\mathbf{x} < \|\mathbf{x}'\|_U^2.$$

Thus, $\mathbf{x}$ is the unique solution of a *generalized least squares problem.*

* Received by the editors May 30, 1990; accepted for publication (in revised form) October 14, 1992. This research was supported by the Deutsche Forschungsgemeinschaft (DFG). Part of this work has been published in the author's doctoral dissertation.

† Institut für Praktische Mathematik, Universität Karlsruhe, Englerstr. 2, W-7500 Karlsruhe, Germany.

The idea behind this paper is to compute linear approximations $\mathbf{x}_k$ fulfilling

$$\mathbf{x}_k = X_k \mathbf{b} \;\to\; A^{(1,2)}_{(W,U)} \mathbf{b}, \qquad k \to \infty. \tag{1.2}$$

More precisely, it will be assumed that $\mathbf{x}_k$ are *polynomial iterates* of a fixed-point equation

$$\mathbf{x} = T\mathbf{x} + Q\mathbf{b}, \qquad Q \in \mathbb{C}^{n\times m},\ T \in \mathbb{C}^{n\times n}, \tag{1.3}$$

i.e., there exist polynomials q_{k-1} of degree $k-1$ such that

$$\mathbf{x}_k = q_{k-1}(T)Q\mathbf{b}. \tag{1.4}$$

Such $\mathbf{x}_k$ may be computed efficiently by a *semi-iterative method* [7], [6],

$$\mathbf{x}_k = \mu_{0,k}(T\mathbf{x}_{k-1} + Q\mathbf{b}) + \mu_{1,k}\mathbf{x}_{k-1} + \cdots + \mu_{k,k}\mathbf{x}_0, \qquad \mathbf{x}_0 = \mathbf{0},$$

$$\mu_{0,k} \neq 0, \quad \mu_{0,k} + \cdots + \mu_{k,k} = 1, \quad k \geq 0,$$

where the weights $\mu_{j,k}$ are the coefficients of the recursive definition of the so-called *residual polynomials*

$$p_k(z) = 1 + (z-1)q_{k-1}(z).$$

Designing such a semi-iterative method, one must pay special attention to the difficulties arising from the potential singularity of (1.3): Relevant to this, it has been shown by Eiermann, Marek, and Niethammer [6] how to select polynomials q_{k-1} such that

$$q_{k-1}(T) \to (I-T)^{(d)}, \qquad k \to \infty,$$

the *Drazin generalized inverse* of $I - T$.

Comparing (1.2) and (1.4), this leads to the stipulation

$$(I-T)^{(d)}Q\mathbf{b} = A^{(1,2)}_{(W,U)}\mathbf{b},$$

and, due to the unknown value of ϵ in (1.1), this requirement should be fulfilled for every $\mathbf{b} \in \mathbb{C}^m$.

DEFINITION 1.1. *The fixed-point equation* (1.3) *is called iteratively consistent with A via the generalized inverse A^- of A if*

$$(I-T)^{(d)}Q = A^-. \tag{1.5}$$

It is called iteratively consistent with A if (1.5) *is a weighted generalized inverse of A.*

This definition, which was introduced in the author's dissertation [9], differs significantly from definitions of Young [17], Kammerer and Plemmons [12], and Berman and Neumann [2]. In these works, $\mathbf{b}$ and $Q\mathbf{b}$ were assumed to be fixed, and, instead of (1.5), conditions were imposed on certain sets of generalized solutions to (1.1) and (1.3).

In the following sections, fixed-point equations are investigated that are iteratively consistent with A. Section 2 presents a survey of certain properties of such equations; some of these results have been derived in an earlier paper with Neumann [10]. Reasonings of perturbation theoretical nature are included in §3. Section 4 contains some applications that include positive definite preconditionings of the normal equations and the SOR and the SSOR methods.

2. Basic properties. In the beginning, the special case is considered where A is of full column rank. This is a common assumption that, for itself, is of interest.

PROPOSITION 2.1. *Let A be of full column rank. If* (1.3) *is iteratively consistent with A then $T = I - QA$.*

Proof. Let A^- denote the weighted generalized inverse in (1.5). Since $\mathcal{N}(A) = \{\mathbf{0}\}$, it follows that $A^-A = I$. Inserting this in (1.5) yields

$$(I - T)^{(d)}QA = A^-A = I,$$

whence $I - T$ is regular. Multiplying $I - T$ from the left verifies the assertion. □

Compare Proposition 2.1 with analogous characterizations of Young [17, Thm. 3-2.6] and Chen [5, Thm. 2.2-3] or [10, Thm. C] concerning their definition of consistency.

If $T = I - QA$ then the fixed-point equation (1.3) may be written as

$$\mathbf{x} = \mathbf{x} + Q(\mathbf{b} - A\mathbf{x}) \tag{2.1}$$

and thus looks like a *preconditioning* of the original problem; Q is called the *preconditioner*. Other fixed-point equations have been derived in the literature from *splittings* of A; they are not considered here. However, many of the present results can be carried over to such equations (cf. [9]): The clue to doing this may be found in [10, Thm. 4].

In the following theorem ([10, Thm. 2]) iteratively consistent fixed-point equations with A are characterized.

THEOREM 2.2. *The following four statements are equivalent:*

(i) *The fixed-point equation* (2.1) *is iteratively consistent with A,*
(ii) $\mathcal{N}(AQ) \oplus \mathcal{R}(A) = \mathbb{C}^m$,
(iii) $\mathcal{R}(QA) \oplus \mathcal{N}(A) = \mathbb{C}^n$,
(iv) $\operatorname{index}(QA) \leq 1$ *and* $\mathcal{R}(A) \cap \mathcal{N}(Q) = \{\mathbf{0}\}$.

Notice that equations of the form (2.1) automatically lead to a $\{2\}$-inverse (1.5) of A ([10, Lemma 1]). Thus, for fixed-point equations of such a form, iterative consistency with A is equivalent to iterative consistency with A via some $\{1\}$-inverse of A.

Let one of the conditions of Theorem 2.2 be fulfilled and define A^- as in (1.5). Then,

$$\mathcal{N}(AQ) = \mathcal{N}(A^-), \qquad \mathcal{R}(QA) = \mathcal{R}(A^-) \tag{2.2}$$

([10, Thm. 2]) and

$$\mathcal{N}(QA) = \mathcal{N}(A), \qquad \mathcal{R}(AQ) = \mathcal{R}(A) \tag{2.3}$$

([10, Coro. 3]). With the help of these equations, it is easy to characterize those preconditioners Q for which A^- is a $\{1,2,3\}$- or a $\{1,2,4\}$-inverse of A; the case $A^- = A^\dagger$ has been dealt with in [10].

Finally, it must be emphasized that iterative consistency with A implies that index $(QA) \leq 1$. This enables the construction of efficient semi-iterative schemes (1.4). Such schemes were explored in [9], and the corresponding results will be published elsewhere.

3. Derivatives and error terms of Wedin's type. The following analysis concerns the stability of (1.5) relative to perturbations of the preconditioner. Therefore, it will be convenient to consider perturbations of oblique projectors; closely related results may be found in Nashed [14] and in Golub and Pereyra [8].

Let $B, B_\omega \in \mathbb{C}^{n\times n}$ be such that $\operatorname{index}(B) = \operatorname{index}(B_\omega) = 1$; occasionally, B_ω is required to be a continuous matrix function of $\omega \in \mathbb{C}^r$ fulfilling the above properties in an environment of $\omega = 0$ with $B_0 = B$.

THEOREM 3.1. *Let* $P = P_{\mathcal{R}(B),\mathcal{N}(B)}$, $P_\omega = P_{\mathcal{R}(B_\omega),\mathcal{N}(B_\omega)}$. *Then*

$$P_\omega - P = B_\omega^{\#}(B_\omega - B)(I - P) + (I - P_\omega)(B_\omega - B)B^{\#}. \tag{3.1}$$

If B_ω *has a Fréchet derivative* $\mathbf{D}B$ *in* $\omega = 0$ *and if the rank of* B_ω *is constant in an environment of* $\omega = 0$, *then* P_ω *is Fréchet differentiable in* $\omega = 0$ *with derivative*

$$\mathbf{D}P = B^{\#}\mathbf{D}B(I - P) + (I - P)\mathbf{D}BB^{\#}.$$

Proof. Equation (3.1) is easily checked using the identities

$$P = B^{\#}B = BB^{\#}, \qquad P_\omega = B_\omega^{\#}B_\omega = B_\omega B_\omega^{\#}.$$

Under the given assumptions, $B_\omega^{\#} \to B^{\#}$ as $\omega \to 0$ [4] and the derivative $\mathbf{D}P$ can be obtained from (3.1) by letting $\omega \to 0$. □

The following lemma is required to prove the main result of this section.

LEMMA 3.2. *Let* (2.1) *be iteratively consistent with* A. *Then* $\operatorname{index}(AQ) \le 1$ *and*

$$(QA)^{\#}Q = Q(AQ)^{\#}, \qquad (AQ)^{\#}A = A(QA)^{\#}. \tag{3.2}$$

Proof. Define A^- as in (1.5). It follows from (2.2) and (2.3) that $\operatorname{index}(AQ) \le 1$ because A^- is a weighted generalized inverse.

Now, $\mathbf{x} \in \mathcal{N}(AQ)$ implies $Q(AQ)^{\#}\mathbf{x} = \mathbf{0}$ as well as $Q\mathbf{x} \in \mathcal{N}(A) \subset \mathcal{N}(QA)$. Therefore, $(QA)^{\#}Q\mathbf{x} = \mathbf{0}$, also.

Let $\mathbf{x} \in \mathcal{R}(AQ)$. Since $\mathcal{R}(AQ) = \mathcal{R}((AQ)^2)$, there exists some vector $\mathbf{u} \in \mathbb{C}^m$ such that $\mathbf{x} = (AQ)^2\mathbf{u}$. As a consequence,

$$\begin{aligned}(QA)^{\#}Q\mathbf{x} &= (QA)^{\#}(QA)^2Q\mathbf{u} = QAQ\mathbf{u}\\ &= Q(AQ)^{\#}(AQ)^2\mathbf{u} = Q(AQ)^{\#}\mathbf{x}.\end{aligned}$$

The first equation in (3.2) has thus been verified for $\mathbf{x} \in \mathbb{C}^n = \mathcal{N}(AQ) \oplus \mathcal{R}(AQ)$; the second equation follows in the same way. □

Let Q_ω be an approximation of $Q \in \mathbb{C}^{n\times m}$. As before, Q_ω may be a matrix function of $\omega \in \mathbb{C}^r$ with $Q_0 = Q$. Theorem 3.3 considers the consistency of the perturbed fixed-point equation

$$\mathbf{x} = \mathbf{x} + Q_\omega(\mathbf{b} - A\mathbf{x}). \tag{3.3}$$

THEOREM 3.3. *Let* (2.1) *be iteratively consistent with* A *via the weighted generalized inverse* A^- *and let* $\mathcal{R} = \mathcal{R}(A^-)$, $\mathcal{N} = \mathcal{N}(A^-)$. *If* $Q_\omega - Q$ *is sufficiently small, then* (3.3) *is iteratively consistent with* A. *If the latter holds, and if* A_ω^- *denotes the corresponding weighted generalized inverse of* A, *then*

$$A_\omega^- - A^- = P_{\mathcal{N}(A),\mathcal{R}(Q_\omega A)}(Q_\omega - Q)(AQ)^{\#} + (Q_\omega A)^{\#}(Q_\omega - Q)P_{\mathcal{N},\mathcal{R}(A)}. \tag{3.4}$$

If Q_ω is Fréchet differentiable in $\omega = 0$ with derivative $\mathbf{D}Q$ then the Fréchet derivative of A_ω^- in $\omega = 0$ exists and is given by

$$\mathbf{D}A^- = P_{\mathcal{N}(A),\mathcal{R}}\mathbf{D}Q(AQ)^\# + (QA)^\#\mathbf{D}QP_{\mathcal{N},\mathcal{R}(A)}.$$

Proof. Recall that $\mathcal{R}$ and $\mathcal{N}(A)$ are complementary subspaces. By Theorem 2.2, $\mathcal{N}(A) = \mathcal{N}(AQA)$, whence there exists a positive number α such that

$$\|AQA\mathbf{x}\| \geq \alpha \quad \forall\, \mathbf{x} \in \mathcal{R},\ \|\mathbf{x}\| = 1$$

($\|\cdot\|$ always denotes the Euclidean norm or the spectral norm, respectively). If $\|Q_\omega - Q\| < \alpha/\|A\|^2$, this implies

$$AQ_\omega A\mathbf{x} \neq \mathbf{0} \quad \forall\, \mathbf{x} \in \mathcal{R}\{\mathbf{0}\}.$$

Thus, $\mathcal{N}(AQ_\omega A) = \mathcal{N}(A)$ so that

$$\mathcal{R}(Q_\omega A) \cap \mathcal{N}(A) = \{\mathbf{0}\}.$$

Since $\dim \mathcal{R}(Q_\omega A) \geq \dim \mathcal{R}(QA)$ as Q_ω is sufficiently close to Q, it follows from Theorem 2.2 (iii) that (3.3) is iteratively consistent with A.

A fundamental equality for weighted generalized inverses is used next (cf., e.g., [1, Thm. 2.10c] or [14, p. 348]):

$$A_\omega^- = P_{\mathcal{R}(A_\omega^-),\mathcal{N}(A)}A^-P_{\mathcal{R}(A),\mathcal{N}(A_\omega^-)}. \tag{3.5}$$

Since $\mathcal{N}(A) = \mathcal{N}(QA) = \mathcal{N}(Q_\omega A)$ and $\mathcal{R}(A) = \mathcal{R}(AQ) = \mathcal{R}(AQ_\omega)$ (cf. (2.3)), it is

$$P_{\mathcal{R}(A_\omega^-),\mathcal{N}(A)} = (Q_\omega A)^\# Q_\omega A, \qquad P_{\mathcal{R}(A),\mathcal{N}(A_\omega^-)} = (AQ_\omega)^\# AQ_\omega.$$

Substituting (3.1) for $P_\omega = P_{\mathcal{R}(A),\mathcal{N}(A_\omega^-)}$ in (3.5) gives

$$A_\omega^- = P_{\mathcal{R}(A_\omega^-),\mathcal{N}(A)}A^-[P_{\mathcal{R}(A),\mathcal{N}} + (AQ_\omega)^\# A(Q_\omega - Q)P_{\mathcal{N},\mathcal{R}(A)}]$$

because the last term in (3.1) vanishes. Using Lemma 3.2, one has

$$\begin{aligned} A_\omega^- &= P_{\mathcal{R}(A_\omega^-),\mathcal{N}(A)}A^- + P_{\mathcal{R}(A_\omega^-),\mathcal{N}(A)}P_{\mathcal{R}(A^-),\mathcal{N}(A)}(Q_\omega A)^\#(Q_\omega - Q)P_{\mathcal{N},\mathcal{R}(A)} \\ &= P_{\mathcal{R}(A_\omega^-),\mathcal{N}(A)}A^- + (Q_\omega A)^\#(Q_\omega - Q)P_{\mathcal{N},\mathcal{R}(A)}. \end{aligned}$$

Now (3.4) follows in a similar way by inserting (3.1) for $P_\omega = P_{\mathcal{R}(A_\omega^-),\mathcal{N}(A)}$.

The derivative is obtained from (3.4) by letting $\omega \to 0$. □

Theorem 3.3 may be interpreted as follows: Iterative consistency with A is a property that remains valid under small perturbations of the preconditioner; in a sense, iterative consistency with A is thus a *robust property.*

Different proofs are possible to derive (3.4) and the present one may not be the most straightforward one. It is given, though, because it is based on formula (3.5). In fact, it will be seen in the following section that in certain special cases the projectors in (3.5) can be expressed explicitly in terms of Q and Q_ω only.

Formulas of type found in (3.1) and (3.4) are similar to the well-known expression of Wedin [16] for the difference of the Moore–Penrose inverses of A and B:

$$B^\dagger - A^\dagger = -B^\dagger(B - A)A^\dagger + P_{\mathcal{N}(B)}(B - A)^\star(AA^\star)^\dagger + (B^\star B)^\dagger(B - A)^\star P_{\mathcal{R}(A)^\perp}.$$

The reader may find further formulae of this kind together with some applications in the survey paper of Stewart [15].

Estimates for the difference of two weighted generalized inverses of the *same* matrix A may be found in [14].

4. Some applications. Let (1.3) have the form

$$\mathbf{x} = \mathbf{x} + D^{-1}A^*(\mathbf{b} - A\mathbf{x}), \qquad D \in \mathbb{C}^{n\times n}. \tag{4.1}$$

If D is Hermitian, positive definite, the *preconditioned conjugate gradient method* may be applied for the iteration of (4.1). With such a choice of D, it is easy to see that (4.1) is iteratively consistent with A.

Let $\mathbf{y} \in \mathbb{C}^m$ be some arbitrary vector; if

$$\mathbf{y}^*AD^{-1}A^*\mathbf{y} = \|D^{-1/2}A^*\mathbf{y}\|^2 = 0,$$

then this implies $D^{-1/2}A^*\mathbf{y} = \mathbf{0}$, whence $\mathbf{y} \in \mathcal{R}(A)^\perp$. Therefore,

$$\mathcal{N}(AD^{-1}A^*) = \mathcal{R}(A)^\perp$$

and the assertion follows from Theorem 2.2(ii).

As an example, let D be the (block-)diagonal part of A^*A: In this case, (4.1) is the fixed-point equation of the (*block-*)*Jacobi method* for the normal equations system. Applications to SOR and SSOR theory follow from the investigation of fixed-point equations of the form

$$\mathbf{x} = \mathbf{x} + \omega D_\omega^{-1}A^*(\mathbf{b} - A\mathbf{x}), \qquad D_\omega \in \mathbb{C}^{n\times n}, \tag{4.2}$$

where D_ω is connected to D in (4.1).

THEOREM 4.1. *Let D and D_ω be regular matrices in $\mathbb{C}^{n\times n}$ and $E_\omega = D_\omega - D$. Assume that* (4.1) *is iteratively consistent with A via the weighted generalized inverse A^-. Then, the fixed-point equation* (4.2) *is iteratively consistent with A if and only if $I + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega$ is regular, in which case the corresponding generalized inverse of A is defined as*

$$A_\omega^- = (I + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega)^{-1}A^-. \tag{4.3}$$

Proof. Denote $\mathcal{R}_\omega = \mathcal{R}(D_\omega^{-1}A^*A)$ and let

$$Z = I + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega = P_{\mathcal{R}(A^-),\mathcal{N}(A)} + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}D_\omega.$$

Since $D^{-1}D_\omega$ maps $\mathcal{R}_\omega$ onto $\mathcal{R}(A^-) = \mathcal{R}(D^{-1}A^*A)$, it follows that

$$Z : \begin{cases} \mathcal{R}_\omega & \to \quad \mathcal{R}(A^-), \\ \mathcal{N}(A) & \to \quad \mathcal{N}(A). \end{cases} \tag{4.4}$$

Let Z be regular. Since (4.1) is iteratively consistent with A, Theorem 2.2(iii) states that $\mathcal{R}(A^-) \oplus \mathcal{N}(A) = \mathbb{C}^n$. From (4.4) and the regularity of Z, it follows that

$$\mathcal{R}_\omega \oplus \mathcal{N}(A) = \mathbb{C}^n$$

(note that $\dim \mathcal{R}_\omega = \dim \mathcal{R}(A^-)$). Therefore, (4.2) is iteratively consistent with A. The respective weighted generalized inverse A_ω^- fulfills (cf. (2.2))

$$\mathcal{R}(A_\omega^-) = \mathcal{R}_\omega, \qquad \mathcal{N}(A_\omega^-) = \mathcal{N}(AD_\omega^{-1}A^*) = \mathcal{R}(A)^\perp = \mathcal{N}(A^-).$$

Using (3.5),

$$A_\omega^- = P_{\mathcal{R}_\omega,\mathcal{N}(A)}A^-P_{\mathcal{R}(A)} = P_{\mathcal{R}_\omega,\mathcal{N}(A)}A^-.$$

With the help of (4.4), it is readily checked that

$$P_{\mathcal{R}_\omega,\mathcal{N}(A)} = Z^{-1}P_{\mathcal{R}(A^-),\mathcal{N}(A)}Z,$$

whence (4.3) follows:

$$A_\omega^- = Z^{-1}P_{\mathcal{R}(A^-),\mathcal{N}(A)}(I + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega)A^- = Z^{-1}A^-.$$

Now, let (4.2) be iteratively consistent with A and suppose $\mathbf{x} \in \mathcal{N}(Z)$. This implies

$$\mathbf{0} = (I + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega)\mathbf{x} = \mathbf{x} + \mathbf{z}$$

with $\mathbf{z} \in \mathcal{N}(A)$. Therefore, $\mathbf{x} \in \mathcal{N}(A)$ and, consequently,

$$\mathbf{0} = (P_{\mathcal{R}(A^-),\mathcal{N}(A)} + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}D_\omega)\mathbf{x} = P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}D_\omega\mathbf{x}.$$

Hence, $D^{-1}D_\omega\mathbf{x} \in \mathcal{R}(A^-)$ so that

$$\mathbf{x} \in D_\omega^{-1}D(\mathcal{R}(A^-)) = \mathcal{R}_\omega.$$

By assumption, $\mathcal{R}_\omega \cap \mathcal{N}(A) = \{\mathbf{0}\}$, which implies $\mathcal{N}(Z) = \{\mathbf{0}\}$. □

COROLLARY 4.2. *Under the conditions of Theorem* 4.1, (4.2) *is iteratively consistent with* A *via* A_ω^- *of* (4.3) *if*

$$\|P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega\| < 1.$$

In this case, the following estimate is valid:

$$\frac{\|A_\omega^- - A^-\|}{\|A^-\|} \leq \frac{\|P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega\|}{1 - \|P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega\|}. \tag{4.5}$$

Proof. If $\|P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega\| < 1$ then $I + P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega$ is invertible, whence (4.2) is iteratively consistent with A via A_ω^- of (4.3). Inserting the convergent Neumann series expansion gives

$$A_\omega^- = \sum_{\nu=0}^{\infty}(-P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega)^\nu A^-,$$

whence

$$\|A_\omega^- - A^-\| \leq \sum_{\nu=1}^{\infty}\|P_{\mathcal{N}(A),\mathcal{R}(A^-)}D^{-1}E_\omega\|^\nu \, \|A^-\|.$$

This proves (4.5). □

Suppose that the triangular splitting of A^*A is given, i.e., let

$$A^*A = D - L - L^*, \tag{4.6}$$

where L is a strictly lower (block-)triangular matrix and D is a (block-)diagonal matrix. Choosing

$$D_\omega = D - \omega L,$$

(4.2) is the fixed-point equation of the SOR method relative to (4.6). If

$$D_\omega = \frac{1}{2-\omega}(D-\omega L)D^{-1}(D-\omega L^*),$$

then (4.2) corresponds to the SSOR method (cf. [17]). In both instances it follows from results of Keller [13] that

$$\mathcal{R}(D_\omega^{-1}A^*A) \oplus \mathcal{N}(D_\omega^{-1}A^*A) = \mathbb{C}^n, \qquad 0 < \omega < 2,$$

which proves the iterative consistency of (4.2) with A for ω in the given range. The respective weighted generalized inverses are defined by (4.3). (For the SSOR method, one should substitute $E_\omega = 2D_\omega - D$.)

Moreover, since $E_\omega \to 0$ as $\omega \to 0$, the conditions of Corollary 4.2 are fulfilled for ω sufficiently close to 0.

For a more specific example, let A be normalized such that $D = I$. Choose $0 < \omega < \min\{2, 1/\|L\|\}$. With this choice of ω the SOR method converges according to [13]; denote the limit by $\hat{\mathbf{x}}$. As in (4.5), it follows that

$$\frac{\|\hat{\mathbf{x}} - A^\dagger \mathbf{b}\|}{\|A^\dagger \mathbf{b}\|} \le \omega \frac{\|L\|}{1-\omega\|L\|}.$$

Using (4.3), the following à-posteriori estimate may be obtained:

$$\|\hat{\mathbf{x}} - A^\dagger \mathbf{b}\| \le \omega \|L\hat{\mathbf{x}}\|.$$

Analogous results have been developed in [9] for fixed-point equations of the form

$$\mathbf{x} = \mathbf{x} + \omega A^* D_\omega^{-1}(\mathbf{b} - A\mathbf{x}),$$

where $D_\omega \in \mathbb{C}^{m\times m}$ is a regular matrix; some of these results have been published in [11].

Acknowledgments. I would like to thank all those who contributed to the accomplishment of my dissertation, especially Professor Wilhelm Niethammer for raising my interest in this topic and advising me through this thesis, Professor Michael Neumann for many valuable discussions and a pleasant stay in Connecticut, and the Deutsche Forschungsgemeinschaft for their financial support.

REFERENCES

[1] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Wiley, New York, London, Sydney, Toronto, 1974.

[2] A. Berman and M. Neumann, *Consistency and splittings*, SIAM J. Numer. Anal., 13 (1976), pp. 877–888.

[3] A. Björck, *Least squares methods*, in Handbook of Numerical Analysis, Vol. I: Finite Difference Methods. Solutions of Equations in $\mathbf{R}^n$, P. G. Ciarlet and J. L. Lions, eds., Elsevier North-Holland, 1990.

[4] S. L. Campbell, *Differentiation of the Drazin inverse*, SIAM J. Appl. Math., 30 (1976), pp. 703–707.

[5] Y.-T. Chen, *Iterative Methods for Linear Least Squares Problems*, Ph.D. thesis, University of Waterloo, Ontario, Canada, 1975.

[6] 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.

[7] M. EIERMANN, W. NIETHAMMER, AND R. S. VARGA, *A study of semi-iterative methods for nonsymmetric systems of linear equations*, Numer. Math., 47 (1985), pp. 505–533.

[8] G. H. GOLUB AND V. PEREYRA, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.

[9] M. HANKE, *Iterative Lösung gewichteter linearer Ausgleichsprobleme*, Ph.D. thesis, Universität Karlsruhe, Karlsruhe, West-Germany, 1989.

[10] M. HANKE AND M. NEUMANN, *Preconditionings and splittings for rectangular systems*, Numer. Math., 57 (1990), pp. 85–95.

[11] M. HANKE AND W. NIETHAMMER, *On the use of small relaxation parameters in Kaczmarz's method*, Z. Angew. Math. Mech., 70 (1990), pp. T575–T576.

[12] W. J. KAMMERER AND R. J. PLEMMONS, *Direct iterative methods for least-squares-solutions to singular operator equations*, J. Math. Anal. Appl., 49 (1975), pp. 512–526.

[13] H. B. KELLER, *On the solution of singular and semidefinite linear systems by iteration*, SIAM J. Numer. Anal., 2 (1965), pp. 281–290.

[14] M. Z. NASHED, *Perturbations and approximations for generalized inverses and linear operator equations*, in Generalized Inverses and Applications, M. Z. Nashed, ed., New York, San Francisco, London, 1976, Academic Press, pp. 325–396.

[15] G. W. STEWART, *On the perturbation of pseudo-inverses, projections and linear least squares problems*, SIAM Rev., 19 (1977), pp. 634–662.

[16] P.-A. WEDIN, *Perturbation theory for pseudo-inverses*, BIT, 13 (1973), pp. 217–232.

[17] D. M. YOUNG, *Iterative Solution of Large Linear Systems.*, Academic Press, New York, London, 1971.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 578–591, April 1994

015

ALGORITHMS FOR COMPUTING BASES FOR THE PERRON EIGENSPACE WITH PRESCRIBED NONNEGATIVITY AND COMBINATORIAL PROPERTIES*

MICHAEL NEUMANN† AND HANS SCHNEIDER‡

Abstract. Let P be an $n \times n$ nonnegative matrix. In this paper the authors introduce a method called the SCANBAS algorithm for computing a union of (Jordan) chains $\mathcal{C}$ corresponding to the Perron eigenvalue of P, such that $\mathcal{C}$ consists of nonnegative vectors only and such that at each height, $\mathcal{C}$ contains the maximal number of nonnegative vectors of that height possible in a height basis for the Perron eigenspace of P. It is further shown that $\mathcal{C}$ can be extended to a height basis for the Perron eigenspace of P. The chains are extracted from transform components of P that are, in turn, polynomials in P. When the Perron eigenspace has a Jordan basis consisting of nonnegative vectors only, this algorithm computes such a basis. The paper concludes with various examples computed by the algorithm using MATLAB. The work here continues and deepens work on computing nonnegative bases for the Perron eigenspace from polynomials in the matrix already begun by Hartwig, Neumann, and Rose and by Neumann and Schneider.

Key words. nonnegative matrices, M-matrices, Perron eigenspace, computations

AMS subject classifications. 15A15, 15A48, 65F15

1. Introduction.

In this paper we continue an investigation begun by Hartwig, Neumann, and Rose [6] and Neumann and Schneider [11] on nonnegative and combinatorial properties of bases for the Perron eigenspace of a nonnegative matrix, which can be extracted from certain polynomials in the matrix.

More specifically, let P be an $n \times n$ nonnegative matrix and $\rho(P)$ its spectral radius which is well known to be an eigenvalue of P, called its *Perron root.* A more comprehensive explanation and detailed background to some of the concepts used in this introduction and appropriate references are given in the next sections. Let Z be the eigenprojection of P at $\rho(P)$. In [6] it was shown that for sufficiently small $\epsilon > 0$, the matrix

$$J(\epsilon) = [(\epsilon + \rho(P))I - P]^{-1}Z \tag{1.1}$$

is nonnegative and its columns contain a basis of nonnegative vectors for the (generalized) eigenspace of P corresponding to $\rho(P)$ known as the *Perron eigenspace of* P. Furthermore, an algorithm for computing ϵ and hence a method for computing such a basis was suggested in [6, Thm. 2.2].

In [11] it was observed that since $J(\epsilon)$ is an analytic function in P, it is a polynomial in P, so that if P is put, say, in block lower triangular Frobenius normal form, then $J(\epsilon)$ would be a block lower triangular matrix conforming to the block partitioning in the Frobenius normal form of P. Thus combinatorial properties possessed by certain nonnegative bases for the Perron eigenspace of P that were present in the first proofs of the existence of such basis for the Perron eigenspace of P obtained by Rothblum [13] and Richman and Schneider [12] could be extracted from the columns

* Received by the editors November 11, 1991; accepted for publication October 14, 1992.

† Department of Mathematics, University of Connecticut, Storrs, Connecticut 06269-3009 (neumann@uconnvm.bitnet). This author's research was supported by National Science Foundation grants DMS-8901860 and DMS-9007030.

‡ Department of Mathematics, University of Wisconsin, Madison, Wisconsin 53706 (hans @math.wisc.edu). This author's research was supported by National Science Foundation grants DMS-8901445 and ECS-8718971.

of $J(\epsilon)$, and this idea led to the investigation in [11]. The combinatorial properties that the authors of [11] had in mind to recapture from the columns of $J(\epsilon)$ were access relations in the directed graph of P or, more precisely, in the block directed graph that can be associated with the Frobenius normal form of P known as the *reduced graph of* P. Indeed, in [11], it was shown that for sufficiently small $\epsilon > 0$, nonnegative bases from the columns of $J_T(\epsilon)$ could be chosen that are *strongly combinatorial.* (See §2 and [11] for precise definitions of these terms.)

The purpose of this paper is to go one step deeper in search of combinatorial and algebraic properties of bases that can be extracted from the columns of $J(\epsilon)$ and certain other nonnegative matrices that are polynomials in the matrix P and to compute such bases. Actually, we think of $J^{(0)}(\epsilon) := \epsilon J(\epsilon)$ as a zeroth *transform component of* P. Other nonnegative matrices that are polynomials in P, which we will work with later, are the higher-order transform components

$$J^{(k)}(\epsilon) = \epsilon^{k+1}(P - \rho(P)I)^k J(\epsilon), \quad k = 1, \ldots, \nu - 1.$$

Here ν is the index of the Perron root as an eigenvalue of P.

Hershkowitz [7] defines the *peak characteristic tuple* $(\xi_1, \ldots, \xi_\nu)$ of the Perron root as an eigenvalue of P, and he shows that for each $h = 1, \ldots, \nu$, ξ_h is the maximal number of nonnegative vectors of height h in a height basis for the Perron eigenspace and that a height basis for the eigenspace exists with that many nonnegative vectors at each height $h = 1, \ldots, \nu$.

The principal question that we shall answer in this paper is this: Can we extract from the columns of the transformed components $J^{(0)}(\epsilon), \ldots, J^{(\nu-1)}(\epsilon)$, a union $\mathcal{C}$ of nonnegative Jordan chains that contains exactly ξ_h vectors of height h, $h = 1, \ldots, \nu$, where the tuple $(\xi_1, \ldots, \xi_\nu)$ is the peak characteristics mentioned above? Moreover, can this union be extended to a height basis for the eigenspace? We achieve this via a *scanning process of the transform components* that begins by stacking the transform components on each other from the lowest to the highest. We call this process the SCANBAS algorithm. The algorithm is set out in §5 after some preparations and preliminary results from §§2–4. We go on to show in Corollary 4 that if $(\eta_1, \ldots, \eta_\nu)$ is the *height characteristic of the* Perron root as an eigenvalue of P with $\eta_k = \xi_k$ for $k = t, \ldots, \nu$, so that by Hershkowitz and Schneider [8, Thm. (6.6)] there is a Jordan basis for the Perron eigenspace of P corresponding to its Perron root such that all Jordan chains of length t and higher consist of nonnegative vectors only, then our SCANBAS algorithm produces such chains. In particular, if the Perron eigenspace of P has a Jordan basis consisting entirely of nonnegative chains, our SCANBAS algorithm computes such a basis. In §6 we conclude the paper by presenting various examples of bases that were produced by our SCANBAS algorithm implemented by using MATLAB. These examples show that generally $\mathcal{C}$ cannot be extended to a Jordan basis for the eigenspace. We end the section with a brief description of the MATLAB programs that were actually used in the computation of the examples.

Finally, we find it convenient to work and state the results of this paper in terms of the minus M–matrix $A = P - \rho(P)I$ that can be associated with our nonnegative matrix P.

2. Notations and preliminaries. For a positive integer n, we denote by $\langle n \rangle$ the set $\{1, \ldots, n\}$.

In all our considerations we assume that A is an $n \times n$ real matrix given in a block lower triangular form with p square diagonal blocks as follows:

$$A = \begin{pmatrix} A_{1,1} & 0 & \dots & 0 \\ A_{2,1} & A_{2,2} & & 0 \\ \vdots & & \ddots & \vdots \\ A_{p,1} & \dots & \dots & A_{p,p} \end{pmatrix}, \tag{2.1}$$

where each diagonal block is either an irreducible matrix or the 1×1 null matrix. The above form is called the *Frobenius normal form* of A. It is well known that any square matrix is symmetrically permutable to such a form. The *reduced graph of* A, $\mathcal{R}(A)$, is defined to be the graph with vertices $\{1, \dots, p\}$, where (i, j) is an arc from i to j if $A_{i,j} \neq 0$. A vertex i in $\mathcal{R}(A)$ is said to be singular if $A_{i,i}$ is singular. Otherwise the vertex is called *nonsingular*. The set of all singular vertices in $\mathcal{R}(A)$ will be denoted by $\mathcal{S}(A)$. A sequence of vertices $(i_1, \dots, i_k)$ in $\mathcal{R}(A)$ is said to be a *path* from i_1 to i_k if there is an arc in $\mathcal{R}(A)$ from i_j to i_{j+1} *for all* $j \in \langle k-1 \rangle$. The path is said to be *simple* if $i_1, \dots, i_k$ are distinct. The empty path will be considered a simple path linking every vertex $i \in \mathcal{R}(A)$ to itself. If there is a path (in $\mathcal{R}(A)$) from i to j, we write that $i \succeq j$. If $i \neq j$ and there is a path from i to j, we write that $i \succ j$.

Let $x \in R^n$. We partition $x = ((x_1)^T, \dots, (x_p)^T)^T$ in conformity with (2.1). Let $i \in \langle p \rangle$. We say that that the *level of* i in $\mathcal{R}(A)$ is k ($\mathrm{lev}(i) = k$) if the maximal number of singular vertices on a path ending at i is k. We say that the *level of* $x \in R^n$ is k ($\mathrm{lev}(x) = k$) if

$$k = \max\{\mathrm{lev}(i) \mid x_i \neq 0\}.$$

For an $n \times n$ matrix A we denote by:

$N(A)$, the nullspace of A;

$E(A)$, the generalized nullspace of A, viz., $N(A^n)$;

$\nu(A)$, the index of 0 as an eigenvalue of A, viz., the size of the largest Jordan block associated with 0. Where no confusion is likely to arise, we write ν for $\nu(A)$.

We let $Z^{(0)}(A)$ be the eigenprojection of A corresponding to the eigenvalue 0 and we put $Z^{(k)}(A) = A^k Z^{(0)}(A)$, $k = 0, \dots, \nu - 1$. Where no confusion is likely to arise, we write $Z^{(k)}$ for $Z^{(k)}(A)$, $k = 1, \dots, \nu - 1$. The matrices $Z^{(k)}$, $k = 0, \dots, \nu - 1$, are called *the principal components of* A (*corresponding to the eigenvalue* 0). For background material on the principal components, see Lancaster and Tismenetski [10, p. 314] and, in the case of nonnegative matrices, see Neumann and Schneider [11].

Let $\alpha \subseteq \langle n \rangle$. By $A[\alpha]$ we denote the principal submatrix of A whose rows and columns are determined by α. Similarly, for an n-vector x, we denote by $x[\alpha]$ the subvector of x whose entries are indexed by α. For an array C, we use $C \geq 0$ to denote when all its entries are nonnegative numbers. $C > 0$ denotes the fact that $C \geq 0$, but $C \neq 0$. $C \gg 0$ denotes the fact that all of the entries of C are positive numbers.

Let P be an $n \times n$ nonnegative matrix. The Perron Frobenius theory (cf. Berman and Plemmons [2]) tells us that the spectral radius of P, given by the quantity

$$\rho(P) = \max\{|\lambda| : \det(P - \lambda I) = 0\},$$

is an eigenvalue of P that corresponds to a nonnegative eigenvector. In particular, if P is irreducible, then $\rho(P)$ is simple and the corresponding eigenvector is, up to

a multiple by a scalar positive. The matrix $A = P - \rho(P)I$, which has all its off-diagonal entries nonnegative, is the $n \times n$ minus M–matrix that we associate with P and, in several sections of our paper, it will be convenient to work with A rather than with P. (We call A a *minus* M-*matrix* if $-A$ is an M-matrix. For the many equivalent conditions for a real matrix with nonpositive off-diagonal entries to be an M-matrix, see Berman and Plemmons [2, Chap. 6].) Suppose now that $m = \dim(E(A))$. It is known that m is equal to the number of singular vertices in $\mathcal{R}(A)$. Rothblum [13] has shown that $\nu(A)$ is equal to the maximum over all lengths of the simple paths in $\mathcal{R}(A)$, a result we shall refer to as the *Rothblum index theorem.* Let $\mathcal{S}(A) = \{\alpha_1, \ldots, \alpha_m\}$. Rothblum [13] and, independently, Richman and Schneider [12] (See also [14]) have shown that $E(A)$ possesses a basis of nonnegative vectors that is strongly combinatorial in the sense defined in Definition 1.

DEFINITION 1. *Let A be the $n \times n$ minus* M-*matrix given in form* (2.1) *and consider $\mathcal{R}(A)$.*

(i) *A nonnegative basis $u^{(1)}, \ldots, u^{(m)}$ is a* (nonnegatively) proper combinatorial basis *for $E(A)$ if*

$$u^{(j)}[i] > 0 \Rightarrow i \succeq \alpha_j$$

and

$$u^{(j)}[\alpha_j] \gg 0$$

for all $i \in \langle p \rangle$ and $j \in \langle m \rangle$.

(ii) *A nonnegative basis $u^{(1)}, \ldots, u^{(m)}$ is called a* (nonnegatively) strongly combinatorial basis *for $E(A)$ if*

$$u^{(j)}[i] = \begin{cases} \gg 0 & \text{iff } i \succeq \alpha_j, \\ 0 & \text{otherwise.} \end{cases}$$

Let $x \in E(A)$. We say that the *height of* x is k ($\mathrm{ht}(x) = k$) if k is the smallest nonnegative integer such that $A^k x = 0$. The *fundament of* x is, according to Hershkowitz and Schneider [9], the vector $A^{k-1}x$. For a set of vectors $\mathcal{S} = \{x, y, \ldots\}$ in $E(A)$, the *fundament of* $\mathcal{S}$ is the set of vectors formed from the fundaments of the elements of $\mathcal{S}$.

Let A be a minus M-matrix. Then it is known that $\mathrm{ht}(x) \leq \mathrm{lev}(x)$ for all $x \in E(A)$, cf. [8, Cor. (4.17)]. A vector $x \in E(A)$ for which $\mathrm{ht}(x) = \mathrm{lev}(x)$ is called a *peak vector.* If x is a peak vector, then $\mathrm{lev}(Ax) = \mathrm{lev}(x) - 1$ by [9, Prop. 6.5]. Also every nonnegative vector in $E(A)$ is a peak vector.

3. Jordan bases with nonnegative chains. Let $A \in R^{nn}$ be given as in (2.1). We shall use the following notation subsequently:

R^n_+ – the set of nonnegative vectors in $\mathcal{R}^n$.
$F = E(A) \cap R^n_+$.
$F_k = N(A^k) \cap R^n_+$, $k = 0, \ldots, \nu$.
$E_k = \mathrm{span}(F_k)$, $k = 0, \ldots, \nu$.
$S_k = A^{k-1}E_k$, $k = 1, \ldots, \nu$.

Henceforth, we let $A \in R^{nn}$ be a (singular) minus M-matrix of index ν. Since every nonnegative vector in $E(A)$ is a peak vector, we have the following graph theoretic classification of F_k and E_k.

$$F_k = \{x \in F : \mathrm{lev}(x) \leq k\}, \quad k = 0, \ldots, \nu. \tag{3.1}$$

$$E_k = \{x \in E(A) \ : \ \mathrm{lev}(x) \leq k \ \}, \quad k = 0, \ldots, \nu. \tag{3.2}$$

We note that

$$\{0\} = E_0 \subseteq E_1 \subseteq \cdots \subseteq E_\nu = E(A). \tag{3.3}$$

Now by (3.1) and (3.2) and because $\mathrm{lev}(Ax) = \mathrm{lev}(x) - 1$ for $x \in F_k$, it follows that

$$AE_k \subseteq E_{k-1}, \quad k = 1, \ldots, \nu. \tag{3.4}$$

Hence we have

$$\{0\} \subseteq S_\nu \subseteq \cdots \subseteq S_1 = E_1 \subseteq N(A). \tag{3.5}$$

DEFINITION 2. (i) (Hershkowitz and Schneider [8, Def. (2.6)]) *The* height characteristic *of A is defined to be the ν-tuple*

$$\eta(A) = (\eta_1(A), \ldots, \eta_\nu(A)),$$

where $\eta_k(A) = \dim(N(A^k)) - \dim N(A^{k-1})$, $k = 1, \ldots, \nu$.

(ii) *The* peak characteristic *of A is defined to be the ν-tuple*

$$\xi(A) = (\xi_1(A), \ldots, \xi_\nu(A)),$$

where $\xi_k(A) = \dim(S_k)$, $k = 1, \ldots, \nu$.

Where no confusion is likely to arise, we denote the height characteristic of A by $\eta = (\eta_1, \ldots, \eta_\nu)$ and the peak characteristic of A by $\xi = (\xi_1, \ldots, \xi_\nu)$. In [7, Def. (4.1)], Hershkowitz defines the peak characteristic of A by letting $\xi_k = \dim E_k - \dim(N(A^{k-1}) \cap E_k)$. Since S_k is isomorphic to $E_k/(N((A)^{k-1}) \cap E_k)$, $k = 1, \ldots, \nu$, it follows that his definition of the peak characersitic of A coincides with the definition given above.

For the sake of completeness, we prove the following proposition that forms part of [7, Thm. (6.5)]. Recall, cf. [8, Def. (3.1)], that a basis $\mathcal{B}$ for E(A) is called a *height basis* if the number of basis elements of height k is equal to η_k, $k = 1, \ldots, \nu$.

PROPOSITION 1. *Let A be a minus* M*-matrix and let $\mathcal{B}$ be a height basis for $E(A)$. Let β_k be the number of peak vectors in $\mathcal{B}$ of height k, $k = 1, \ldots, \nu$. Then $\beta_k \leq \xi_k$, $k = 1, \ldots, \nu$.*

Proof. Let $1 \leq k \leq \nu$ and let $x^{(1)}, \ldots, x^{(s)}$ be peak vectors of height k in the height basis $\mathcal{B}$ for E. Then $x^{(1)}, \ldots, x^{(s)}$ are linearly independent mod $N(A^{k-1})$ by [8, Prop. (3.14)]. Hence $A^{k-1}x^{(1)}, \ldots, A^{k-1}x^{(s)}$ are linearly independent vectors. □

Hershkowitz [7, Thm. (6.5)] also proves that for every minus M-matrix, there exists a height basis that has ξ_k nonnegative vectors of height k, for $k = 1, \ldots, \nu$. In §5 we give an algorithm that (in exact arithmetic) computes such a basis.

4. The transform components. We begin by introducing the (ϵ) transform components of A.

DEFINITION 3. *For $\epsilon > 0$ and for $k = 0, \ldots, \nu - 1$, we define the* kth transform component of A *by*

$$J^{(k)}(\epsilon) = Z^{(k)} + \frac{Z^{(k+1)}}{\epsilon} + \cdots + \frac{Z^{(\nu-1)}}{\epsilon^{\nu-k-1}}. \tag{4.1}$$

In [6] it was shown that provided $\epsilon > 0$ is sufficiently small, a basis of nonnegative vectors for $E(A)$ can be chosen from the columns of $J^{(0)}(\epsilon)$. Moreover, a method was given for determining such ϵ's, cf. [6, Thm. 2.2]. The results of [6] were improved in [11], where it was shown that provided $\epsilon > 0$ is sufficiently small, the nonnegative basis for $E(A)$ chosen from the columns of $J^{(0)}(\epsilon)$ can be chosen to be strongly combinatorial. We now strengthen the results of both papers by showing that the method used in [6] can be adapted to compute ϵ's that ensure that the columns of $J^{(0)}(\epsilon)$ contain a strongly combinatorial basis and that all transform components are nonnegative and have interesting combinatorial properties. To this end, for $1 \le i, j \le p$ and for $0 \le k \le \nu - 1$, let $\mu_{i,j}^{(k)}$ be the least element in $Z_{i,j}^{(k)}$ and let

$$\gamma_{i,j}^{(k)} = |\min\{\mu_{i,j}^{(k)}, 0\}|.$$

We note that by [11, Thm. 1], $\mu_{i,j}^{(d-1)} > 0$, when $d = d(i,j) \ge 1$. Let

$$\mu = \min \frac{\mu_{i,j}^{(d-1)}}{\gamma_{i,j}^{(k)} + \cdots + \gamma_{i,j}^{(d-2)}}, \tag{4.2}$$

where the minimum is taken over all i, j, k such that $1 \le i, j \le p$, $0 \le k \le \nu - 1$, and $d = d(i,j) > k$. We comment that we here take a ratio $p/0$, where $p > 0$, to be $+\infty$.

LEMMA 1. *Let A be a minus* M*-matrix and suppose $1 \le i, j \le p$ and $0 \le k \le \nu - 1$. Let $0 < \epsilon \in \min\{1, \mu\}$, where μ is given by* (4.2).

(i) *If $d(i,j) \le k$, then $J_{i,j}^{(k)}(\epsilon) = 0$.*

(ii) *If $d(i,j) > k$, then $J_{i,j}^{(k)}(\epsilon) \gg 0$.*

Proof. (i) By [11, Lemma 2], if $d(i,j) \le k$, then $Z_{i,j}^{(q)} = 0$ for all q such that $k \le q \le \nu - 1$ and the result follows.

(ii) Let $d = d(i,j) > k$. Then

$$J_{i,j}^{(k)}(\epsilon) = Z_{i,j}^{(k)} + \frac{Z_{i,j}^{(k+1)}}{\epsilon} + \cdots + \frac{Z_{i,j}^{(d-1)}}{\epsilon^{d-k+1}}.$$

By [11, Thm. 1] $Z_{i,j}^{(d-1)} \gg 0$ and so $\mu_{i,j}^{(d-1)} > 0$. Let $0 < \epsilon < 1$ and let α be the least element in $J_{i,j}^{(k)}(\epsilon)$. Then

$$\begin{aligned} \alpha &\ge -\gamma_{i,j}^{(k)} - \cdots - \frac{\gamma^{(d-2)}}{\epsilon^{d-k-2}} + \frac{\mu_{i,j}^{(k-1)}}{\epsilon^{d-k-1}} \\ &\ge -\frac{1}{\epsilon^{d-k-2}}(\gamma_{i,j}^{(k)} + \cdots + \gamma_{i,j}^{(d-2)}) + \frac{\mu_{i,j}^{(k-1)}}{\epsilon^{d-k-1}}. \end{aligned}$$

Hence $\alpha > 0$ if

$$\epsilon < \frac{\mu_{i,j}^{(k-1)}}{\gamma_{i,j}^{(k)} + \cdots + \gamma_{i,j}^{(d-2)}}$$

and the result follows. □

We now make more precise a result mentioned in [11].

COROLLARY 1. *Let $\alpha_1, \ldots, \alpha_m$ be the singular vertices of $\mathcal{R}(A)$. Let $v^{(j)}$ be a column of $J^{(0)}(\epsilon)$ chosen from the columns of the α_jth block column of $J^{(0)}(\epsilon)$, $j = 1, \ldots, m$. Then $v^{(1)}, \ldots, v^{(m)}$ is a strongly combinatorial basis for $E(A)$ and, what is more, they satisfy:*

$$(A^k v^{(j)})_i \gg 0 \quad \text{if } d(i, \alpha_j) > k,$$
$$(A^k v^{(j)})_i = 0 \quad \text{if } d(i, \alpha_j) \leq k.$$

Proof. We observe that $A^k v^{(j)}$ is a column of $J^{(k)}(\epsilon)$ belonging to the α_jth block column $J^{(k)}(\epsilon)$, $k = 0, \ldots, \nu - 1$ and $j = 1, \ldots, m$. □

We note that this basis satisfies the properties of Rothblum, [13, Thm. 3.1]. Additionally we have the following corollary.

COROLLARY 2. *Let $v^{(1)}, \ldots, v^{(m)}$ be a basis of $E(A)$ which satisfies the conclusion of Corollary 1. The subset consisting of those vectors whose level does not exceed k forms a basis for E_k, $k = 0, \ldots, \nu - 1$.*

Proof. It holds that $v^{(1)}, \ldots, v^{(m)}$ is a strongly combinatorial basis for $E(A)$. □

5. The SCANBAS algorithm. From now on we shall assume that ϵ has been chosen so that the transform components $J^{(k)}(\epsilon)$, $k = 0, \ldots, \nu - 1$ satisfy the conclusions of Lemma 1.

Observe that in the algorithm below, the index h is decreased in each iteration. Thus when we determine the sets $\mathcal{F}_h$ and the chains $\mathcal{C}_{i,h}$, the sets $\mathcal{F}_k$ and $\mathcal{C}_{i,k}$ are already determined for $k = h + 1, \ldots, \nu$.

THE SCANBAS ALGORITHM

Set $h = \nu$.

Step 1. Scan $J^{(h-1)}(\epsilon)$ to extract a set

$$\mathcal{F}_h = \{u^{(h,1,h)}, \ldots, u^{(h,s_h,h)}\}$$

of null vectors of A, which is maximal with respect to the property that the union $\mathcal{G}_h$ of $\mathcal{F}_h$ and the sets $\mathcal{F}_k$, $k = h + 1, \ldots, \nu$ is linearly independent.

Step 2. Then for each $u^{(h,i,h)}$, $i = 1, \ldots, s_h$, select the chain

$$\mathcal{C}_{i,h} = \{u^{(j,i,h)} \mid j = 1, \ldots, h\},$$

which consists of the columns in $J^{(0)}(\epsilon), \ldots, J^{(h-1)}(\epsilon)$ corresponding to $u^{(h,i,h)}$, i.e., if $u^{(h,i,h)}$ is the rth column of $J^{(h-1)}(\epsilon)$, then $u^{(j,i,h)}$ is the rth column of $J^{(j-1)}(\epsilon)$, $j = 1, \ldots, h$.

If $h > 1$, reduce h by 1, and repeat.
If $h = 1$, then stop.

Remark. Note that $u^{(j,i,h)}$ is the vector of height $h - j + 1$ in the ith chain of length h.

THEOREM 1. *Let $\mathcal{C}$ be the union of the chains*

$$\mathcal{C}_{i,h} = \{u^{(j,i,h)} \mid j = 1, \ldots, h\}, \quad i = 1, \ldots, s_h, \quad h = 1, \ldots, \nu.$$

Then

(i) *$\mathcal{C}$ consists of nonnegative vectors.*
(ii) *$\mathcal{C}$ is a linearly independent set of vectors.*
(iii) *Let $1 \le h \le \nu$. Then $\mathcal{G}_h = \cup_{k=h}^{\nu} \mathcal{F}_k$ is a basis for S_h.*
(iv) *$\mathcal{C}$ contains exactly ξ_h vectors of height h, $h = 1, \ldots, \nu$.*
(v) *$\mathcal{C}$ can be extended to a height basis for $E(A)$.*

Proof. (i) Each vector in $\mathcal{C}$ appears in a column in some $J^{(h)}$, $h = 0, \ldots, \nu - 1$, and these matrices are nonnegative.

(ii) The set $\mathcal{G}_1$ defined above is the fundament of $\mathcal{C}$ and, by construction, $\mathcal{G}_1$ is linearly independent. Hence $\mathcal{C}$ is linearly independent, e.g., Bru and Neumann [3].

(iii) Let $1 \le h \le \nu$. First, let $x \in \mathcal{G}_h$. Then $x \in \mathcal{F}_k$ for some k, $h \le k \le \nu$. Hence x is a column of $J^{(k-1)}(\epsilon)$ and therefore $x = A^{k-1}y$, where y is a column of $J^{(0)}(\epsilon)$. Since $x \in N(A)$, it follows that y must be in F_k. Hence $x \in A^{k-1}E_k = S_k \subseteq S_h$.

Conversely, let $x \in S_h$. Then $x = A^{h-1}y$, where $y \in E_h$ and so, by Corollary 2, y is a linear combination of columns of $J^{(0)}(\epsilon)$ that lie in F_h. Hence x is a linear combination of columns of $J^{(h-1)}(\epsilon)$ that lie in $N(A)$. Since by the first part of the proof of (iii), $\mathcal{F}_k \subseteq S_h$, $k = h + 1, \ldots, \nu$, it now follows that the set $\mathcal{G}_h$ obtained in Step 2 of the SCANBAS algorithm is a basis for S_h.

(iv) Let $\mathcal{C}_h$ be the set of all vectors in $\mathcal{C}$ of height h. Then the map $x \to A^{h-1}x$ is a bijection of $\mathcal{C}_h$ onto $\mathcal{F}_h$, and hence (iv) follows from (iii).

(v) Since, by (iv), $A^{h-1}\mathcal{C}_h = \mathcal{F}_h$ and $\mathcal{F}_h$ is linearly independent, it follows that $\mathcal{C}_h$ is linearly independent mod E_{h-1}. Hence we can extend $\mathcal{C}_h$ to a set $\mathcal{B}_h$, which is a basis E_h mod E_{h-1}. It follows that $\mathcal{B} = \cup_{h=1}^{\nu} \mathcal{B}_h$ is a height basis for $E(A)$, cf. [8, Prop. 3.14]. □

COROLLARY 3. *It holds that*

$$\xi_h = s_h + \cdots + s_\nu, \quad h = 1, \ldots, \nu.$$

Let $x \in E(A)$ be a vector of height k. Then the *the chain derived from x* is defined to be the set $\{x, Ax, \ldots, A^{k-1}x\}$. The *chains derived from a subset* of $E(A)$ is the union of all chains derived from the vectors in this set. The technique used to prove the following important corollary is related to the proof of Hershkowitz and Schneider [8, Prop. (6.1)].

COROLLARY 4. *Let $1 \le t \le \nu$ and let $\eta_k = \xi_k$, $k = t, \ldots, \nu$. Then the chains $\mathcal{C}_{i,h}$, where $1 \le i \le s_h$ and $t - 1 \le h \le \nu$, can be embedded (extended) to a Jordan basis for $E(A)$.*

Proof. For $k = t, \ldots, \nu$, let $\mathcal{H}_k$ consist of all vectors $u^{(1,i,k)} \in \mathcal{C}$, $i = 1, \ldots, s_k$. Since $\mathrm{ht}(u^{(1,i,k)}) = k$ and $\xi_k = \eta_k$, it follows that $\cup_{r=k}^{\nu} A^{r-k}\mathcal{H}_r$ is a basis for $N(A^k)$ mod $(N(A^{k-1}))$. Now let $k = t - 1$. Then the set of vectors $u^{(1,i,k)}$, $i = 1, \ldots, s_k$, can be completed to a set $\mathcal{H}_k$ such that $\cup_{r=k}^{\nu} A^{r-k}\mathcal{H}_r$ is a basis for $N(A^k)$ mod $(N(A^{k-1}))$. Furthermore, for $k = 1, \ldots, t-2$, there exist sets $\mathcal{H}_k$ such that $\cup_{r=k}^{\nu} A^{r-k}\mathcal{H}_r$ is a basis

for $N(A^k)$ mod $N(A^{k-1})$. The chains derived from $\cup_{k=1}^{\nu}\mathcal{H}_k$ now form a Jordan basis for $E(A)$ with the required properties. □

Remark. In [7, Thm. (6.6)] it is shown that there exists a Jordan basis for $E(A)$ such that all chains of length greater than or equal to t are nonnegative if and only if $\xi_k = \eta_k$, $k = t, \ldots, \nu$. Thus, if a Jordan basis exists such that all chains of length t or greater are nonnegative, then the SCANBAS algorithm will produce such chains. In particular, if a nonnegative Jordan basis for $E(A)$ exists (viz., $\xi_k = \eta_k$, $k = 2, \ldots, \nu$ or see [9, Thm. 6.6] for many other equivalent conditions), then the SCANBAS algorithm produces a nonnegative Jordan basis for $E(A)$. Finally, since we always have $\xi_\nu = \eta_\nu$, cf. [8, Prop. (4.2)], the SCANBAS algorithm always produces a set of nonnegative chains of length ν which can be extended to a Jordan basis $E(A)$ by adding chains of length at most $\nu - 1$. The result that there is a Jordan basis for $E(A)$ such that all chains of length ν are nonnegative is known; see [8, Cor. (6.12)] for the existence of such chains.

6. Examples and concluding remarks. We call a set $\mathcal{C}$ of vectors a maximal nonnegative union of chains (MNUC) provided $\mathcal{C}$ is a union of nonnegative chains, $\mathcal{C}$ is linearly independent, and $\mathcal{C}$ contains ξ_h vectors of height h. By Theorem 1, the SCANBAS alogorithm produces an MNUC. In this section we give several examples of MNUCs for various matrices and the relation of these MNUCs to Jordan bases.

We call a diagram of pluses with ξ_h pluses in row h (counting from the bottom) the *Peak diagram* of the matrix. Similarly we call a diagram of stars with η_h pluses in row h (counting from the bottom) the *Jordan diagram* of the matrix. (As is very well known, the number of stars in each column, read from the left, yields the Jordan (Segre) characteristic of the matrix.)

Example 1. We begin with an example where the MNUC consists of complete Jordan chains and may be completed to a Jordan basis by adjoining an eigenvector.

Let

$$a = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 2 & 1 & 2 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

We put $sca = \text{scanbas}(a)$. Then

$$sca = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 2 & 3 & 3 & 1 & 1 \\ 4 & 6 & 6 & 3 & 2 \end{pmatrix}.$$

Then

$$jna = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 2 & 3 & 3 & 1 & 1 & 0 \\ 4 & 6 & 6 & 3 & 2 & 0 \end{pmatrix}$$

is a Jordan basis since

$$a \times jna = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 1 & 0 \\ 0 & 4 & 6 & 0 & 3 & 0 \end{pmatrix}.$$

We observe that the Jordan and Peak diagrams can be combined as

$$\begin{matrix} + & & & & \\ + & + & & & \\ + & + & * & . \end{matrix}$$

Example 2. We now give an example of a minus M-matrix whose Perron eigenspace has a nonnegative Jordan basis and the basis with such specifications produced by our SCANBAS algorithm. Let

$$b = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 0 \end{pmatrix}.$$

Here the SCANBAS algorithm yields the MNUC $scb = \text{scanbas}(b)$ given by

$$scb = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 2 & 0 & 1 & 1 \\ 3 & 5 & 5 & 2 & 3 & 3 \\ 4 & 5 & 5 & 3 & 4 & 4 \end{pmatrix}.$$

This is easily seen to be a nonnegative Jordan basis consisting of two chains each of length 3.

Example 3. We give an example of a matrix that possesses an MNUC that can be embedded in a Jordan basis, but where no MNUC can consist of complete Jordan chains.

Let

$$c = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Then the index of c is 3 and, if we choose $\epsilon = 1$, we obtain the transform components

$$j0c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 4 & 3 & 1 & 1 & 1 & 0 \\ 3 & 3 & 1 & 1 & 0 & 1 \end{pmatrix},$$

$$j1c = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 4 & 3 & 1 & 1 & 0 & 0 \\ 3 & 3 & 1 & 1 & 0 & 0 \end{pmatrix},$$

and

$$j2c = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

As we can see by inspection of the transform components, our SCANBAS algorithm yields $scc = \text{scanbas}(c)$ given by

$$scc = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 2 & 4 & 4 & 1 \\ 2 & 3 & 3 & 0 \end{pmatrix}.$$

This set may be extended to a Jordan basis

$$jnc = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & -1 \\ 2 & 4 & 4 & 1 & 0 & 0 \\ 2 & 3 & 3 & 0 & 0 & 0 \end{pmatrix},$$

where

$$c \times jnc = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 0 \end{pmatrix}.$$

Thus the combined Peak and Jordan diagrams here are

$$\begin{array}{cccc} + & & & \\ + & * & & \\ + & + & * & . \end{array}$$

We label the columns of jnc (from left to right) by v^{11}, v^{12}, v^{13}, v^{21}, v^{22}, v^{31}. Then we have the Jordan chains (v^{13}, v^{12}, v^{11}), (v^{22}, v^{21}) and (v^{31}). If some MNUC can be extended to a Jordan basis, then we would also get a combined Peak and Jordan diagram for c of the form

$$\begin{array}{cccc} + & & & \\ + & * & & \\ + & * & + & . \end{array}$$

We shall show that this is impossible; for let (w^{13}, w^{12}, w^{13}), (w^{22}, w^{21}), and (w^{31}) be another Jordan basis. Note that w^{31} is a linear combination of v^{11}, v^{21}, and v^{31} with nonzero coefficients for v^{31}, since w^{31} does not belong to range(c), see Bru, Rodman, and Schneider [4] for arguments of this type. But then, by inspection of the vectors, w^{31} cannot be nonnegative.

Example 4. We give an example of a matrix for which it is impossible to embed the chains of any MNUC into a Jordan basis.

Let

$$d = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Then the SCANBAS algorithm yields the MNUC $scd = \text{scanbas}(d)$

$$scd = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 4 & 4 & 4 & 1 & 1 \\ 2 & 3 & 3 & 3 & 0 & 0 \end{pmatrix}.$$

A Jordan basis for d is given by

$$jnd = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 2 & 4 & 4 & 4 & 1 & 0 & 0 & 0 & 0 \\ 2 & 3 & 3 & 3 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Thus the peak diagram for d is

```
+
+
+   +
+   +
```

and the Jordan diagram is

```
*
*   *
*   *
*   *   *   *.
```

We claim that no Jordan basis for d is an extension of an MNUC. We label the columns of *jnd* (from left to right) as x^{11}, x^{12}, x^{13}, x^{14}, x^{21}, x^{22}, x^{23}, x^{31}, x^{41}.

Suppose that there is a Jordan basis whose elements of height 3 are w^{13} and w^{23}, where w^{13} is of form $d(w^{14})$. Then, w^{13} is a multiple of x^{13}, while w^{23} is a linear combination of x^{13} and x^{23}, where x^{23} must have a nonzero coefficient. Hence w^{23} is not nonnegative. But if the Jordan basis is an extension of an MNUC, w^{23} must be nonnegative. Our claim follows.

Finally, we outline how our SCANBAS algorithm is implemented using MATLAB. The entire process is controlled by a function called *scanbas.m* whose input is the minus M-matrix A and whose output is an MNUC. This function first calls another MATLAB function *nnb.m* that returns an $\epsilon > 0$ and $J^{(0)} \geq 0$. The value value of $\epsilon > 0$, which is returned, is also sufficient to ensure that all higher-order transform components of A are nonnegative. To achieve its purpose, *nnb.m* initially determines the eigenprojection $Z^{(0)}$ by calling on a function *drazin.m*. The original version of *drazin.m* was written by Professor Robert E. Hartwig of North Carolina State University. This function computes the eigenprojection via the evaluation of the Drazin inverse A^D, viz., $Z^{(0)} = I - AA^D$, which is carried out using an algorithm due to Hartwig [5]. (For other methods of computing the Drazin inverse of a matrix, see the *shuffle* algorithm due to Anstreicher and Rothblum [1].) We mention that in *drazin.m*, the reduction steps used to implement Hartwig's algorithm are executed using the [q,r]=qr(·) command of MATLAB, not only for accuracy, but for the convenience of having the reducing matrices that this method needs from step to step [5]. The function *drazin.m* also returns ν, the index of A at 0. With $Z^{(0)}$ and ν at hand, *nnb.m* calls the function *macse.m*, which computes an $\epsilon > 0$ such that all transform components are nonnegative. This is done by generating iteratively all the principal

components of A. With all this data at hand, *nnb.m* finally computes $J^{(0)}$ and returns the control to *scanbas.m* which now proceeds to compute an MNUC according to Steps 1 and 2 of the SCANBAS algorithm given in §5. This segment of *scanbas.m* starts by setting up an array W that contains, juxtaposed, all, say up to a multiple, transform components generated iteratively from $J^{(0)}$. Steps 1 and 2 are now carried out using a nested for/if loops.

REFERENCES

[1] K. M. ANSTREICHER AND U. G. ROTHBLUM, *Using Gauss–Jordan elimination to compute the index, generalized nullspace, and Drazin inverses*, Linear Algebra Appl., 85 (1987), p. 239.

[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[3] R. BRU AND M. NEUMANN, *Nonnegative Jordan bases*, Linear Multilinear Algebra, 23 (1988), pp. 95–109.

[4] R. BRU, L. RODMAN, AND H. SCHNEIDER, *Extensions of Jordan bases for invariant subspaces of a matrix*, Linear Algebra Appl., 150 (1991), pp. 209–225.

[5] R. E. HARTWIG, *A method for computing A^d*, Math. Japon., 26 (1981), pp. 37–43.

[6] R. E. HARTWIG, M. NEUMANN, AND N. J. ROSE, *An algebraic-analytic approach to nonnegative basis*, Linear Algebra Appl., 133 (1990), pp. 77–88.

[7] D. HERSHKOWITZ, *Peak characteristic and nonnegative signature*, Linear Algebra Appl., 147 (1991), pp. 55–73.

[8] D. HERSHKOWITZ AND H. SCHNEIDER, *Height bases, level bases, and the equality of the height and level characteristics of an M-matrix*, Linear Multilinear Algebra, 25 (1989), pp. 149–147.

[9] ———*Combinatorial bases, derived Jordan sets, and the equality of the height and level characteristics of an M-matrix*, Linear Multilinear Algebra, 29 (1991), pp. 21–42.

[10] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, 2nd ed., Academic Press, New York, 1985.

[11] M. NEUMANN AND H. SCHNEIDER, *Principal components of minus* M*-matrices*, Linear Multilinear Algebra, 32 (1992), pp. 131–148.

[12] D. RICHMAN AND H. SCHNEIDER, *On the singular graph and Weyr characteristic of an* M*-matrix*, Aequations Math., 17 (1978), pp. 208–234

[13] U. G. ROTHBLUM, *Algebraic eigenspaces for nonnegative matrices*, Linear Algebra Appl., 12 (1975), pp. 281–292.

[14] H. SCHNEIDER, *The influence of the marked reduced graph of a nonnegative matrix on the Jordan form and related properties: A survey*, Linear Algebra Appl., 84 (1986), pp. 161–189.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 592-622, April 1994

016

ON RANK-REVEALING FACTORISATIONS*

SHIVKUMAR CHANDRASEKARAN† AND ILSE C. F. IPSEN†

Abstract. The problem of finding a rank-revealing QR (RRQR) factorisation of a matrix A consists of permuting the columns of A such that the resulting QR factorisation contains an upper triangular matrix whose linearly dependent columns are separated from the linearly independent ones. In this paper a systematic treatment of algorithms for determining RRQR factorisations is presented.

In particular, the authors start by presenting precise mathematical formulations for the problem of determining a RRQR factorisation, all of them optimisation problems. Then a hierarchy of "greedy" algorithms is derived to solve these optimisation problems, and it is shown that the existing RRQR algorithms correspond to particular greedy algorithms in this hierarchy. Matrices on which the greedy algorithms, and therefore the existing RRQR algorithms, can fail arbitrarily badly are presented.

Finally, motivated by an insight from the behaviour of the greedy algorithms, the authors present "hybrid" algorithms that solve the optimisation problems almost exactly (up to a factor proportional to the size of the matrix). Applying the hybrid algorithms as a follow-up to the conventional greedy algorithms may prove to be useful in practice.

Key words. condition estimation, pivoting, orthogonal factorisation, numerical rank, singular values

AMS subject classifications. 65F20, 65F25, 65F35, 15A42

1. Introduction.

The problem of finding a rank-revealing QR (RRQR) factorisation of a matrix A consists of permuting the columns of A such that the resulting QR factorisation contains an upper triangular matrix whose linearly dependent columns are separated from the linearly independent ones. RRQR factorisations are useful in problems such as subset selection and linear dependence analysis [21], [29], [37], [39], subspace tracking [6], [14], and rank determination [9]. Further applications are given in [12] and [17].

To determine a RRQR factorisation one could just adopt the brute force approach and inspect all possible column permutations until one has found a factorisation to one's liking. The operation count, of course, is guaranteed to be combinatorial. Consequently, much effort has gone in designing RRQR algorithms whose operation count is polynomial in the size of the matrix.

The first such algorithm, the QR factorisation with column pivoting [7], [16], [19], was developed by Golub in 1965 and by Faddeev, Kublanovskaya, and Faddeeva in 1966. It makes use of column permutations and orthogonal rotations to maintain the triangular structure of the matrix. About ten years later a second algorithm was published by Golub, Klema, and Stewart [20], based on applying the first algorithm to certain singular vectors of the matrix. At about the same time, a third algorithm appeared in a paper by Gragg and Stewart [22] that works on the inverse of the matrix. These three algorithms constitute the basis for all known RRQR algorithms.

Yet, it took another ten years for the next batch of algorithms by Stewart [32], Foster [17], and Chan [9] to appear. By this time it was known that there are matrices

* Received by the editors December 23, 1991; accepted for publication (in revised form) October 26, 1992. The work presented in this paper was supported by National Science Foundation grant CCR-9102853.

† Department of Computer Science, Yale University, New Haven, Connecticut 06520 (ipsen@math.ncsu.edu, chandras@math.ncsu.edu).

for which Golub's RRQR algorithm [7], [16], [19] can fail arbitrarily badly; Kahan's matrix [28] is such an example.

Again the field lay fallow for several years. Recently Hong and Pan [26] proved that an optimal RRQR factorisation is able to produce an estimate of a singular value that is accurate up to a factor proportional to the matrix size. This result implies that, in exact arithmetic and with a combinatorial operation count, RRQR factorisations have the potential of being accurate and reliable. (Much more than that, though, the result represents a statement about the relation between matrix columns and singular values: it says that there are k columns in the matrix that can reproduce, up to a factor in the matrix size, the kth singular value of the matrix.)

These days, the potential of RRQR factorisations is investigated for use in truncated singular value decompositions [10], [23], Lanczos methods [14], total least squares [37], and sparse matrix computations [3]–[5], [30]. Stewart has extended the RRQR factorisation by allowing orthogonal rotations from the right, resulting in the so-called URV decomposition [1], [31], [35], [36].

The state of affairs regarding RRQR factorisations can be summed up as follows. Despite the variety of algorithms, the problem of what it means to find a RRQR decomposition has never been clearly defined. Most definitions of a RRQR factorisation are about as fuzzy as the one we gave in the first sentence of this paper. Relationships or connections among the different RRQR algorithms are not known. All algorithms have the potential of failing badly. For some, we know the matrices where they fail badly. No criteria, other than a few test matrices, are known for comparing algorithms and judging their quality. Surprisingly, in numerical experiments, most RRQR algorithms turn out to be accurate and fast.

In this paper we present a systematic treatment of algorithms for determining RRQR factorisations. We start by presenting three precise mathematical formulations for the problem of determining a RRQR factorisation: one is a maximisation problem, one is a minimisation problem, and a third one is a combination of the two. We derive a hierarchy of "greedy" algorithms to solve the maximisation problem. It turns out that algorithms for solving the minimisation problem can be obtained by running algorithms for the maximisation problem on the inverse of the matrix and vice versa. This gives two parallel hierarchies of greedy algorithms for determining RRQR factorisations. We show that the existing RRQR algorithms correspond to particular greedy algorithms in this hierarchy. Moreover, we present matrices on which the greedy algorithms, and therefore the existing RRQR algorithms, fail arbitrarily badly.

Finally, motivated by our insight from the behaviour of the greedy algorithms, we present three "hybrid" algorithms that solve the optimisation problems with an accuracy given by the bounds of Hong and Pan [26]. Although the worst-case operation count of the hybrid algorithms may be combinatorial, we have not been able to find a matrix where this occurs. We present a few numerical experiments to demonstrate that applying the hybrid algorithms as a follow-up to the conventional RRQR algorithms may prove to be useful in practice.

2. The problem. In this section we give mathematical formulations of the problem of determining a rank-revealing QR (RRQR) factorisation of a matrix M.

Let M be a real $m \times n$ matrix and $m \geq n$. We assume that the singular values $\sigma_i(M)$ of M are arranged in decreasing order

$$\sigma_1(M) \geq \cdots \geq \sigma_n(M).$$

We also assume that k is a given integer such that $1 \leq k < n$ and $\sigma_k(M) > 0$. In the

applications where rank-revealing factorisations are of relevance, $\sigma_k(M)$ and $\sigma_{k+1}(M)$ are usually "well-separated," and $\sigma_{k+1}(M)$ is "small," of the order of the error in the computation, which means that the matrix has numerical rank k. Although our algorithms do not use this, it is useful to keep it in mind.

Denote by

$$M\Pi = QR$$

the QR factorisation of M with its columns permuted according to the $n \times n$ permutation matrix Π. The real $m \times n$ matrix Q has orthonormal columns, and the real $n \times n$ matrix R is upper triangular with positive diagonal elements. We block-partition R as

$$\begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \end{array} = R,$$

where R_{11} is a $k \times k$ matrix.

The RRQR problem. The problems to be discussed in this paper are how to choose permutations Π such that

$$\sigma_{\min}(R_{11}) \approx \sigma_k(M)$$

or

$$\sigma_{\max}(R_{22}) \approx \sigma_{k+1}(M)$$

or both hold simultaneously. So there are three objectives leading to three different problems, all of which we refer to as "rank-revealing problems." It is an open question whether these are really three *different* objectives. That is, if we find a permutation such that $\sigma_{\min}(R_{11}) \approx \sigma_k(M)$, does it imply that $\sigma_{\max}(R_{22}) \approx \sigma_{k+1}(M)$? Our attempts at answering this question have not yielded sufficiently good answers, and in this paper we will consider them as three independent objectives.

Satisfaction of the third objective, where both bounds are satisfied simultaneously, implies that the leading k columns of $M\Pi$ have condition number $\sigma_1(M)/\sigma_k(M)$ and approximate the range space of M to an "accuracy" of $\sigma_{k+1}(M)$.

Before proceeding any further we should be more specific about those $\approx$ signs. According to the interlacing properties of singular values (Corollary 8.3.3 in [21] applied to R^T) the bounds

$$(I1) \qquad \sigma_{\min}(R_{11}) \leq \sigma_k(M),$$
$$(I2) \qquad \sigma_{\max}(R_{22}) \geq \sigma_{k+1}(M)$$

hold for *any* permutation Π. So the RRQR problems can be precisely formulated as

Problem-I: $\max_\Pi \sigma_{\min}(R_{11})$;
Problem-II: $\min_\Pi \sigma_{\max}(R_{22})$

or that both can be solved simultaneously, though that may not be possible all the time.

Because we believe that the time complexity of these problems is combinatorial, we are content to find permutations Π that guarantee

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{p(n)}$$

or

$$\sigma_{\max}(R_{22}) \leq q(n)\sigma_{k+1}(M)$$

or that both bounds hold simultaneously. Here $p(n)$ and $q(n)$ are low degree polynomials in n. We say that a permutation Π that achieves one or both of these inequalities gives rise to a RRQR factorisation $M\Pi = QR$. An algorithm that attempts to solve Problem-I is called a Type-I algorithm and has the suffix I in its name. An algorithm that attempts to solve Problem-II is called a Type-II algorithm and has the suffix II in its name.

3. Overview of RRQR algorithms. We accomplish two tasks in this paper: first, we demonstrate that all existing RRQR algorithms form a hierarchy of greedy algorithms; and second, we present a set of new algorithms that are more accurate than the existing RRQR algorithms.

The existing algorithms in the literature guarantee that

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{n2^k} \qquad \text{or} \qquad \sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)n2^{n-k},$$

where the bounds are worst-case bounds. In practice, however, the existing algorithms perform quite well and the worst-case bounds are rarely obtained. There also exists an algorithm [20] with simultaneous worst-case bounds

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{n2^{\min(k,n-k)}}, \qquad \sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)n2^{\min(k,n-k)}.$$

In contrast, our new algorithms guarantee

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{k(n-k+1)}}$$

or

$$\sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)\sqrt{(k+1)(n-k)},$$

or both. The *existence* of such RRQR factorisations was established in [26]. Although we believe that the operation count of our new algorithms is combinatorial in the worst case, preliminary numerical experiments indicate that they may be fast in practice.

We ignore brute force algorithms for finding permutations Π because they do not exploit any properties of the matrix. Their operation count is therefore *always* combinatorial.

Now we start with the presentation of a unified approach to the existing RRQR algorithms. Our approach simplifies the presentation and analysis of these algorithms, and it also directly motivates our new algorithms. To this end, we make the following simplification. If $M\Pi = QR$ is a QR factorisation of M for some permutation Π, and if $R\bar{\Pi} = \bar{Q}\bar{R}$ is a RRQR factorisation of R, then

$$M\Pi\bar{\Pi} = Q\bar{Q}\bar{R}$$

is a RRQR factorisation of M. Hence one can ignore the original matrix M and work with the triangular matrix R instead.

4. Type-I greedy algorithms. It is our goal to find algorithms to solve Problem-I

$$\max_{\Pi} \sigma_{\min}(R_{11})$$

that guarantee

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{p(n)},$$

where $p(n)$ is a low degree polynomial in n. Problem-I is likely to represent a combinatorial optimisation problem, and this suggests that a greedy algorithm might do well.

The basic idea for our greedy algorithm, which we call Greedy-I, is very simple. The objective of Problem-I is to find k well-conditioned columns of M. So suppose that we already have $l < k$ well-conditioned columns of M. Then Greedy-I picks a column from the remaining $n - l$ columns of M such that the smallest singular value of the given l columns plus the new column is as large as possible. Starting with $l = 0$ this is done k times to pick k well-conditioned columns of M. Note that Greedy-I does not discard a column once it has been chosen.

ALGORITHM GREEDY-I
$R^{(0)} = R$
For $l = 0$ **to** $k - 1$ **do**
Set

$$\begin{array}{c} \\ l \\ n-l \end{array} \begin{array}{c} \begin{array}{cc} l & n-l \end{array} \\ \begin{pmatrix} A & B \\ & C \end{pmatrix} \end{array} = R^{(l)}.$$

Denote the columns of B and C by $b_i = Be_i$ and $c_i = Ce_i$.

1. Find the next column $l + j$ of $R^{(l)}$ such that

$$\max_{1 \leq i \leq n-l} \sigma_{\min} \begin{pmatrix} A & b_i \\ & c_i \end{pmatrix} = \sigma_{\min} \begin{pmatrix} A & b_j \\ & c_j \end{pmatrix}.$$

2. Exchange columns $l + 1$ and $l + j$ of $R^{(l)}$, and retriangularise it from the left with orthogonal transformations to get $R^{(l+1)}$.

In iteration $l = 0$, Greedy-I selects the column of R with largest norm. If everything goes right, then $R^{(k)}$ should be a rank-revealed upper triangular matrix. It is important to keep in mind that the dimensions of A, B, and C change with every iteration of Greedy-I.

Step 1 of Greedy-I, which selects the next column to be added to A, is very expensive. We make it cheaper, while at the same time retaining the greedy strategy, by performing step 1 only approximately. Thus the algorithm becomes less greedy and more efficient. Since Greedy-I can only find an approximate solution at best, further approximations will hopefully not make matters much worse.

Before continuing we make a small simplification. If $\gamma_i = \|c_i\|$, where $\|\cdot\|$ represents the two-norm, then

$$\sigma_{\min} \begin{pmatrix} A & b_i \\ 0 & c_i \end{pmatrix} = \sigma_{\min} \begin{pmatrix} A & b_i \\ 0 & \gamma_i \end{pmatrix}.$$

This means, only the two-norm of the columns of C matters rather than individual elements in a column. Therefore the problem amounts to determining the smallest singular values of an upper triangular matrix of order $l+1$.

Now we present a sequence of successively less greedy approximations to step 1 of Greedy-I that give rise to most of the existing RRQR algorithms. In other words, we show that most existing RRQR algorithms can be viewed as approximations to algorithm Greedy-I.

In the first approximation, the determination of the smallest singular values $\sigma_{\min}(\cdot)$ is replaced by directly computable quantities. We choose to approximate the smallest singular value of a matrix by the reciprocal of the largest two-norm of the rows of its inverse: if D is a nonsingular matrix of order n and

$$D^{-1} = \begin{pmatrix} r_1^T \\ r_2^T \\ \vdots \\ r_n^T \end{pmatrix},$$

then

$$\sigma_{\min}(D) \leq \min_{1 \leq i \leq n} \frac{1}{\|r_i\|} \leq \sigma_{\min}(D)\sqrt{n}.$$

Consequently, the smallest singular value of a nonsingular matrix of order n can be estimated up to a factor of $\sqrt{n}$.

ALGORITHM GREEDY-I.1
Replace step 1 in algorithm Greedy-I by:
Find the next column $l+j$ of $R^{(l)}$ such that

$$\max_{1 \leq i \leq n-l} \min_h \left\| e_h^T \begin{pmatrix} A & b_i \\ 0 & \gamma_i \end{pmatrix}^{-1} \right\|^{-1} = \min_h \left\| e_h^T \begin{pmatrix} A & b_j \\ 0 & \gamma_j \end{pmatrix}^{-1} \right\|^{-1},$$

where e_h^T is the hth row of the identity matrix of order $l+1$.

An algorithm similar to Greedy-I.1 was proposed by Stewart [34] where, for reasons of efficiency, the Frobenius norm rather than the two-norm is used.

Although we say that Greedy-I.1 is an approximation to Greedy-I, this does not necessarily imply that Greedy-I reveals the rank better than Greedy-I.1. It only means that Greedy-I.1 is less greedy than Greedy-I. In particular, if in iteration l Greedy-I and Greedy-I.1 have the same submatrix A, then the $\sigma_{\min}$ of the leading $l+1$ columns from Greedy-I is larger than or equal to the $\sigma_{\min}$ of the corresponding columns from Greedy-I.1. But there is no guarantee that in the subsequent iteration $l+1$ the $\sigma_{\min}$ of the leading $l+2$ columns of Greedy-I will be larger than or equal to the $\sigma_{\min}$ of the corresponding columns of Greedy-I.1. This is because the greedy algorithms are not allowed to change their minds and to throw out a column selected in a previous iteration, and the best local choice in one step does not necessarily lead to the global optimum.

Because

$$\begin{pmatrix} A & b_i \\ 0 & \gamma_i \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & -A^{-1}b_i\gamma_i^{-1} \\ 0 & \gamma_i^{-1} \end{pmatrix},$$

the upper left $l \times l$ block A^{-1} is already available from the previous step, and only the last column of the inverse needs to be computed for each i, which requires $n - l$ matrix vector multiplications. But carrying the inverse along with us at every stage is costly in terms of space and we first get rid of that.

If the greedy algorithms have not failed at the lth stage, the l leading columns must be "well conditioned." Hence A must be a well-conditioned matrix. Therefore $\sigma_{\min}(A)$ cannot be "small," which in turn implies that no row of A^{-1} can have a large two-norm. But if the addition of a new column, say the ith, produces a small singular value, then the two-norm of some row of the inverse of the corresponding matrix must be large. But since we assumed that no row of A^{-1} is large, this must mean that some component of the last column of the inverse

$$\begin{pmatrix} -A^{-1}b_i\gamma_i^{-1} \\ \gamma_i^{-1} \end{pmatrix}$$

must be large in magnitude. Thus the second approximation to step 1 of Greedy-I,

$$\left\| e_h^T \begin{pmatrix} A & b_i \\ 0 & \gamma_i \end{pmatrix}^{-1} \right\| \approx \left| \begin{pmatrix} -A^{-1}b_i\gamma_i^{-1} \\ \gamma_i^{-1} \end{pmatrix}_h \right|,$$

still avoids the selection of a very bad column.

ALGORITHM GREEDY-I.2
Replace step 1 in algorithm Greedy-I by:
Find the next column $l + j$ of $R^{(l)}$ such that

$$\max_{1 \le i \le n-l} \min_h \left| \begin{pmatrix} -A^{-1}b_i\gamma_i^{-1} \\ \gamma_i^{-1} \end{pmatrix}_h \right|^{-1} = \min_h \left| \begin{pmatrix} -A^{-1}b_j\gamma_j^{-1} \\ \gamma_j^{-1} \end{pmatrix}_h \right|^{-1}.$$

To eliminate the $n - l$ backsolves $A^{-1}b_i$ in Greedy-I.2, we make further use of the observation that A is probably well conditioned, so $\|A^{-1}b_i\| \approx 1$, and any large value must come from γ_i. Thus the third approximation to step 1 of Greedy-I,

$$\min_h \left| \begin{pmatrix} -A^{-1}b_i\gamma_i^{-1} \\ \gamma_i^{-1} \end{pmatrix}_h \right|^{-1} \approx \gamma_i,$$

still tries to avoid selecting a very bad column. This is nothing but the standard QR algorithm with column pivoting [7], [19], which is also described in [16].

ALGORITHM GREEDY-I.3 (GOLUB-I)
Replace step 1 in algorithm Greedy-I by:
Find the next column $l + j$ of $R^{(l)}$ such that $\max_{1 \le i \le n-l} \gamma_i = \gamma_j$.

This algorithm can be implemented efficiently because the column norms γ_i need only be updated during each iteration, rather than recomputed from scratch [7].

The approximations still to be discussed do not result in algorithms that are faster than Golub-I; in fact, they may be slower, but they are necessary to derive the remaining existing RRQR algorithms.

The goal is to make a further approximation to

$$\max_{1 \le i \le n-l} \gamma_i \equiv \alpha_{l+1}$$

in iteration l, where α_l is the lth diagonal element in the final upper triangular matrix in Golub-I. To this end, compute the right singular vector of the submatrix C of $R^{(l)}$ corresponding to its largest singular value $\|C\|$. Therefore the next approximation consists of finding the $(n-l) \times 1$ vector v such that

$$Cv = \|C\|u, \qquad \|v\| = \|u\| = 1,$$

and choosing as the next column the column j that corresponds to the largest component in magnitude of v,

$$|v_j| = \max_{1 \le i \le n-l} |v_i|.$$

ALGORITHM GREEDY-I.4 (CHAN-I)
Replace step 1 in algorithm Greedy-I by:
Find the next column $l+j$ of $R^{(l)}$ for which $|v_j| = \max_{1 \le i \le n-l} |v_i|$.

This algorithm was discovered independently by Chan and Hansen [11] and is related to the algorithm in [9]. Its choice of column j can be justified as follows. The Cauchy–Schwartz inequality gives

$$\gamma_j = \|Ce_j\| = \|u\| \, \|Ce_j\| \ge |u^T Ce_j| = \|C\| \, |v_j|.$$

As v has $n-l$ elements and satisfies $\|v\| = 1$, it must have a component v_j for which $|v_j| \ge 1/\sqrt{n-l}$. This is true in particular for the largest component in magnitude of v. Using this in $\gamma_j \ge \|C\| \, |v_j|$ gives

$$\frac{\alpha_{l+1}}{\sqrt{n-l}} \le \gamma_j \le \alpha_{l+1}, \qquad \alpha_{l+1} = \max_{1 \le i \le n-l} \gamma_i.$$

That is, the γ_j from algorithm Chan-I will be almost as large as that from algorithm Golub-I, if both algorithms were given the same l columns in A.

5. Threshold pivoting algorithms. We can make even further approximations to Chan-I. Algorithms Golub-I and Chan-I can be viewed as selecting large diagonal elements (*pivots*) at each stage to keep the smallest singular value as large as possible. According to the interlacing property (*I2*) of singular values, $\|C\| \ge \sigma_{l+1}(M)$, so

$$\alpha_{l+1} \ge \frac{\sigma_{l+1}(M)}{\sqrt{n-l}}, \quad 0 \le l \le k-1, \quad \text{where } \alpha_{l+1} = \max_{1 \le i \le n-l} \|Ce_i\|.$$

But all that is really needed is

$$\sigma_{\min}(R_{11}) \approx \sigma_k(M),$$

which means one may be able to get away with choosing pivots that are only as large as $\sigma_k(M)$. That is, instead of trying to achieve

$$|(R_{11})_{ll}| \approx \sigma_l(M), \qquad 1 \le l \le k,$$

we only try to ensure that

$$|(R_{11})_{ll}| \approx \sigma_k(M), \qquad 1 \le l \le k.$$

Golub-I and Chan-I try to keep all the pivots as large as possible at each stage. But since $\sigma_{\min}(R_{11})$ will be smaller than the smallest pivot, we are hoping that only the size of the smallest pivot is important, so that the conditions on the larger pivots can be relaxed.

Versions of Golub-I based on this approximation also go by the name of "threshold pivoting," and we now present two such algorithms. The first algorithm represents one of the first RRQR algorithms [20], [21] and, as we will show later, has the distinction of being able to solve both Problem-I and Problem-II simultaneously. Our name for the algorithm derives from the last names of its authors, Golub, Klema, and Stewart.

ALGORITHM GKS-I
Let $R = U\Sigma V^T$ be the singular value decomposition of R with

$$V = \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} V_1 & V_2 \end{pmatrix} \end{array}.$$

1. Compute V_1.
2. Apply algorithm Golub-I to the rows of V_1, $V_1^T\Pi = Q_v\bar{V}_1^T$.
3. Compute the QR decomposition $R\Pi = \bar{Q}\bar{R}$, which is the required rank-revealing factorisation.

To see that this is indeed a threshold pivoting algorithm, partition the singular value decomposition (SVD) of R as follows

$$R = U \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 & V_2 \end{pmatrix}^T.$$

Substituting the result of step 3, $\bar{Q}^T R\Pi = \bar{R}$, in step 2 gives

$$\bar{V}_1^T \bar{R}^{-1} = Q_v^T \Sigma_1^{-1} U^T \bar{Q}.$$

Since $\bar{R}^{-1}$ and the leading k columns of $\bar{V}_1^T$ represent upper triangular matrices,

$$\frac{\left|(\bar{V}_1)_{ii}\right|}{\left|(\bar{R})_{ii}\right|} \leq \|\bar{V}_1^T \bar{R}^{-1}\| = \|\Sigma_1^{-1}\| \leq \frac{1}{\sigma_k(M)}.$$

Because $\bar{V}_1^T$ is the result of QR with column pivoting on a matrix with orthonormal rows, the largest element in magnitude in the ith row of $\bar{V}_1^T$ is $(\bar{V}_1)_{ii}$ and $|(\bar{V}_1)_{ii}| \geq 1/\sqrt{n}$. Combining the inequalities gives a lower bound on the pivots,

$$\left|(\bar{R})_{ii}\right| \geq \frac{\sigma_k(M)}{\sqrt{n}}.$$

So algorithm GKS-I behaves like a threshold pivoting algorithm.

We now describe a threshold pivoting algorithm that we call Foster-I because it is related to an algorithm proposed by Foster (see Algorithm 2 in [17]). For a given δ, where δ is presumably about as big as $\sigma_k(M)$, Foster-I tries to achieve $\sigma_{\min}(R_{11}) \approx \delta$ by choosing pivots greater than or equal to δ. To this end it searches the rows of R, bottom up, for an element of magnitude greater than δ. When it finds such an element it adds the corresponding column to R_{11} and continues the search. As in all greedy algorithms for Problem-I, once a column has been added to R_{11} it is never discarded again. The algorithm halts when it has finished searching n rows. If it

succeeds in finding k elements larger than δ, then the first k pivots are at least as large as δ.

ALGORITHM FOSTER-I
$i = n$, *count* $= n$, $l = 0$
While (*count* ≥ 1) **do**
  Find the maximal element in row i: $|R_{ij}| = \max\{|R_{ii}|, \ldots, |R_{in}|\}$
  If ($|R_{ij}| \geq \delta$) **then**
    Insert column j between the lth and $(l+1)$st columns
    Retriangularise R
    $l = l + 1$
  else
    $i = i - 1$
  count $=$ *count* $- 1$

Here we have come to the end of our approximations to Greedy-I, which was a greedy algorithm for solving the Type-I problem

$$\max_{\Pi} \sigma_{\min}(R_{11}).$$

6. (Pessimistic) analysis of the greedy algorithms. In the previous sections we presented a succession of approximations to algorithm Greedy-I with little formal justification. Now we need to investigate how big $\sigma_k(M)/\sigma_{\min}(R_{11})$ from these algorithms can be. Algorithm Greedy-I represents the "best" method in the greedy sense, so we expect its worst-case behaviour to be indicative of that of the other greedy algorithms.

Suppose Greedy-I has already set aside l columns

$$R^{(l)} = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix},$$

where A is a $l \times l$ matrix. It then chooses as the $(l+1)$st column that column j which when added to A maximises the smallest singular value, so

$$\bar{\sigma}_{l+1} = \max_{1 \leq i \leq n-l} \sigma_{\min} \begin{pmatrix} A & b_i \\ 0 & c_i \end{pmatrix} = \sigma_{\min} \begin{pmatrix} A & b_j \\ 0 & c_j \end{pmatrix}.$$

To estimate how small $\bar{\sigma}_{l+1}$ can be we need to compute a lower bound on the smallest singular value. To this end we compute a lower bound instead for the column Golub-I would select, given the same A, because this also serves as a lower bound for the column Greedy-I picks. So assume that Golub-I picks column q. This column has the largest norm among all columns of C.

Just as in algorithm Greedy-I.1, we estimate $\sigma_{\min}$ by the reciprocal of the largest two-norm of the rows of the inverse

$$\begin{pmatrix} A & b_q \\ 0 & \gamma_q \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & -A^{-1} b_q \gamma_q^{-1} \\ 0 & \gamma_q^{-1} \end{pmatrix}.$$

The norm of the row with the largest norm among the leading l rows of the inverse is bounded from above by

$$\max_i \|e_i^T A^{-1}\| + \|A^{-1} b_q\| \gamma_q^{-1} \leq \frac{1}{\bar{\sigma}_l} + \frac{\|b_q\|}{\bar{\sigma}_l} \gamma_q^{-1} \leq \sqrt{2} \frac{\sqrt{\gamma_q^2 + \|b_q\|^2}}{\bar{\sigma}_l} \gamma_q^{-1} \leq \sqrt{2} \frac{\|M\|}{\bar{\sigma}_l} \gamma_q^{-1},$$

where the penultimate inequality is a result of the Cauchy–Schwartz inequality. The norm of the $(l+1)$st row of the inverse clearly cannot exceed the upper bound on the maximal norm of the leading l rows. Since the maximal row norm of the inverse approximates the smallest singular value of the matrix, we have

$$\bar{\sigma}_{l+1} \geq \frac{1}{\sqrt{2(l+1)}} \frac{\bar{\sigma}_l}{\sigma_1(M)} \gamma_q.$$

Using the interlacing property $(I2)$

$$\gamma_q = \max_{1 \leq i \leq n-l} \gamma_i \geq \frac{\sigma_{l+1}(M)}{\sqrt{n-l}}$$

with the above inequality gives a lower bound for the smallest singular value

$$\bar{\sigma}_{l+1} \geq \sigma_{l+1}(M) \frac{\bar{\sigma}_l}{\sigma_1(M)} \frac{1}{\sqrt{2(l+1)(n-l)}}.$$

This goes to show that even if the leading l columns had been selected so that $\bar{\sigma}_l$ was as accurate as possible, there could be a potentially serious deterioration in the quality of estimation from lth to $(l+1)$st singular value if the $(l+1)$st column is chosen according to a greedy strategy. This is because a greedy algorithm, once it has decided on a column, can never get rid of it. And a column that participates in an *accurate* estimation of $\bar{\sigma}_l$ may not be a column to be included in an accurate estimation of $\bar{\sigma}_{l+1}$. In particular, the estimate $\bar{\sigma}_{l+1}$ worsens with the ill conditioning of the leading l columns in $R^{(l)}$.

In fact, there exist matrices that almost achieve the above bound. One such example is the Kahan matrix [28]

$$K_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & s & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & s^{n-1} \end{pmatrix} \begin{pmatrix} 1 & -c & \cdots & -c \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -c \\ 0 & \cdots & 0 & 1 \end{pmatrix},$$

where $c^2 + s^2 = 1$. Greedy-I, Greedy-I.1, Greedy-I.2, and Golub-I do not cause any permutation of the columns of K_n. We prove this for Greedy-I by induction. Since all columns of K_n have unit norm, no column permutations are necessary in the first iteration of Greedy-I. Suppose no permutations are necessary during the first l iterations, so

$$K_n = \begin{pmatrix} K_l & b_1 & \cdots & b_{n-l} \\ & c_1 & \cdots & c_{n-l} \end{pmatrix}.$$

In the $(l+1)$st iteration Greedy-I selects the $(l+1)$st column by examining

$$\sigma_{\min} \begin{pmatrix} K_l & b_i \\ & c_i \end{pmatrix} = \sigma_{\min} \begin{pmatrix} K_l & b_i \\ & \gamma_i \end{pmatrix}, \quad \gamma_i = \|c_i\|, \quad 1 \leq i \leq n-l.$$

But all b_i are identical for the Kahan matrix, as are all γ_i; hence no permutations are necessary in iteration $l+1$.

Yet K_n is not in rank-revealed form. For $n = 100$, $k = 99$, and $c = 0.2$, the singular values are

$$\sigma_{100}(K_{100}) \approx 3 \times 10^{-9},$$
$$\sigma_{99}(K_{100}) \approx 0.1482,$$
$$\bar{\sigma}_{99} \approx 4 \times 10^{-9}.$$

Although the 99th and 100th singular values are well separated, the smallest singular value of the first 99 columns chosen by the greedy algorithms is exponentially smaller than $\sigma_{99}(K_{100})$.

Traditionally, the Kahan matrix has served as an example to demonstrate the failure of algorithm Golub-I to make the *last* diagonal element of the same order of magnitude as $\sigma_{100}(K_{100})$. But from our discussion it is clear that Golub-I pursues a different mission: it wants to make $\bar{\sigma}_{99} \approx \sigma_{99}(K_{100})$. And it fails in *that*.

7. (Optimistic) analysis of the greedy algorithms. Now that we have seen how badly the greedy algorithms do, we wonder why they do so well in practice? This question seems to be related to other rare matrix events like pivot growth in Gaussian elimination with partial pivoting. Foster [18] considers this question for QR without column pivoting. The case of QR with column pivoting seems to be much harder to analyse, and we can only give informal reasons why the greedy algorithms Golub-I, Chan-I, and GKS-I are so effective.

The basic idea is to derive a lower bound for $\sigma_{\min}(R_{11})$ of the form

$$\frac{\sigma_k(M)}{n\|W^{-1}\|} \leq \sigma_{\min}(R_{11}) \leq \sigma_k(M),$$

where W is a $k \times k$ triangular matrix with

$$|W| \leq 1, \qquad |W_{ii}| = 1,$$

and the inequality is componentwise. The lower triangular matrix in Gaussian elimination with partial pivoting satisfies these same two properties as the W matrices and is usually well conditioned. (Or as Kahan [28] would say, "intolerable pivot growth is a phenomenon that happens only to numerical analysts who are looking for that phenomenon.") Of course, this does not prove anything and more work is needed in this regard.

We start with the derivation of the above bound for algorithm Golub-I. Here we define the matrix W by

$$R_{11} = DW, \qquad D = \operatorname{diag}(R_{11}),$$

where $\operatorname{diag}(R_{11})$ is a diagonal matrix whose diagonal elements are the same as those of R_{11}. The diagonal elements of R_{11} in Golub-I satisfy $|(R_{11})_{ii}| \geq |(R_{11})_{ij}|$; hence W fulfills the required conditions

$$|W| \leq 1, \qquad |W_{ii}| = 1.$$

The interlacing properties $(I2)$ of singular values and the first few inequalities in § 5 imply that

$$|(R_{11})_{ii}| = \alpha_i \geq \frac{\sigma_i(M)}{\sqrt{n-i+1}}, \qquad 1 \leq i \leq k.$$

Since D is a diagonal matrix, its singular values equal its diagonal elements, so

$$\frac{1}{\sigma_{\min}(D)} = \|D^{-1}\| \leq \frac{\sqrt{n}}{\sigma_k(M)}.$$

From $\sigma_{\min}(R_{11}) \geq \sigma_{\min}(D)\sigma_{\min}(W)$ the desired bound for Golub-I follows

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{n}\|W^{-1}\|}.$$

Next we derive the bound for algorithm Chan-I. The proof is similar to that of Theorem 3.1 in [9]. We first define the $n \times k$ auxiliary matrix Z. Its columns are composed of the right singular vectors $v^{(l)}$ associated with the largest singular values of the lower right block of order $n-l+1$, $R_{22}^{(l)}$, of the final triangular matrix R. That is, Z is a lower trapezoidal matrix with columns

$$Ze_l = Z_l = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ v^{(l)} \end{pmatrix}, \qquad 1 \leq l \leq k,$$

where $R_{22}^{(l)} v^{(l)} = \|R_{22}^{(l)}\| u^{(l)}$ and $\|v^{(l)}\| = \|u^{(l)}\| = 1$. Then the lower triangular matrix W for Chan-I is given by

$$Z = \begin{pmatrix} W \\ * \end{pmatrix} D, \qquad D = \operatorname{diag}(Z_{11} \quad \ldots \quad Z_{kk}),$$

where Z_{ii} are the diagonal elements of Z.

According to algorithm Chan-I, the first component of $v^{(l)}$ is the largest in magnitude, hence

$$|W| \leq 1, \qquad |W_{ii}| = 1.$$

Moreover, $\|v^{(l)}\| = 1$ implies

$$|Z_{ll}| = |v_1^{(l)}| \geq \frac{1}{\sqrt{n}}, \qquad 1 \leq l \leq k,$$

and

$$\frac{1}{\|D^{-1}\|} = \sigma_{\min}(D) \geq \frac{1}{\sqrt{n}}.$$

From the interlacing property $(I2)$ of singular values, $\|R_{22}^{(l)}\| \geq \sigma_l(M)$, and the fact that $v^{(l)}$ is a right singular vector with

$$[v^{(l)}]^T [R_{22}^{(l)}]^{-1} = \frac{1}{\|R_{22}^{(l)}\|} [u^{(l)}]^T,$$

we get

$$\|Z_l^T R^{-1}\| = \|[v^{(l)}]^T [R_{22}^{(l)}]^{-1}\| \leq \frac{1}{\sigma_l(M)}.$$

Since Z^T and R are upper triangular this implies

$$\|DW^T R_{11}^{-1}\| \leq \|Z^T R^{-1}\| \leq \sqrt{k} \max_{1 \leq l \leq k} \|Z_l^T R^{-1}\| \leq \frac{\sqrt{k}}{\sigma_k(M)}.$$

Hence

$$\frac{\|R_{11}^{-1}\|}{\|D^{-1}\| \, \|W^{-1}\|} \leq \frac{\sqrt{k}}{\sigma_k(M)}$$

gives the desired bound for algorithm Chan-I

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{n\|W^{-1}\|}.$$

At last we derive the bound for algorithm GKS-I, which is also given in [20] and in Theorem 12.2.1 in [21]. Let $R = U\Sigma V^T$ be the SVD of R and partition

$$\begin{matrix} & k & n-k \\ V = & (V_1 & V_2\ ). \end{matrix}$$

Algorithm GKS applies algorithm Golub-I to V_1^T, so

$$V_1^T \Pi = Q_v \bar{V}_1^T,$$

where $\bar{V}_1$ is a lower trapezoidal matrix. The matrix W for GKS-I is defined by

$$\bar{V}_1 = \begin{pmatrix} W \\ * \end{pmatrix} D, \qquad D = \operatorname{diag}(\, \bar{V}_{11} \quad \dots \quad \bar{V}_{kk}\,),$$

where $\bar{V}_{ii}$ are diagonal elements of $\bar{V}_1$ and W is a lower triangular matrix. Because $\bar{V}_1$ comes from algorithm Golub-I, its diagonal elements are the largest elements in magnitude in each column, so $|\bar{V}_{ii}| \geq |\bar{V}_{ji}|$ and the matrix W satisfies the required properties

$$|W| \leq 1, \qquad |W_{ii}| = 1.$$

Since each column of $\bar{V}_1$ has unit norm, $|\bar{V}_{ii}| \geq 1/\sqrt{n}$, and since W^T and the final matrix R are upper triangular, one gets

$$\frac{\|R_{11}^{-1}\|}{\sqrt{n}\|W^{-1}\|} \leq \frac{\|R_{11}^{-1}\|}{\|W^{-T}\| \, \|D^{-1}\|} \leq \|DW^T R_{11}^{-1}\| \leq \|\bar{V}_1^T R^{-1}\|.$$

Moreover, from § 5 we know that (with $\bar{R}$ now renamed R)

$$\|\bar{V}_1^T R^{-1}\| = \|\Sigma_1^{-1}\| = \frac{1}{\sigma_k(M)}.$$

Combining the last two inequalities yields

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{n}\|W^{-1}\|}.$$

To summarise, we have demonstrated in this section that the failure of algorithms Golub-I, Chan-I, and GKS-I depends on $\|W^{-1}\|$, where W is a triangular matrix satisfying

$$|W| \leq 1, \qquad |W_{ii}| = 1.$$

The lower triangular matrix L in Gaussian elimination with partial pivoting satisfies the same properties as W, and it generally turns out that $\|L^{-1}\|$ is small, say, like $O(n)$. Although this does not prove anything, it does show that all these rare matrix events are closely related. The probability of pivot growth in Gaussian elimination with partial pivoting is closely related to the probabilities of Golub-I, Chan-I, and GKS-I failing.

For the above matrices W of order k a tight upper bound on $\|W^{-1}\|$ is [16], [28]

$$\|W^{-1}\| \leq \frac{1}{3}\sqrt{4^k + 6k - 1} \leq \sqrt{k}2^k, \qquad k > 1,$$

and, as illustrated in § 6, the Kahan matrix essentially achieves this bound.

8. Unification. After having discussed greedy algorithms for the solution of Problem-I

$$\max_{\Pi} \sigma_{\min}(R_{11}),$$

we now turn to greedy algorithms for Problem-II

$$\min_{\Pi} \sigma_{\max}(R_{22}).$$

Fortunately, a simple observation greatly reduces this task.

Section 3 explains why it suffices to solve Problem-I for triangular matrices $\bar{R}$ and to consider

$$\bar{R}\Pi = QR, \qquad R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}.$$

Suppose that $\bar{R}$ is nonsingular, invert both sides of the above equation,

$$\Pi^T \bar{R}^{-1} = \begin{pmatrix} R_{11}^{-1} & -R_{11}^{-1}R_{12}R_{22}^{-1} \\ 0 & R_{22}^{-1} \end{pmatrix} Q^T,$$

and take transposes on both sides

$$\bar{R}^{-T}\Pi = Q \begin{pmatrix} R_{11}^{-T} & 0 \\ -R_{22}^{-T}R_{12}^T R_{11}^{-T} & R_{22}^{-T} \end{pmatrix}.$$

Now Problem-II can be formulated as

$$\begin{aligned} \min_{\Pi} \sigma_{\max}(R_{22}) &= \min_{\Pi} \frac{1}{\sigma_{\min}(R_{22}^{-1})} \\ &= \frac{1}{\max_{\Pi} \sigma_{\min}(R_{22}^{-1})} \\ &= \frac{1}{\max_{\Pi} \sigma_{\min}(R_{22}^{-T})}. \end{aligned}$$

Hence solving Problem-II is equivalent to solving Problem-I for the inverse. We call this *the unification principle* as it lets us unify the algorithms and analyses of Problem-I and Problem-II.

Applying a Type-I algorithm to the inverse gives

$$\bar{R}^{-T}\bar{\Pi} = \bar{Q}\begin{pmatrix} P_{11} & P_{12} \\ 0 & P_{22} \end{pmatrix},$$

where P_{11} is an upper triangular matrix of order $n-k$, P_{22} is an upper triangular matrix of order k, and (hopefully) $\sigma_{\min}(P_{11}) \approx \sigma_{n-k}(\bar{R}^{-T})$. Hence we need to make some adjustments as P_{11} should correspond to R_{22}^{-T}, which is lower triangular. Moreover, P_{11} should really have been the lower right block.

The necessary adjustments are achieved by a sequence of permutations, which can be accumulated in Q and Π. First permute the two block columns and the two block rows,

$$\begin{pmatrix} P_{11} & P_{12} \\ 0 & P_{22} \end{pmatrix} \rightarrow \begin{pmatrix} P_{12} & P_{11} \\ P_{22} & 0 \end{pmatrix} \rightarrow \begin{pmatrix} P_{22} & 0 \\ P_{12} & P_{11} \end{pmatrix}.$$

Then reverse the ordering of the columns and of the rows in P_{11} and P_{22} separately. This is accomplished by means of permutation matrices J_p of order p that have ones on the antidiagonal,

$$\begin{pmatrix} P_{22} & 0 \\ P_{12} & P_{11} \end{pmatrix} \rightarrow \begin{pmatrix} J_k P_{22} J_k & 0 \\ J_{n-k} P_{12} J_k & J_{n-k} P_{11} J_{n-k} \end{pmatrix}.$$

Now the resulting matrix has the desired form; it is lower triangular with P_{11} in the lower right corner.

Therefore, the postprocessing step consisting of the above permutations proves that applying a Type-I algorithm to the rows of the inverse amounts to executing a Type-II algorithm. In fact, we call such an algorithm the *Type-II version* of the *Type-I algorithm*. This notion is completely symmetric with respect to the two types, as one can equally well construct a *Type-I version* of a *Type-II algorithm* to solve Problem-I.

Unification principle. Running a Type-I algorithm on the rows of the inverse of the matrix yields a Type-II algorithm.

9. Type-II greedy algorithms. In this section we illustrate the unification principle by exhibiting the Type-II version of algorithm Golub-I, and by proving that algorithm GKS-I also solves Problem-II.

We use the name Stewart-II for the Type-II version of algorithm Golub-I, as it

was first proposed in [33], though not quite in the form in which we are presenting it.

ALGORITHM STEWART-II
$R^{(0)} = R$
For $l = 0$ **to** $n - k - 1$ **do**
Set

$$\begin{array}{c} \\ n-l \\ l \end{array} \begin{array}{c} \begin{array}{cc} n-l & l \end{array} \\ \begin{pmatrix} A & B \\ & C \end{pmatrix} \end{array} = R^{(l)},$$

1. Find the next column j of $R^{(l)}$ such that

$$\max_{1 \le i \le n-l} \|e_i^T A^{-1}\| = \|e_j^T A^{-1}\|.$$

2. Exchange columns $n-l$ and j of $R^{(l)}$, and retriangularise it from the left with orthogonal transformations to get $R^{(l+1)}$.

Clearly, algorithm Stewart-II obtains the right ordering of the columns by sending the selected columns to the right end of the matrix. In all other matters it is completely equivalent to running Golub-I on the rows of the inverse.

A few clarifying remarks may be in order. Just because a Type-II version of an algorithm can be constructed by applying a Type-I algorithm to the rows of the inverse of the matrix, this does not mean that is also how it should be implemented. There may very well be a way to reformulate the Type-II version so that it avoids explicit dealings with inverses.

Furthermore, it is important to realise that a Type-I algorithm and its Type-II version, in general, come up with different column permutations; and that solving Problem-I does not entail solving Problem-II. All the unification principle says is that if there is an algorithm for solving Problem-I, then a simple modification will give an algorithm for solving Problem-II and vice versa.

There is another advantage of the unification principle. It allows us to carry over the analyses and worst-case examples for a Type-I algorithm, with suitable modifications, to its Type-II version and vice versa. A few examples follow.

In § 6 we explained that the lower bounds for the singular value estimates from algorithms Golub-I, Chan-I, and GKS-I can be cast in the form

$$\sigma_{\min}(R_{11}) \ge \frac{\sigma_k(M)}{n\|W^{-1}\|},$$

where W are triangular matrices satisfying

$$|W| \le 1, \qquad |W_{ii}| = 1.$$

The unification principle therefore admits upper bounds for the singular value estimates from the Type-II versions of Golub-I, Chan-I, and GKS-I of the form

$$\sigma_{\max}(R_{22}) \le \sigma_{k+1}(M) n \|W^{-1}\|,$$

where, again, W are triangular matrices satisfying

$$|W| \le 1, \qquad |W_{ii}| = 1.$$

As for other existing Type-II algorithms, the Type-II version of Chan-I, which we call Chan-II, was published apparently independently in [22], [9], [17]. The Type-II version of GKS-I was first published in [20] and will be called GKS-II. The Type-II version of Foster-I, which we refer to as Foster-II, was first published in [17]. The detailed exposition of Foster-II in [17] also serves to illuminate our algorithm Foster-I.

We still owe a justification of our claim that GKS-I also solves Problem-II [20], [21]. Let $R = U\Sigma V^T$ be the SVD of the final triangular matrix R, where

$$\begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \end{array} = V, \qquad \Sigma = \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix}.$$

This implies

$$\frac{1}{\sigma_{\min}(R_{11})\|V_{11}^{-1}\|} = \frac{\|R_{11}^{-1}\|}{\|V_{11}^{-1}\|} \leq \|V_{11}^T R_{11}^{-1}\| \leq \|\Sigma_1^{-1}\| = \frac{1}{\sigma_k(M)},$$

and

$$\frac{1}{\|R_{22}\|} = \sigma_{\min}(R_{22}^{-1}) \geq \sigma_{\min}(\Sigma_2^{-1}V_{22}) \geq \sigma_{\min}(\Sigma_2^{-1})\sigma_{\min}(V_{22}) = \frac{1}{\sigma_{k+1}(M)\|V_{22}^{-1}\|},$$

so

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\|V_{11}^{-1}\|}, \qquad \|R_{22}\| \leq \sigma_{k+1}(M)\|V_{22}^{-1}\|.$$

According to the CS decomposition, §2.6 in [21],

$$\|V_{11}^{-1}\| = \|V_{22}^{-1}\|.$$

Since GKS-I attempts to keep $\|V_{11}^{-1}\|$ small, it therefore automatically also tries to keep $\|V_{22}^{-1}\|$ small. Therefore GKS-I solves both, Problem-I and Problem-II.

At last we demonstrate how the worst-case example of a Type-I algorithm can be converted to a worst-case example for its Type-II version. Section 6 illustrates that the Kahan matrix

$$K_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & s & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & s^{n-1} \end{pmatrix} \begin{pmatrix} 1 & -c & \cdots & -c \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -c \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

represents a worst case for algorithms Greedy-I, Greedy-I.1, Greedy-I.2, and Golub-I. It follows from the unification principle that the modified Kahan matrix whose inverse is given by

$$\bar{K}_n^{-1} = \begin{pmatrix} 1 & -c & \cdots & -c \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -c \\ 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} s^{n-1} & 0 & \cdots & 0 \\ & s^{n-2} & \ddots & \vdots \\ & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix},$$

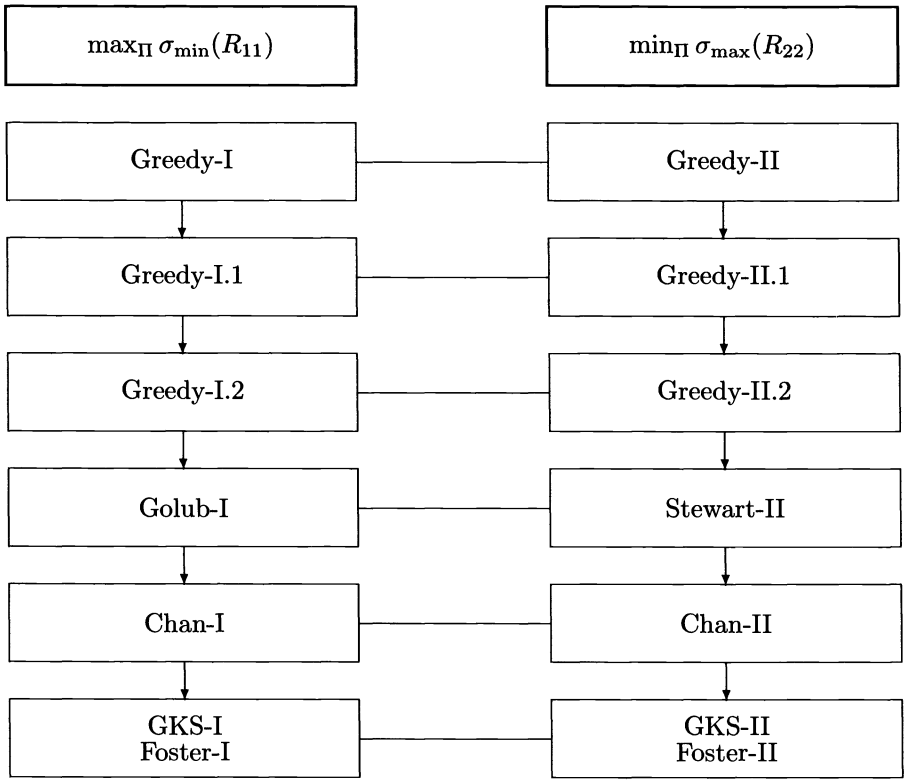


FIG. 10.1. *The greedy algorithms.*

where $c^2 + s^2 = 1$, represents a worst case for algorithms Greedy-II, Greedy-II.1, Greedy-II.2, and Stewart-II, the Type-II versions of the respective Type-I algorithms.

10. Summary. This ends our presentation of the existing RRQR algorithms. We gave three mathematical problems that we called rank-revealing problems,

Problem-I: $\max_\Pi \sigma_{\min}(R_{11})$,
Problem-II: $\min_\Pi \sigma_{\max}(R_{22})$,

and the third was to solve Problem-I and Problem-II simultaneously. We then exhibited a sequence of successively less greedy algorithms to solve Problem-I. By means of the unification principle, we demonstrated the existence of Type-II versions of these algorithms, which are also greedy but solve Problem-II instead. Figure 10.1 illustrates the two parallel hierarchies made up from the Type-I and Type-II algorithms, where the corresponding Type-I and Type-II algorithms are next to each other, and each algorithm is less greedy than the one above it. Each of the existing RRQR algorithms has a place in this hierarchy. Examples of exponential failure of these greedy algorithms are provided by the Kahan and modified Kahan matrices.

We have ignored the greedy algorithms based on condition number estimators for triangular matrices, e.g., [2]–[5], [25], [34], because their behaviour depends very much on the particular condition number estimator.

The worst-case bounds

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{n2^k}, \qquad \|R_{22}\| \leq \sigma_{k+1}(M)n2^{n-k}$$

reveal that Type-I greedy algorithms work pretty well for small k, while Type-II greedy algorithms work well when k is close to n. This prompts the question whether a Type-I and a Type-II greedy algorithm can be combined into a single algorithm that works all the time. The answer is given in the next section.

11. Overview of the hybrid algorithms. In this section we present algorithms Hybrid-I and Hybrid-II. They are guaranteed to solve Problem-I and Problem-II, respectively. We also present algorithm Hybrid-III. It is guaranteed to solve both Problem-I and Problem-II simultaneously.

In particular, Algorithm Hybrid-I guarantees that

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{k(n-k+1)}},$$
$$\sigma_{\max}(R_{22}) \leq \sigma_{\min}(R_{11})\sqrt{k(n-k+1)}.$$

Note that Hybrid-I does *not* solve Problem-II. According to the unification principle, the Type-II version of Hybrid-I, which we call Hybrid-II, must guarantee that

$$\sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)\sqrt{(k+1)(n-k)},$$
$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_{\max}(R_{22})}{\sqrt{(k+1)(n-k)}}.$$

Note again that Hybrid-II does not solve Problem-I. Hybrid-III does solve both Problem-I and Problem-II simultaneously, and it guarantees that

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{k(n-k+1)}},$$
$$\sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)\sqrt{(k+1)(n-k)}.$$

Of course, the brute force algorithm, which tries every combination of columns, also solves these problems, but its operation count is combinatorial. What about the hybrid algorithms? Unfortunately, we lack a complete analysis of the worst-case operation count of the hybrid algorithms, although we believe that it may be combinatorial as well. However, preliminary experimental results in §15 demonstrate that the hybrid algorithms are rather efficient in practice.

As in the previous sections we assume that k is given. Although this may not be a realistic assumption, a proper choice of k depends very much on the problem to be solved, and we refer to [20], [33] for the discussion of this issue.

12. Algorithm Hybrid-I. The algorithm Hybrid-I is a combination of Golub-I and Stewart-II, though in a practical implementation one may want to replace Stewart-II by Chan-II.

The obvious strategy of running Stewart-II after Golub-I is not guaranteed to solve Problem-I because Golub-I and Stewart-II almost always produce a unique ordering of columns, so the result of this strategy would merely equal the result of Stewart-II.

Instead, our idea is to alternate between Golub-I and Stewart-II and to let each work on a different part of the matrix: Stewart-II works on the (1,1) block of order k,

and Golub-I works on the (2,2) block of order $n-k+1$ of the matrix. Suppose Golub-I has picked the best column from the (2,2) block and put it in position k. Stewart-II then determines whether the kth column is indeed a good column. If not, it puts the worst column from the (1,1) block into position k. Now it is again Golub-I's turn to put the best column from the (2,2) block in position k. This process continues until Golub-I and Stewart-II agree on the kth column. To understand the resulting algorithm Hybrid-I, we briefly review Golub-I and Stewart-II.

Golub-I is good at approximating the largest singular value of $M\Pi = QR$. In its first iteration it finds the "most linearly independent" column of R, i.e., the column with largest norm. Suppose we permute this column to the first position and retriangularise the matrix. Then the first column $r_{11}e_1$ of the resulting triangular matrix approximates the largest singular value of M,

$$|r_{11}| \leq \sigma_{\max}(M) \leq \sqrt{n}|r_{11}|.$$

Since Stewart-II is the Type-II version of Golub-I, it is good at approximating the largest singular value of M^{-1} by finding the most linearly independent row of R^{-1}. Suppose we permute this row of R^{-1} to the last position and retriangularise the inverse to get the triangular matrix $\bar{R}^{-1} = \bar{\Pi}R^{-1}\bar{Q}$. Then the last column $r_{nn}^{-1}e_n$ of $\bar{R}^{-1}$ approximates the largest singular value of M^{-1},

$$|r_{nn}^{-1}| \leq \sigma_{\max}(M^{-1}) \leq \sqrt{n}|r_{nn}^{-1}|.$$

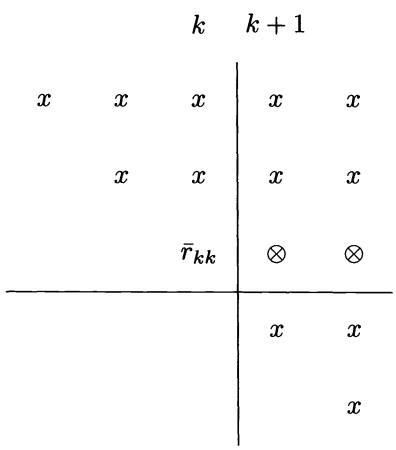
But since $\bar{R}^{-1}$ is triangular, r_{nn} is the trailing diagonal element of $\bar{R}$ and it approximates the smallest singular value of M,

$$\sigma_{\min}(M) \leq |r_{nn}| \leq \sqrt{n}\sigma_{\min}(M).$$

We illustrate Hybrid-I on a 5×5 example, where $k = 3$ and the symbol "x" represents nonzero matrix elements. First we run Golub-I on the $(2,2)$ block of order $n-k+1$ so that diagonal element r_{kk} has largest norm among all columns of the $(2,2)$ block

$$\begin{array}{cc|ccc} & & k-1 & k & \\ x & x & x & x & x \\ & x & x & x & x \\ \hline & & r_{kk} & x & x \\ & & & x & x \\ & & & & x \end{array}$$

Now we enlarge the $(1,1)$ block from order $k-1$ to order k so that the kth diagonal element can transfer information between the two algorithms. Then we run Stewart-II on the $(1,1)$ block of order k so that the (modified) diagonal element $\bar{r}_{kk}$ has smallest norm.



A run of Golub-I followed by Stewart-II constitutes one iteration. The circled elements in the $(1,2)$ block are modified by orthogonal rotations from the left due to retriangularisation in Stewart-II. They are part of the $(2,2)$ block for the subsequent run of Golub-I and illustrate how one algorithm changes the part of the matrix associated with the other algorithm. The $(1,1)$ block input to Stewart-II undergoes similar changes in column k due to column permutations during Golub-I.

ALGORITHM HYBRID-I(k)
$\bar{R}^{(0)} = R$, $l = 0$
Repeat
$l = l + 1$, *permuted* $= 0$
Set

$$\bar{R}^{(l)} = \begin{pmatrix} \bar{A} & \bar{B} \\ & \bar{C} \end{pmatrix},$$

where $\bar{A}$ is of order $k-1$ and $\bar{C}$ is of order $n-k+1$.
Golub-I:
1. Find the column $k+j-1$ of $\bar{R}^{(l)}$ such that $\|\bar{C}e_j\| = \max_{1\le i\le n-k+1} \|\bar{C}e_i\|$
2. **If** $\|\bar{C}e_1\| < \|\bar{C}e_j\|$ **then**
   permuted $= 1$
   Exchange columns k and $k+j-1$ of $\bar{R}^{(l)}$
   Retriangularise it from the left with orthogonal transformations to get

$$R^{(l)} = \begin{pmatrix} A & B \\ & C \end{pmatrix},$$

   where A is of order k and C is of order $n-k$.
Stewart-II:
3. Find the column j of $R^{(l)}$ such that $\|e_j^T A^{-1}\| = \max_{1\le i\le k} \|e_i^T A^{-1}\|$
4. **If** $\|e_k^T A^{-1}\| < \|e_j^T A^{-1}\|$ **then**
   permuted $= 1$
   Exchange columns j and k of $R^{(l)}$
   Retriangularise it from the left with orthogonal transformations to get $\bar{R}^{(l+1)}$.

until not *permuted*

The final matrix is

$$R = \begin{pmatrix} \bar{R}_{11} & \bar{R}_{12} \\ & \bar{R}_{22} \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix},$$

where $\bar{R}_{11}$ is of order $k-1$ and R_{11} is of order k.

The two if statements assure that permutations are performed only in case of a strict inequality but not in case of a tie.

We proceed with an analysis of Hybrid-I because it is not clear that Hybrid-I eventually halts, and that it indeed increases $\sigma_{\min}(R_{11})$. We first show that *if* Hybrid-I halts then

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{k(n-k+1)}},$$
$$\sigma_{\max}(R_{22}) \leq \sigma_{\min}(R_{11})\sqrt{k(n-k+1)}.$$

Suppose Hybrid-I halts. Then Golub-I applied to $\bar{R}_{22}$ does not change the first column $r_{kk}e_1$ of $\bar{R}_{22}$, where r_{kk} is the kth diagonal element of R. Hence

$$|r_{kk}| \geq \frac{\sigma_{\max}(\bar{R}_{22})}{\sqrt{n-k+1}} \geq \frac{\sigma_{\max}(R_{22})}{\sqrt{n-k+1}}$$

since R_{22} is a submatrix of $\bar{R}_{22}$. Moreover, Stewart-II applied to R_{11} does not change the last row $r_{kk}e_k^T$ of R_{11}, and

$$|r_{kk}| \leq \sigma_{\min}(R_{11})\sqrt{k}.$$

Combining the two inequalities for r_{kk} gives the first desired bound

$$\sigma_{\max}(R_{22}) \leq \sigma_{\min}(R_{11})\sqrt{k(n-k+1)}.$$

Applying the interlacing property $(I2)$ to $\bar{R}_{22}$,

$$|r_{kk}| \geq \frac{\sigma_{\max}(\bar{R}_{22})}{\sqrt{n-k+1}} \geq \frac{\sigma_k(M)}{\sqrt{n-k+1}},$$

and combining the previous two inequalities yields the second desired bound

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{k(n-k+1)}}.$$

Thus, if Hybrid-I halts, it solves Problem-I.

To prove that Hybrid-I indeed halts, we make use of the fact that columns are permuted only in case of strict inequalities. The basic idea is to show that $|\det(A)|$ is a strictly increasing function during the algorithm. Remember that A is the leading principal submatrix of order k. Since $|\det(A)|$ is unique for any given column ordering, no column ordering repeats if $|\det(A)|$ is strictly increasing. As there are only a finite number of column orderings, Hybrid-I must eventually halt.

It remains to show that $|\det(A)|$ is strictly increasing during Hybrid-I. By assumption from § 2 we have that $\sigma_k(M) > 0$. So we can assume that our initial ordering of columns is such that $|\det(A)| > 0$. Stewart-II does not change $\det(A)$ because $|\det(A)|$ is invariant under application of orthogonal transformations from

the left to $(A \quad B)$ and to C; and under permutation of the columns of A and of the columns of $\binom{B}{C}$. To see how Golub-I affects $\det(A)$, we divide Golub-I into two phases: the first phase keeps $|\det(A)|$ invariant, while the second one may change $|\det(A)|$. Accordingly, we identify and separate the first column $(b^T \quad \gamma e_1^T)^T$ of the matrix affected by Golub-I,

$$\begin{pmatrix} \bar{B} \\ \bar{C} \end{pmatrix} = \begin{pmatrix} b & \tilde{B} \\ \gamma e_1 & \tilde{C} \end{pmatrix}.$$

In the first phase the columns of $\binom{\tilde{B}}{\tilde{C}}$ are permuted, so that the first column of the permuted $\tilde{C}$ has largest norm among all columns of $\tilde{C}$, and then the permuted $\tilde{C}$ is retriangularised to give $\dot{C}$. In the second phase, the relevant matrix elements are γ and the nonzero elements α and β of $\dot{C}e_1$,

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cccc} k & k+1 & & \end{array} \\ \begin{pmatrix} * & * & * & * \\ & \gamma & \alpha & * \\ & & \beta & * \\ & & & * \end{pmatrix} \end{array}.$$

Golub-I permutes columns k and $k+1$ if $\gamma^2 < \alpha^2 + \beta^2$, in which case the matrix becomes

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cccc} k & k+1 & & \end{array} \\ \begin{pmatrix} * & * & * & * \\ & \alpha & \gamma & * \\ & \beta & & * \\ & & & * \end{pmatrix} \end{array}.$$

The matrix is retriangularised by eliminating β via a Givens rotations from the left, which affects only rows k and $k+1$ and results in

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cccc} k & k+1 & & \end{array} \\ \begin{pmatrix} * & * & * & * \\ & \sqrt{\alpha^2+\beta^2} & x & * \\ & & x & * \\ & & & * \end{pmatrix} \end{array},$$

where the two x represent new numbers. Other than the kth diagonal element, which changed from γ to $\sqrt{\alpha^2+\beta^2}$, no diagonal element of A changed. But the kth diagonal element underwent a strict increase in magnitude since $|\gamma| < \sqrt{\alpha^2+\beta^2}$, and therefore $|\det(A)|$ is a strictly increasing function during Hybrid-I. Consequently, algorithm Hybrid-I must halt.

Section 15 presents some numerical experiments on the running time of Hybrid-I.

13. Algorithm Hybrid-II. In this section we present algorithm Hybrid-II, the Type-II version of Hybrid-I. According to the unification principle, Hybrid-II guarantees that

$$\sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)\sqrt{(k+1)(n-k)}$$
$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_{\max}(R_{22})}{\sqrt{(k+1)(n-k)}}$$

From the interlacing properties $(I1)$ and $(I2)$ it follows that Hybrid-I(k+1) guarantees the same bounds as Hybrid-II(k). Thus, one way to implement Hybrid-II(k) is via Hybrid-I(k+1).

ALGORITHM HYBRID-II(k)
Hybrid-I(k+1)

Although nonsingularity is needed for the application of the unification principle, this implementation of Hybrid-II(k) has the advantage of doing without the requirement that the matrix be nonsingular. However, to reduce the proof that Hybrid-I(k+1) halts to the proof for Hybrid-I(k) requires $\sigma_{k+1}(M) > 0$, which may not be true. Our proof that Hybrid-II halts does so without this assumption, and it also enables us to design the more accurate algorithm Hybrid-III by providing additional insight into the nature of the problem.

The basic idea of the proof is again to demonstrate the strict increase of the determinant of the leading $k \times k$ principal submatrix during Hybrid-II. Unfortunately, we cannot prove that the absolute value of the determinant of the leading $(k+1) \times (k+1)$ block is strictly increasing because that would necessitate the assumption $\sigma_{k+1}(M) > 0$. To facilitate understanding of the proof, we first describe in more detail the implementation of Hybrid-II(k) based on Hybrid-I(k+1).

ALGORITHM HYBRID-II(k)
$R^{(0)} = R$, $l = 0$
Repeat
$l = l + 1$, *permuted* $= 0$
Set

$$R^{(l)} = \begin{pmatrix} A & B \\ & C \end{pmatrix},$$

where A is of order k and C is of order $n-k$.
Golub-I:

1. Find the column $k+j$ of $R^{(l)}$ such that $\|Ce_j\| = \max_{1 \leq i \leq n-k} \|Ce_i\|$
2. **If** $\|Ce_1\| < \|Ce_j\|$ **then**
   permuted $= 1$
   Exchange columns $k+1$ and $k+j$ of $R^{(l)}$
   Retriangularise it from the left with orthogonal transformations to get

$$\hat{R}^{(l)} = \begin{pmatrix} \hat{A} & \hat{B} \\ & \hat{C} \end{pmatrix},$$

where $\hat{A}$ is of order $k+1$ and $\hat{C}$ is of order $n-k-1$.
Stewart-II:

3. Find the column j of $\hat{R}^{(l)}$ such that $\|e_j^T \hat{A}^{-1}\| = \max_{1 \le i \le k+1} \|e_i^T \hat{A}^{-1}\|$
4. **If** $\|e_{k+1}^T \hat{A}^{-1}\| < \|e_j^T \hat{A}^{-1}\|$ **then**
   permuted = 1
   Exchange columns j and $k+1$ of $\hat{R}^{(l)}$
   Retriangularise it from the left with orthogonal transformations to get $R^{(l+1)}$.

until not *permuted*
The final matrix is

$$R = \begin{pmatrix} \hat{R}_{11} & \hat{R}_{12} \\ & \hat{R}_{22} \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix},$$

where $\hat{R}_{11}$ is of order $k+1$ and R_{11} is of order k.

As in Hybrid-I, the two if statements assure that permutations are performed only in case of a strict inequality but not in case of a tie.

Again, as we had assumed that $\sigma_k(M) > 0$, we can assume that our initial ordering of columns is such that $|\det(A)| > 0$. Although this proof is based on Hybrid-I(k+1) it is slightly different from the proof we gave for Hybrid-I(k) because now we are focusing on column k instead of column $k+1$. Clearly, Golub-I does not affect $|\det(A)|$ but Stewart-II does. We divide Stewart-II into two phases. The first phase keeps $|\det(A)|$ invariant while the second phase may change $|\det(A)|$. Accordingly, we identify and separate the last column $(\, a^T \quad \alpha e_1^T \,)^T$ of the matrix affected by Stewart-II,

$$\hat{A} = \begin{pmatrix} \tilde{A} & a \\ & \alpha \end{pmatrix}.$$

In the first phase the columns of $\tilde{A}$ are permuted, so that the last row of $\tilde{A}^{-1}$ has largest norm among all rows of $\tilde{A}^{-1}$, and then the permuted $\tilde{A}^{-1}$ is retriangularised from the right to give $\dot{A}^{-1}$. In the second phase the relevant matrix elements are α, the element β above it, and the trailing nonzero γ of $e_k^T \dot{A}$,

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{ccc} k & k+1 & \end{array} \\ \begin{pmatrix} * & * & * \\ & \gamma & \beta \\ & & \alpha \end{pmatrix} \end{array}.$$

Since Stewart-II is the Type-II version of Golub-I, it permutes to the last position the column corresponding to the row with largest norm in the inverse, whose relevant elements are

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{ccc} k & k+1 & \end{array} \\ \begin{pmatrix} * & * & * \\ & \frac{1}{\gamma} & -\frac{\beta}{\gamma\alpha} \\ & & \frac{1}{\alpha} \end{pmatrix} \end{array}.$$

Stewart-II permutes columns k and $k+1$ if

$$\frac{1}{\gamma^2} + \frac{\beta^2}{\gamma^2\alpha^2} > \frac{1}{\alpha^2} \qquad \text{or} \qquad \gamma^2 < \alpha^2 + \beta^2.$$

But this is the same situation as in Hybrid-I(k), and it follows that the kth diagonal element of R_{11} changes from γ to $\sqrt{\alpha^2+\beta^2}$ while all other diagonal elements of R_{11} remain unchanged. As we just proved that $|\gamma| < \sqrt{\alpha^2+\beta^2}$, $|\det(A)|$ is strictly increasing during Hybrid-II.

Because we were able to prove that Hybrid-II halts, requiring only that $\sigma_k(M) > 0$, we can show directly that Hybrid-II(k) satisfies

$$\sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)\sqrt{(k+1)(n-k)},$$
$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_{\max}(R_{22})}{\sqrt{(k+1)(n-k)}}.$$

The proof is similar to the one that establishes the bounds for Hybrid-I.

14. Algorithm Hybrid-III. Our last new algorithm is Hybrid-III, which satisfies

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{k(n-k+1)}},$$
$$\sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)\sqrt{(k+1)(n-k)}.$$

There are several implementations of Hybrid-III. We present the one that is simplest to describe. This implementation, motivated by the fact that the determinant of the leading principal submatrix of order k is a strictly increasing function in both Hybrid-I and Hybrid-II, consists of running Hybrid-I and Hybrid-II in alternation until no more permutations take place.

ALGORITHM HYBRID-III(k)
Repeat
    Hybrid-I(k)
    Hybrid-II(k)
Until no permutations occur

The halting argument for Hybrid-III follows easily from the halting of Hybrid-I and Hybrid-II. We had shown earlier that during Hybrid-I and Hybrid-II, the determinant of the leading $k \times k$ principal submatrix is a strictly increasing function. So it must be true during Hybrid-III also. Hence Hybrid-III halts.

When Hybrid-III has halted, both Hybrid-I and Hybrid-II do not cause any further permutations in the matrix. Therefore the bounds guaranteed by Hybrid-I and Hybrid-II must hold simultaneously now. That is

$$\sigma_{\min}(R_{11}) \geq \frac{\sigma_k(M)}{\sqrt{k(n-k+1)}},$$
$$\sigma_{\max}(R_{22}) \leq \sigma_{k+1}(M)\sqrt{(k+1)(n-k)}$$

must be true.

Because we do not know whether the solution of Problem-I also implies the solution of Problem-II or vice versa, it is not clear whether the output of algorithm Hybrid-I also satisfies the bounds that govern the output from Hybrid-II. In particular we are therefore not able to compare the operation counts of Hybrid-III with those of Hybrid-I or Hybrid-II.

REMARK 14.1. As we mentioned earlier, it is more practical to replace algorithm Stewart-II in the hybrid algorithms with algorithm Chan-II.

Moreover, there are other ways of implementing algorithm Hybrid-III. For example, replacing the Stewart-II part of Hybrid-I with a simpler algorithm results in an Hybrid-III algorithm: Instead of moving the most linearly dependent column to the kth position, in turn permute *every* one of the leading k columns to the kth position. Obviously, this algorithm solves Problem-I and Problem-II simultaneously. The corresponding Type-II version, which involves replacing the Golub-I part of Hybrid-II, also solves Problem-I and Problem-II simultaneously according to the unification principle. This idea has been taken up in [32].

However, we believe that the original version of Hybrid-III described at the beginning of this section is more efficient in practice than the latter two (provided it is properly implemented with good condition number estimators in place of Golub-I and Stewart-II, and run as a postprocessor to either Golub-I or Chan-II).

15. Some numerical experiments. Although we have demonstrated that the three hybrid algorithms halt in exact arithmetic, we know very little about their worst-case running times. In this section we present some preliminary numerical results for Hybrid-I, which also apply to Hybrid-II and Hybrid-III as the implementations for the latter two algorithms can be based on Hybrid-I. In practice, the hybrid algorithms are best run as postprocessors to the more efficient greedy algorithms, like Golub-I or Chan-II.

In the experiments to follow, we counted the number of iterations in Hybrid-I when it is run after Golub-I. To prevent cycling in the algorithm due to roundoff errors, we carried out permutations only if the pivot increased by more than $n^2\epsilon$, where ϵ is the machine precision. To estimate the dependence of the running time of Hybrid-I on the matrix size n and the separation of the singular values $\sigma_k(M)/\sigma_{k+1}(M)$, we generated fifty random matrices of size fifty, to which we applied Hybrid-I with $k = 37$. Then we multiplied the last $n - k$ singular values of these fifty matrices by 0.1 to increase the separation between the singular values but did not change the singular vectors. Hybrid-I was applied to these fifty new matrices. The same process was repeated on one hundred random matrices of size one hundred with $k = 75$. Table 15.1 shows how many times Hybrid-I required a certain number of iterations. Hybrid-I seems to require fewer iterations when the gap between $\sigma_k(M)$ and $\sigma_{k+1}(M)$ is larger, and—in these experiments, at least—the number of iterations does not deteriorate too much with increase in matrix size.

16. Conclusion. In this paper we proposed three optimisation problems which we called rank-revealing QR (RRQR) problems. We presented a unifying treatment of the existing algorithms by placing them in a hierarchy of greedy algorithms. Finally, we presented three new hybrid algorithms for solving the three rank-revealing problems. Unfortunately, we were not able to estimate the worst-case running time of the hybrid algorithms.

Most of the discussion for the RRQR factorisations can be extended in a simple manner to rank-revealing LU (RRLU) factorisations [8], [27] by replacing orthogonal transformations with elementary Gauss transformations and row interchanges for partial pivoting. Partial pivoting prevents the ill conditioning of the Gauss transformations. Compared to RRQR factorisations, the bounds for RRLU factorisations

TABLE 15.1
Hybrid-I run-time estimate.

Matrix size →	50	50	100	100
$k \rightarrow$	37	37	75	75
$\text{Avg}(\frac{\sigma_k(M)}{\sigma_{k+1}(M)}) \rightarrow$	1.0804	10.804	1.0406	10.406
No. of iter. ↓	↓ no. of occurrences ↓			
1	21	25	16	45
2	7	8	3	8
3	5	3	9	9
4	5	4	15	15
5	4	4	11	6
6	1	5	13	6
7	0	0	6	1
8	1	0	7	5
9	4	1	4	1
10	1	0	7	1
11	1	0	1	1
12	0	0	0	2
13	0	0	4	0
14	0	0	0	0
15	0	0	0	0
16	0	0	1	0
17	0	0	1	0
18	0	0	1	0
19	0	0	1	0
Total	50	50	100	100

are generally worse and, due to pivoting and the resulting fill-in, their operation are counts higher. It is not clear to us which applications would benefit from RRLU factorisations.

In a subsequent paper [13] we show that very naturally the hybrid algorithms give rise to new algorithms for computing the URV decomposition [34]–[36] and also to a new divide-and-conquer algorithm for the SVD. In fact, using a preceding RRQR algorithm to accelerate the computation of eigenvalues or singular values is not new, see for instance [15], [24], [38] where a Jacobi method is preceded by QR with column pivoting.

In this paper, we present only one algorithm for each of the three optimisation problems, but one can easily design other kinds of approximate and exact algorithms. Our motivation for the three hybrid algorithms was to perform column interchanges based on what we believed would result in a high rate of convergence. But sometimes one may want to trade off number of column exchanges for maintainance of sparsity [3], [4], [30] or minimisation of communication costs.

The ideas presented in this paper may aid in the design of special-purpose algorithms. Instead of choosing the best two columns to exchange, one could compromise and choose a column exchange that maintains sparsity or keeps communication costs low, while still ensuring that the determinant of the leading $k \times k$ principal submatrix increases strictly so that the algorithm halts. We hope that the ideas presented in this paper prove helpful in developing algorithms for such problems.

Acknowledgments. We thank Françoise Chatelin and Axel Ruhe for helpful discussions and for bringing reference [16] to our attention.

REFERENCES

[1] G. ADAMS, M. GRIFFIN, AND G. STEWART, *Direction-of-arrival estimation using the rank-revealing URV decomposition*, International Conference on Acoustics, Speech and Signal Processing 91, 1991, pp. 1385–1388.

[2] C. BISCHOF, *Incremental condition estimation*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 312–322.

[3] C. BISCHOF AND P. HANSEN, *A block algorithm for computing rank-revealing QR-factorizations*, Numerical Algorithms, 2 (1992), pp. 371–392.

[4] ———, *Structure-preserving and rank-revealing QR-factorizations*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1332–1350.

[5] C. BISCHOF, D. PIERCE, AND J. LEWIS, *Incremental condition estimation for sparse matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 644–659.

[6] C. BISCHOF AND G. SHROFF, *On updating signal subspaces*, IEEE Transactions on Signal Processing, 40 (1992), pp. 96–105.

[7] P. BUSINGER AND G. GOLUB, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.

[8] T. CHAN, *On the existence and computation of LU-factorizations with small pivots*, Math. Comp, 42 (1984), pp. 535–547.

[9] ———, *Rank revealing QR factorizations*, Linear Algebra and its Applications, 88/89 (1987), pp. 67–82.

[10] T. CHAN AND P. HANSEN, *Computing truncated singular value decomposition least squares solutions by rank revealing QR factorizations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 519–530.

[11] ———, *Low-rank revealing QR factorizations*, Tech. Report 91-08, UCLA Comp. and App. Math., Los Angeles, 1991; Numer. Linear Algebra Appl., to appear.

[12] ———, *Some applications of the rank revealing QR factorization*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 727–741.

[13] S. CHANDRASEKARAN AND I. IPSEN, *Analysis of a QR algorithm for computing singular values*, Research Report 917, Department of Computer Science, Yale University, New Haven, CT, 1992.

[14] P. COMON AND G. GOLUB, *Tracking a few extreme singular values and vectors in signal processing*, Proc. IEEEE, 78 (1990), pp. 1327–1343.

[15] J. DEMMEL AND K. VESELIĆ, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[16] D. FADDEEV, V. KUBLANOVSKAYA, AND V. FADDEEVA, *Sur les systèmes linéaires algébraiques de matrices rectangulaires et mal-conditionnées*, Colloq. Internat. du C.N.R.S. Besançon 1966, No. 165 (1968), pp. 161–170.

[17] L. FOSTER, *Rank and null space calculations using matrix decomposition without column interchanges*, Linear Algebra Appl., 74 (1986), pp. 47–71.

[18] ———, *The probability of large diagonal elements in the QR factorization*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 531–544.

[19] G. GOLUB, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[20] G. GOLUB, V. KLEMA, AND G. STEWART, *Rank degeneracy and least squares problems*, Tech. Report STAN-CS-76-559, Computer Science Department, Stanford University, Stanford, CA, 1976.

[21] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, The Johns Hopkins Press, Baltimore, MD, 1989.

[22] W. GRAGG AND G. STEWART, *A stable variant of the secant method for solving nonlinear equations*, SIAM J. Numer. Anal., 13 (1976), pp. 889–903.

[23] P. HANSEN, *Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 503–518.

[24] V. HARI AND K. VESELIĆ, *On Jacobi methods for singular value decompositions*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 741–754.

[25] N. HIGHAM, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29 (1987), pp. 575–596.

[26] H. HONG AND C. PAN, *The rank-revealing QR decomposition and SVD*, Math. Comp., 58 (1992), pp. 213–232.

[27] T. HWANG, W. LIN, AND E. YANG, *Rank-revealing LU-factorizations*, Linear Algebra Appl., 175 (1992), pp. 115–141.

[28] W. KAHAN, *Numerical linear algebra*, Canadian Math. Bull., 9 (1966), pp. 757–801.

[29] V. KANE, R. WARD, AND G. DAVIS, *Assessment of linear dependencies in multivariate data*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 1022–1032.
[30] J. LEWIS AND D. PIERCE, *Sparse rank revealing QR factorization*, Tech. Report MEA-TR-193, Boeing Computer Services, Seattle, WA, 1992.
[31] R. MATHIAS AND G. W. STEWART, *A Block QR Algorithm and the Singular Value Decomposition*, Linear Algebra Appl., 182 (1993), pp. 91–100.
[32] C. PAN AND P. TANG, *Bounds on singular values revealed by QR factorizations*, Preprint MCS-P332-1032, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1992.
[33] G. STEWART, *Rank degeneracy*, SIAM J. Sci. Stat. Comput., 5 (1984), pp. 403–413.
[34] ———, *Incremental condition calculation and column selection*, Tech. Report UMIACS-TR 90-87, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, 1990.
[35] ———, *An updating algorithm for subspace tracking*, IEEE Transactions on Signal Processing, SP-40 (1992), pp. 1535–1541.
[36] ———, *Updating a rank-revealing ULV decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 494–499.
[37] S. VAN HUFFEL AND J. VANDEWALLE, *Subset selection using the total least squares approach in collinearity problems with errors in the variables*, Linear Algebra Appl., 88/89 (1987), pp. 695–714.
[38] K. VESELIĆ AND V. HARI, *A note on a one-sided jacobi algorithm*, Numer. Math, 56 (1989), pp. 627–633.
[39] S. WOLD, A. RUHE, H. WOLD, AND W. DUNN, III, *The collinearity problem in linear regression. The partial least squares* (PLS) *approach to generalized inverses*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 735–743.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 623–635, April 1994

017

SYMMETRIC TOEPLITZ MATRICES WITH TWO PRESCRIBED EIGENPAIRS*

MOODY T. CHU†‡ AND MELISSA A. ERBRECHT†

Abstract. The inverse problem of constructing a real symmetric Toeplitz matrix based on two prescribed eigenpairs is considered. Two new results are obtained. First, it is shown that the dimension of the subspace of Toeplitz matrices with two generically prescribed eigenvectors is independent of the size of the problem, and, in fact, is either two, three, or four, depending upon whether the eigenvectors are symmetric or skew-symmetric and whether n is even or odd. This result is quite notable in that when only one eigenvector is prescribed the dimension is known to be at least $[(n+1)/2]$. Taking into account the prescribed eigenvalues, the authors then show how each unit vector in the null subspace of a certain matrix uniquely determines a Toeplitz matrix that satisfies the prescribed eigenpairs constraint. The cases where two prescribed eigenpairs uniquely determine a Toeplitz matrix are explicitly characterized.

Key words. Toeplitz matrix, eigenvector, inverse problem

AMS subject classifications. 65F15, 15A18

1. Introduction.

A real $n \times n$ matrix $T = (t_{ij})$ is symmetric and Toeplitz if there exist real scalars $r_1, \ldots, r_n$ such that

$$t_{ij} = r_{|i-j|+1}$$

for all i and j. Clearly a symmetric Toeplitz matrix is uniquely determined by the entries of its first column. Thus we shall denote a symmetric Toeplitz matrix by $T(r)$ if its first column is given by the vector $r \in R^n$.

Due to their role in important applications like the trigonometric moment problem, the Szegö theory, and signal processing, many properties of Toeplitz matrices have been studied over the years. For example, efficient algorithms have been devised to solve a Toeplitz system of equations in $O(n^2)$ time. Brief discussions of algorithms and more references for solving Toeplitz systems can be found in [7, §4.7]. In this paper, we are more interested in the spectral properties of a symmetric Toeplitz matrix.

It is easy to see that if $Tv = \lambda v$ and λ is an eigenvalue of multiplicity one, then either $Ev = v$ or $Ev = -v$, where $E = (e_{ij}) \in R^{n \times n}$ is the exchange matrix defined by

$$e_{ij} = \begin{cases} 1 & \text{if } i + j = n + 1, \\ 0 & \text{otherwise.} \end{cases}$$

Accordingly, we call such an eigenvector either symmetric or skew-symmetric. For eigenvalues of multiplicity greater than one, the corresponding eigenspace has an orthonormal basis that splits as evenly as possible between symmetric and skew-symmetric eigenvectors [5, Thm. 8]. Thus it is sensible to say that the eigenvectors of a symmetric Toeplitz matrix can be split into two classes. More specifically, as any

* Received by the editors October 31, 1991; accepted for publication (in revised form) July 24, 1992.

† Department of Mathematics, North Carolina State University, Raleigh, North Carolina 27695-8205 (chu@gauss.math.ncsu.edu)

‡ This author's research was supported in part by National Science Foundation grants DMS-9006135 and DMS-9123448.

symmetric centrosymmetric matrix [2, Thm. 2], a symmetric Toeplitz matrix of order n has $\lceil n/2 \rceil$ symmetric and $\lfloor n/2 \rfloor$ skew-symmetric eigenvectors. For convenience, we use $\sigma^+(T)$ and $\sigma^-(T)$ to denote, respectively, the spectrum of eigenvalues corresponding to symmetric and skew-symmetric eigenvectors. Other spectral properties of Toeplitz matrices can be found in [2], [5], [9], [10], and the references contained therein.

The inverse Toeplitz eigenvalue problem (ITEP) has been an interesting yet difficult question studied in the literature. The problem is to find a vector $r \in R^n$ such that the Toeplitz matrix $T(r)$ has a prescribed real spectrum $\{\lambda_1, \ldots, \lambda_n\}$. At present, the ITEP remains unsolved when $n \geq 5$ [5]. (See the note added in proof at the end of this paper.) Partial results and numerical algorithms for the ITEP can be found in, for example, [3], [6], [8], and [11].

In [2, Thm. 3] it is claimed that any real $n \times n$ matrix that has a set of n real orthonormal eigenvectors, each being either symmetric or skew-symmetric, is both symmetric and centrosymmetric. Apparently it is another interesting and difficult problem to identify an orthogonal matrix so that its columns are eigenvectors of some Toeplitz matrix.

In [4] it is proved that being symmetric or skew-symmetric is sufficient for a single vector to be an eigenvector of a Toeplitz matrix. In fact, let

$$S_0(v) := \{r \in R^n | T(r)v = 0\} \tag{1}$$

denote the collection of (the first columns of) all symmetric Toeplitz matrices for which v is an eigenvector corresponding to the eigenvalue 0. It can be shown that $S_0(v)$ is a linear subspace with dimension [4, Cor. 1]

$$\dim(S_0(v)) = n - \pi(v), \tag{2}$$

where

$$\pi(v) := \begin{cases} \lceil n/2 \rceil & \text{if } v \text{ is symmetric,} \\ \lfloor n/2 \rfloor & \text{if } v \text{ is skew-symmetric} \end{cases} \tag{3}$$

is called the index of v. Clearly $T(r)v = \lambda v$ if and only if $r - \lambda w \in S_0(v)$, where $w = [1, 0, \ldots, 0]^T$ is the first standard basis vector in R^n. Thus the set

$$S(v) := \{r \in R^n | T(r)v = \lambda v \text{ for some } \lambda \in R\} \tag{4}$$

is precisely the direct sum $\langle w \rangle \oplus S_0(v)$.

Suppose now $\{v^{(1)}, \ldots, v^{(k)}\}$, $k \geq 1$, is a set of real orthonormal vectors, each being symmetric or skew-symmetric. Then $\cap_{i=1}^k S(v^{(i)})$ contains all symmetric Toeplitz matrices for which each v_i is an eigenvector. Evidently, $w \in S(v^{(i)})$ for all i. So $\cap_{i=1}^k S(v^{(i)})$ is at least of dimension 1. We are interested in studying the following problem.

Problem 1. Obtain a nontrivial lower bound on the dimension of $\bigcap_{i=1}^k S(v^{(i)})$.

Toward this end, we show in this paper that for the case $k = 2$, the dimension of $\bigcap_{i=1}^2 S(v^{(i)})$ is almost always independent of of the size of the problem, and, in fact, is either two, three, or four, depending upon whether the eigenvectors are symmetric or skew-symmetric.

In view of the ITEP, another interesting inverse problem arises.

Problem 2. Given a set of real orthonormal vectors, $\{v^{(1)}, \ldots, v^{(k)}\}$, $k \geq 1$, each symmetric or skew-symmetric, and a set of real numbers $\{\lambda_1, \ldots, \lambda_k\}$, find a symmetric Toeplitz matrix T (other than a scalar matrix) such that

$$Tv^{(i)} = \lambda_i v^{(i)}, \qquad i = 1, \ldots, k. \tag{5}$$

We note in Problem 2 that T is required to be Toeplitz; thus the description of the given eigenpairs cannot be totally arbitrary. For instance, it is improper to request that all vectors be symmetric while $k > \lceil n/2 \rceil$. We recall a conjecture in [5] that a *universal* distribution of eigenvalues for Toeplitz matrices should be such that $\sigma^+(T)$ and $\sigma^-(T)$ interlace. Thus a Toeplitz matrix whose spectrum does not satisfy the interlaced distribution is perhaps more difficult to find [8]. On the other hand, as far as Problem 2 is concerned, there is a possibility that the remaining unspecified eigenpairs could make up the total spectrum so that the interlaced condition is eventually realized.

For the case $k = 2$, we show in this paper that in each direction in the subspace $\bigcap_{i=1}^{2} S(v^{(i)})$ there is one and only one Toeplitz matrix for Problem 2. In particular, we show that if n is odd and if at least one of the given eigenvectors is symmetric, or if n is even and one eigenvector is symmetric and the other is skew-symmetric, then the Toeplitz matrix is uniquely determined.

2. An example. As we shall only consider the case $k = 2$ throughout the paper, it is more convenient to denote, henceforth, the eigenvectors $v^{(1)}$ and $v^{(2)}$ by u and v, respectively.

We begin our study of the set $S(u) \cap S(v)$ with the special case where $n = 3$. The example should shed some insights on the higher dimensional case.

Due to the special eigenstructure of symmetric Toeplitz matrices, it is necessary that one of the two given eigenvectors, say u, must be symmetric. Denote $u = [u_1, u_2, u_1]^T$ where $2{u_1}^2 + u_2^2 = 1$. It can also be proved that the skew-symmetric vector $\hat{u} = [1/\sqrt{2}, 0, -1/\sqrt{2}]^T$ is a universal eigenvector for every symmetric Toeplitz matrix of order 3. Thus, given u, we imply from the orthogonality condition that the second prescribed eigenvector v must be either the second or the third column of the matrix

$$Q = \begin{bmatrix} u_1 & \frac{1}{\sqrt{2}} & -\frac{u_2}{\sqrt{2}} \\ u_2 & 0 & u_1\sqrt{2} \\ u_1 & -\frac{1}{\sqrt{2}} & -\frac{u_2}{\sqrt{2}} \end{bmatrix}. \tag{6}$$

In other words, one symmetric eigenvector completely determines all three orthonormal eigenvectors (up to a $\pm$ sign). It follows, from [4], that $\dim S(u) \cap S(v) = 2$. (Note that if the orthogonality condition is violated, then trivially $S(u) \cap S(v) = \langle w \rangle$.)

Let $\Lambda = \operatorname{diag}\{\lambda_1, \lambda_2, \lambda_3\}$. We already know $Q\Lambda Q^T$ is a centrosymmetric matrix. It is not difficult to see that $Q\Lambda Q^T$ is Toeplitz if and only if

$$(3{u_1}^2 - 1)\lambda_1 + \frac{\lambda_2}{2} + \left(\frac{1}{2} - 3{u_1}^2\right)\lambda_3 = 0. \tag{7}$$

From (7), the following facts can easily be observed.

LEMMA 2.1. *Let* $2{u_1}^2 + u_2^{21} = 1$. *Then:*

(i) *If λ_1 and λ_3 are given scalars, there is a unique symmetric Toeplitz matrix T such that*

$$T\begin{bmatrix} u_1 & -\frac{u_2}{\sqrt{2}} \\ u_2 & u_1\sqrt{2} \\ u_1 & -\frac{u_2}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} u_1 & -\frac{u_2}{\sqrt{2}} \\ u_2 & u_1\sqrt{2} \\ u_1 & -\frac{u_2}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_3 \end{bmatrix}. \tag{8}$$

(ii) *If $u_1 \neq \pm\sqrt{1/6}$ and λ_1 and λ_2 are given scalars, there is a unique symmetric Toeplitz matrix T such that*

$$T\begin{bmatrix} u_1 & \frac{1}{\sqrt{2}} \\ u_2 & 0 \\ u_1 & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} u_1 & \frac{1}{\sqrt{2}} \\ u_2 & 0 \\ u_1 & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \tag{9}$$

(iii) *If $u_1 = \pm\sqrt{1/6}$, then there are infinitely many symmetric Toeplitz matrices T that satisfy* (9) *if $\lambda_1 = \lambda_2$; however, if $\lambda_1 \neq \lambda_2$, then* (9) *does not hold for any symmetric Toeplitz matrix T.*

3. General consideration. We now consider the case for general n. When v is an eigenvector, the idea of rewriting the matrix-vector product [4]

$$T(r)v = M(v)r \tag{10}$$

can be very useful. The following lemma can be observed.

LEMMA 3.1. *The columns of $M(v)$ have the same symmetry as v has. That is, $EM(v) = \pm M(v)$ if and only if $Ev = \pm v$.*

Thus only the first $\pi(v)$ rows of $M(v)$ need to be considered. For convenience, let $p := \lceil n/2 \rceil$ and let $N(v)$ denote the $p \times n$ submatrix of the first p rows of $M(v)$. It is easy to verify that $N(v)$ can be decomposed into blocks

$$N(v) = [h(v), H(v), 0] + [0, L(v), 0] + [0, 0, U(v)], \tag{11}$$

where $h(v)$ is a $p \times 1$ column vector, $\tilde{H}(v) := [h(v), H(v)] = (\tilde{h}_{ij}(v))$ is the $p \times p$ Hankel matrix

$$\tilde{h}_{ij}(v) := v_{i+j-1}, \tag{12}$$

$\tilde{L}(v) := [0, L] = (\tilde{l}_{ij}(v))$ is the $p \times p$ lower triangular matrix

$$\tilde{l}_{ij}(v) := \begin{cases} v_{i-j+1} & \text{if } 1 < j \leq i, \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

and $U(v) = (u_{ij}(v))$ is the $p \times (n-p)$ triangular matrix

$$u_{ij}(v) := \begin{cases} v_{p+i+j-1} & \text{if } i + j \leq n - p + 1, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

We note that the last row of $N(v)$ is identically zero when n is odd and v is skew-symmetric. The rows of $N(u)$ and $N(v)$ will be used to construct a larger matrix.

Suppose that

$$\begin{aligned} T(r)u &= \lambda_1 u, \\ T(r)v &= \lambda_2 v. \end{aligned} \tag{15}$$

Then the vector r must be such that the linear equations

$$\begin{aligned} N(u)(r - \lambda_1 w) &= 0, \\ N(v)(r - \lambda_2 w) &= 0 \end{aligned} \tag{16}$$

are satisfied. If we write

$$x := [r_1 - \lambda_2, r_1 - \lambda_1, r_2, \ldots, r_n]^T, \tag{17}$$

then the system (16) is equivalent to

$$\tilde{M}(u, v)x = 0, \tag{18}$$

where $\tilde{M}(u, v)$ is the $(2p) \times (n + 1)$ matrix defined by

$$\tilde{M}(u, v) := \begin{bmatrix} 0 & h(u) & H(u) + L(u) & U(u) \\ h(v) & 0 & H(v) + L(v) & U(v) \end{bmatrix}. \tag{19}$$

Given symmetric or skew-symmetric vectors u and v, a solution to (18) can be used to construct a Toeplitz matrix in the following way.

LEMMA 3.2. *Suppose that* $[x_0, x_1, \ldots, x_n]^T$ *is a solution to* (18). *For arbitrary real numbers* λ_1 *and* α, *define*

$$\begin{aligned} r_1 &:= \alpha x_1 + \lambda_1, \\ r_i &:= \alpha x_i \quad \text{for } i = 2, \ldots, n, \end{aligned} \tag{20}$$

and

$$\lambda_2 := \alpha(x_1 - x_0) + \lambda_1. \tag{21}$$

Then u *and* v *are eigenvectors of the Toeplitz matrix* $T(r)$. *In other words,* $S(u) \cap S(v)$ *is the direct sum of the subspace spanned by* w *and the subspace obtained by deleting the first component from* $\ker(\tilde{M})$.

On the other hand, suppose the two eigenvalues λ_1 and λ_2 are prescribed. Then (21) implies that the constant α in (20) must be

$$\alpha = \frac{\lambda_1 - \lambda_2}{x_0 - x_1}, \tag{22}$$

provided $x_0 \neq x_1$. The conclusion is made in the following lemma.

LEMMA 3.3. *Suppose that* x *is a nontrivial solution of* (18) *satisfying* $x_0 \neq x_1$. *Then corresponding to the direction of* x, *there is a unique solution to Problem* 2 *when* $k = 2$.

The question now is to determine the null space of $\tilde{M}(u, v)$. It is convenient to use the abbreviated notation $\tilde{M} = \tilde{M}(u, v)$. It turns out that the dimension depends upon whether n is even or odd and whether the two eigenvectors are symmetric or skew-symmetric. In any case, we shall show that $\tilde{M}$ has a nontrivial null space. It is most interesting to note that the dimension does not depend upon the size of n. We discuss the following different cases.

Case 1. n is odd and both eigenvectors are symmetric. When n is odd, the Hankel matrix $\tilde{H}(v)$ for a symmetric vector v takes the special form

$$\tilde{H}(v) = \begin{bmatrix} v_1 & v_2 & \cdots & v_{p-1} & v_p \\ v_2 & v_3 & & v_p & v_{p-1} \\ \vdots & & & & \vdots \\ v_{p-1} & v_p & \cdots & v_3 & v_2 \\ v_p & v_{p-1} & \cdots & v_2 & v_1 \end{bmatrix}. \tag{23}$$

The corresponding $U(v)$ becomes

$$U(v) = EL(v) = \begin{bmatrix} v_{p-1} & v_{p-2} & \cdots & v_2 & v_1 \\ v_{p-2} & & \cdots & v_1 & 0 \\ \vdots & & & & \vdots \\ v_1 & 0 & \cdots & & 0 \\ 0 & & \cdots & & 0 \end{bmatrix}. \tag{24}$$

It is useful to illustrate the basic structure of $\tilde{M}$ with a simple example when both u and v are symmetric. When $n = 5$, we have

$$\tilde{M} = \begin{bmatrix} 0 & u_1 & u_2 & u_3 & u_2 & u_1 \\ 0 & u_2 & u_3 + u_1 & u_2 & u_1 & 0 \\ 0 & u_3 & 2u_2 & 2u_1 & 0 & 0 \\ v_1 & 0 & v_2 & v_3 & v_2 & v_1 \\ v_2 & 0 & v_3 + v_1 & v_2 & v_1 & 0 \\ v_3 & 0 & 2v_2 & 2v_1 & 0 & 0 \end{bmatrix}. \tag{25}$$

The matrix $\tilde{M}$ in general is a square matrix of order $n + 1$. The determinant of $\tilde{M}$ is an algebraic expression involving independent variables $v_1, \ldots, v_p, u_1, \ldots, u_p$. It would not be too surprising if $\det(\tilde{M}) \neq 0$ for generic u and v (see the Appendix). Nevertheless, under the additional condition that u and v are perpendicular to each other, we will show by elementary row operations that $\tilde{M}$ is in fact rank deficient.

For any symmetric vector v, let the $p \times p$ upper triangular matrix $G(v)$ be defined by

$$G(v) := \begin{bmatrix} 2v_1 & 2v_2 & \cdots & 2v_{p-1} & v_p \\ 0 & 2v_1 & & 2v_{p-2} & v_{p-1} \\ \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & & & 2v_1 & v_2 \\ 0 & 0 & 0 & 0 & v_1 \end{bmatrix}. \tag{26}$$

We also define the $2p \times 2p$ matrix

$$\tilde{G} = \tilde{G}(u, v) := \begin{bmatrix} I & 0 \\ -G(v) & G(u) \end{bmatrix}, \tag{27}$$

which, in fact, is the accumulation of a sequence of elementary row operations. We remark here that the ordering of u and v is immaterial. If $u_1 = 0$, then the roles of u and v may as well be switched. The extremely rare case where both $u_1 = v_1 = 0$ can be reduced to a lower dimensional problem. Without loss of generality, therefore, we

may assume that $u_1 \neq 0$ and, hence, the matrix $\tilde{G}(u,v)$ is a nonsingular matrix. It follows that the product $\tilde{W} := \tilde{G}\tilde{M}$ has the same rank as $\tilde{M}$.

We now make an important claim.

LEMMA 3.4. *Suppose that n is odd and that the two symmetric vectors u and v are orthogonal. Then the matrix $\tilde{W}$ is rank deficient. In fact,*

$$p+1 \leq \operatorname{rank}(\tilde{\mathrm{W}}) \leq \mathrm{n}. \tag{28}$$

Proof. The proof is tedious but straightforward. For $i = 1, \ldots, p$, the ith component of $G(u)h(v)$ is given by

$$2 \sum_{\substack{s=1 \\ t-s=i-1}}^{p-i} u_s v_t + u_{p-i+1} v_p. \tag{29}$$

The first component is trivially seen to be

$$2 \sum_{s=1}^{p-1} u_s v_s + u_p v_p, \tag{30}$$

which is zero because u and v are perpendicular to each other. For the same reason, the first component of $-G(v)h(u)$ is zero.

The product $G(u)U(v)$ has the same triangular structure as $U(v)$. On the other hand, the (i,j)th component of $G(u)U(v)$ with $i+j \leq p$ is given by

$$2 \sum_{s+t=p-i-j+2} u_s v_t. \tag{31}$$

It is important to note that the summation (31) is a symmetric function of u and v. It follows that the $p \times (n-p)$ block

$$-G(v)U(u) + G(u)U(v) \tag{32}$$

is identically zero.

For $i = 1, \ldots, p$ and $j = 1, \ldots, p-1$, the (i,j)th component of $G(u)H(v)$ is given by

$$2 \sum_{\substack{s=1 \\ t-s=i+j-1}}^{p-i-j+1} u_s v_t + 2 \sum_{\substack{s=p-i-j+2 \\ t+s=2p-i-j+1}}^{p-i} u_s v_t + u_{p-i+1} v_{p-j} \tag{33}$$

if $j < p-i+1$; or

$$2 \sum_{\substack{s=1 \\ t+s=2p-i-j+1}}^{p-i} u_s v_t + u_{p-i+1} v_{p-j} \tag{34}$$

if $j \geq p-i+1$. The (i,j)th component of $G(u)L(v)$ is given by

$$2 \sum_{\substack{s=1 \\ t-s=i-j-1}}^{p-i} u_s v_t + u_{p-i+1} v_{p-j} \tag{35}$$

if $j \leq i-1$; or

$$2 \sum_{\substack{s=j-i+2 \\ t-s=i-j-1}}^{p-i} u_s v_t + u_{p-i+1} v_{p-j} \tag{36}$$

if $j > i-1$. Using (33) and (36), it follows that the $(1,j)$th component of $G(u)(H(v)+L(v))$ is given by

$$2 \sum_{\substack{s=1 \\ t-s=j}}^{p-j} u_s v_t + 2 \sum_{\substack{s=p-j+1 \\ t+s=2p-j}}^{p-1} u_s v_t + 2 \sum_{\substack{t=1 \\ s-t=j}}^{p-j} u_s v_t. \tag{37}$$

The first and the last summations in (37) are symmetric to each other. The second summation in (37) is a symmetric function of u and v. Overall, (37) is a symmetric function of u and v, which will be completely canceled by its counterpart in $-G(v)(H(u)+L(u))$.

By now we have proved that the $(p+1)$th row of $\tilde{W}$ is identically zero. If follows that the null space of $\tilde{M}$ is at least of dimension one.

Using (34), (35), and (36), it is further observed that the $(i, p-1)$th component of $G(u)(H(v)+L(v))$ is given by

$$2 \sum_{\substack{s=1 \\ t+s=p+2-i}}^{p} u_s v_t, \tag{38}$$

which, once again, is a symmetric function of u and v, and will be completely canceled by its counterpart in $-G(v)(H(u)+L(u))$. The zero structure of $\tilde{W}$ clearly indicates that (28) is true. □

The following example for the case $n = 5$ illustrates the typical structure of $\tilde{W}$:

$$\tilde{W} = \begin{bmatrix} 0 & u_1 & u_2 & u_3 & u_2 & u_1 \\ 0 & u_2 & u_3+u_1 & u_2 & u_1 & 0 \\ 0 & u_3 & 2u_2 & 2u_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2v_2u_1+v_3u_2 & -2v_1u_2-v_2u_3 & 2v_3u_1-2v_1u_3 & 0 & 0 & 0 \\ v_3u_1 & -v_1u_3 & 2v_2u_1-2v_1u_2 & 0 & 0 & 0 \end{bmatrix}. \tag{39}$$

Let W denote the lower left $(p-1) \times p$ submatrix of $\tilde{W}$. That is, W is the matrix obtained by deleting the first row and the last column of

$$[G(u)h(v), -G(v)h(u), -G(v)(H(u)+L(u)) + G(u)(H(v)+L(v))]. \tag{40}$$

The rank of $\tilde{W}$ can be less than n if and only if W is rank deficient, which will be true if and only if values of u_i and v_i are such that $\det(WW^T) = 0$. We note that $\det(WW^T)$ is a polynomial in the independent variables u_i and v_i. We note also that $\det(WW^T)$ is not identically zero (see the Appendix for a proof). Thus rank($\tilde{W}$) < n if and only if u_i and v_i come from a codimension-one surface. We conclude, therefore, that for almost all u and v satisfying $u^Tv = 0$, the matrix $\tilde{W}$ is of rank n. Unfortunately, even for the case $n = 5$ (see (39)), it is fairly complicated to express the rank deficiency of W in terms of components of u and v. At present

we cannot provide a further characterization of the set where W is rank deficient. Given the fact that the orthogonality of u and v has already been used to prove the rank deficiency of $\tilde{W}$, it is conceivably true that the orthogonality condition cannot be used again to reduce the rank of W.

In conclusion, we have proved the following theorem.

THEOREM 3.5. *Suppose that n is odd and that u and v are two symmetric vectors satisfying $u^T v = 0$. Then:*

1. *The dimension of $S(u) \cap S(v)$ is at least two.*
2. *For almost all u and v, the dimension of $S(u) \cap S(v)$ is exactly two.*
3. *For almost all u and v and for any values of λ_1 and λ_2, there exists a unique symmetric Toeplitz matrix T satisfying $Tu = \lambda_1 u$ and $Tv = \lambda_2 v$.*

Case 2. n is odd and both eigenvectors are skew-symmetric. When n is odd, the Hankel matrix $\tilde{H}(v)$ for a skew-symmetric vector v takes the form

$$\tilde{H}(v) = \begin{bmatrix} v_1 & v_2 & \cdots & v_{p-1} & 0 \\ v_2 & v_3 & & 0 & -v_{p-1} \\ \vdots & & & & \vdots \\ v_{p-1} & 0 & \cdots & -v_3 & -v_2 \\ 0 & -v_{p-1} & \cdots & -v_2 & -v_1 \end{bmatrix}. \tag{41}$$

The corresponding $U(v)$ becomes

$$U(v) = -EL(v) = \begin{bmatrix} -v_{p-1} & -v_{p-2} & \cdots & -v_2 & -v_1 \\ -v_{p-2} & & \cdots & -v_1 & 0 \\ \vdots & & & & \vdots \\ -v_1 & 0 & \cdots & & 0 \\ 0 & 0 & \cdots & & 0 \end{bmatrix}. \tag{42}$$

It follows that the last row of $N(v)$ is identically zero. For (18), it is now obvious that the kernel of $\tilde{M}$ is of dimension at least two. In fact, we can show the following lemma.

LEMMA 3.6. *Suppose that n is odd and that the skew-symmetric vectors u and v are perpendicular. Then*

$$p \leq \operatorname{rank}(\tilde{\mathrm{W}}) \leq \mathrm{n} - 2. \tag{43}$$

Indeed, for almost all u and v, $\operatorname{rank}(\tilde{\mathrm{W}}) = \mathrm{n} - 2$.

Proof. The proof is very similar to that of Lemma 3.4. So we simply outline a recipe for constructing the transformation matrix that does the elimination. The details of justification are omitted.

It suffices to consider the $2(p-1) \times (n+1)$ submatrix $\hat{M}$ obtained by deleting the pth and the $2p$th rows of $\tilde{M}$. For a skew-symmetric vector v, define the $(p-1) \times (p-1)$ matrix $G(v)$ by

$$G(v) := \begin{bmatrix} -v_1 & -v_2 & \cdots & -v_{p-2} & -v_{p-1} \\ 0 & v_1 & & v_{p-3} & v_{p-2} \\ \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & & & v_1 & v_2 \\ 0 & 0 & 0 & 0 & v_1 \end{bmatrix}. \tag{44}$$

Then construct the $2(p-1) \times 2(p-1)$ transformation matrix $\tilde{G}(u,v)$ in the same way as is defined in (27). It can be proved now that the pth row of the product $\hat{W} := \tilde{G}\hat{M}$ is identically zero. Furthermore, the lower right $(p-1) \times (p-1)$ submatrix of $\hat{W}$ is also identically zero. The assertion follows from these observations. □

As an example, for $n = 5$, the matrix $\hat{M}$ takes the form

$$\hat{M} = \begin{bmatrix} 0 & u_1 & u_2 & 0 & -u_2 & -u_1 \\ 0 & u_2 & u_1 & -u_2 & -u_1 & 0 \\ v_1 & 0 & v_2 & 0 & -v_2 & -v_1 \\ v_2 & 0 & v_1 & -v_2 & -v_1 & 0 \end{bmatrix}, \tag{45}$$

and after the transformation the matrix $\hat{W}$ looks like

$$\hat{W} = \begin{bmatrix} 0 & u_1 & u_2 & 0 & -u_2 & -u_1 \\ 0 & u_2 & u_1 & -u_2 & -u_1 & 0 \\ -v_1u_1 - v_2u_2 & v_1u_1 + v_2u_2 & 0 & 0 & 0 & 0 \\ u_1v_2 & -v_1u_2 & 0 & v_1u_2 - u_1v_2 & 0 & 0 \end{bmatrix}. \tag{46}$$

The third row of $\hat{W}$ is identically zero because $u^Tv = 0$.

We conclude this case by the following theorem.

THEOREM 3.7. *Suppose that $n \geq 5$ is odd and that u and v are two skew-symmetric vectors satisfying $u^Tv = 0$. Then:*

1. *The dimension of $S(u) \cap S(v)$ is at least four.*
2. *For almost all u and v, the dimension of $S(u) \cap S(v)$ is exactly four.*
3. *For almost all u and v and for any values of λ_1 and λ_2, the symmetric Toeplitz matrices T satisfying $Tu = \lambda_1 u$ and $Tv = \lambda_2 v$ form a two-dimensional manifold.*

Proof. By Lemma 3.6, the dimension of $\ker(\tilde{W})$ is almost always three. The first two assertions then follow from Lemma 3.2. The last assertion follows from Lemma 3.3. □

Case 3. n is odd, one eigenvector is symmetric and the other is skew-symmetric. A symmetric vector is always orthogonal to a skew-symmetric vector regardless of what the values of the components are. Thus, unlike the previous two cases, the orthogonality condition $u^Tv = 0$ no longer helps to reduce the rank of $\tilde{M}$. As $\tilde{M}$ does contain an identically zero row, we should have the same conclusion as in Theorem 3.5, which is given in Theorem 3.8.

THEOREM 3.8. *Suppose that n is odd and that u and v are symmetric and skew-symmetric vectors, respectively. Then:*

1. *The dimension of $S(u) \cap S(v)$ is at least two.*
2. *For almost all u and v, the dimension of $S(u) \cap S(v)$ is exactly two.*
3. *For almost all u and v and for any values of λ_1 and λ_2, there exists a unique symmetric Toeplitz matrix T satisfying $Tu = \lambda_1 u$ and $Tv = \lambda_2 v$.*

Case 4. n is even and both eigenvectors are symmetric. When n is even, the Hankel matrix $\tilde{H}(v)$ for a symmetric vector v takes the special form

$$\tilde{H}(v) = \begin{bmatrix} v_1 & v_2 & \cdots & v_{p-1} & v_p \\ v_2 & v_3 & & v_p & v_p \\ \vdots & & & & \vdots \\ v_{p-1} & v_p & \cdots & v_4 & v_3 \\ v_p & v_p & \cdots & v_3 & v_2 \end{bmatrix}. \tag{47}$$

Once again, we define

$$G(v) := \begin{bmatrix} v_1 & v_2 & \cdots & v_{p-1} & v_p \\ 0 & v_1 & & v_{p-2} & v_{p-1} \\ \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & & & v_1 & v_2 \\ 0 & 0 & 0 & 0 & v_1 \end{bmatrix} \tag{48}$$

and construct $\tilde{G}(u, v)$ according to (27). If the two symmetric vectors u and v are orthogonal, then it can be shown that the rank of the $n \times (n+1)$ matrix $\tilde{W} := \tilde{G}\tilde{M}$ satisfies

$$p + 1 \leq \operatorname{rank}(\tilde{W}) \leq n - 1 \tag{49}$$

and for almost all u and v, the dimension of $\ker(\tilde{M})$ is exactly two. So the following theorem is true.

THEOREM 3.9. *Suppose that n is even and that u and v are two symmetric vectors satisfying $u^T v = 0$. Then:*

1. *The dimension of $S(u) \cap S(v)$ is at least three.*
2. *For almost all u and v, the dimension of $S(u) \cap S(v)$ is exactly three.*
3. *For almost all u and v and for any values of λ_1 and λ_2, the symmetric Toeplitz matrices T satisfying $Tu = \lambda_1 u$ and $Tv = \lambda_2 v$ form a one-dimensional manifold.*

Case 5. n is even and both eigenvectors are skew-symmetric. When n is even, the Hankel matrix $\tilde{H}(v)$ for a skew-symmetric vector v takes the special form

$$\tilde{H}(v) = \begin{bmatrix} v_1 & v_2 & \cdots & v_{p-1} & v_p \\ v_2 & v_3 & & v_p & -v_p \\ \vdots & & & & \vdots \\ v_{p-1} & v_p & \cdots & -v_4 & -v_3 \\ v_p & -v_p & \cdots & -v_3 & -v_2 \end{bmatrix}. \tag{50}$$

To construct the transformation matrix $\tilde{G}(u, v)$, the matrix $G(v)$ for a skew-symmetric vector v is defined in exactly the same way as (48). The conclusion is stated in Theorem 3.10.

THEOREM 3.10. *Suppose that n is even and that u and v are two skew-symmetric vectors satisfying $u^T v = 0$. Then:*

1. *The dimension of $S(u) \cap S(v)$ is at least three.*
2. *For almost all u and v, the dimension of $S(u) \cap S(v)$ is exactly three.*
3. *For almost all u and v and for any values of λ_1 and λ_2, the symmetric Toeplitz matrices T satisfying $Tu = \lambda_1 u$ and $Tv = \lambda_2 v$ form a one-dimensional manifold.*

Case 6. n is even, one eigenvector is symmetric and the other is skew-symmetric. Just like Case 3, the orthogonality condition does not help to reduce the rank of $\tilde{M}$. The $n \times (n+1)$ matrix $\tilde{M}$ in general is of full rank. The conclusion in Theorem 3.11, therefore, is similar to Theorem 3.8.

THEOREM 3.11. *Suppose that n is even and that u and v are symmetric and skew-symmetric vectors, respectively. Then:*

1. *The dimension of $S(u) \cap S(v)$ is at least two.*
2. *For almost all u and v, the dimension of $S(u) \cap S(v)$ is exactly two.*
3. *For almost all u and v and for any values of λ_1 and λ_2, there exists a unique symmetric Toeplitz matrix T satisfying $Tu = \lambda_1 u$ and $Tv = \lambda_2 v$.*

4. Conclusion. We have shown by symbolic computation that the dimension of the subspace $S(u) \cap S(v)$ of Toeplitz matrices with two generically prescribed eigenvectors u and v is independent of the size of the problem. We have further shown that the dimension is either two, three, or four, depending upon whether the eigenvectors are symmetric or skew-symmetric. All the cases are justified to the extent that the transformation matrices that result in the desired elimination are fully described in terms of the components of u and v. Only one proof (Lemma 3.4) is detailed, but the rest can be done in a very similar way.

Our result extends a previous result by Cybenko [4] who considers the structure of Toeplitz matrices with only one prescribed eigenvector. On the other hand, our discovery that the dimension is independent of the size of the problem is quite a surprising and remarkable fact.

We also have studied the inverse problem of constructing a Toeplitz matrix from two prescribed eigenpairs. We have shown that in almost every direction of $\ker(\tilde{M})$, there is one and only one Toeplitz matrix with the prescribed eigenpairs. In particular, it is shown that if n is odd and if at least one of the given eigenvectors is symmetric, or if n is even and one eigenvector is symmetric and the other is skew-symmetric, then the Toeplitz matrix is unique.

Appendix. In the proof of Theorem 3.5 we need to show that $\det(WW^T)$ is not identically zero. This can be done by simply showing that $\det(WW^T) \neq 0$ for a certain u and v. In particular, choose

$$u_i = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{if } 1 < i \leq p; \end{cases}$$

$$v_i = \begin{cases} 1 & \text{if } i = p, \\ 0 & \text{if } 1 \leq i < p. \end{cases}$$

Then it is easy to see that the $(p-1) \times p$ submatrix W is given by

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & & 2 & 0 & 0 \\ \vdots & \vdots & & & & & \vdots & \vdots \\ 0 & 0 & 0 & 2 & & & 0 & \\ 0 & 0 & 2 & 0 & \dots & & 0 & 0 \\ 1 & 0 & \dots & & \dots & & 0 & 0 \end{bmatrix}.$$

Obviously W is of full rank $p-1$.

With the same choice of u and v, it is also easy to see that

$$x = \left[0, -\sqrt{.5}, 0, \dots, 0, \sqrt{.5}\right]^T$$

is a solution to (18). Specifically, we have proved that x_0 cannot be identical to x_1 for all u and v.

Similar arguments can be deduced for the proof of other cases.

Note added in proof. As the paper was being printed, Landau announced that he has proved that every set of n real numbers is the set of eigenvalues of an $n \times n$ real symmetric Toeplitz matrix. See [12].

REFERENCES

[1] S. L. CAMPBELL AND C. D. MEYER, *Generalized Inverses of Linear Transformations*, Pitman, London, 1979.

[2] A. CANTONI AND P. BUTLER, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl., 13 (1976), pp. 275–288.

[3] M. CHU, *Matrix differential equations: A continuous realization process for linear algebra problems*, Nonlinear Anal., TMA, 18 (1992), pp. 1125–1146.

[4] G. CYBENKO, *On the eigenstructure of Toeplitz matrices*, IEEE Trans. Acoust. Speech Signal Process., ASSP-32 (1984), pp. 918–920.

[5] P. DELSARTE AND Y. GENIN, *Spectral properties of finite Toeplitz matrices*, Proc. 1983 International Symp. Math. Theory Networks Systems, Beer-Sheva, Israel, pp. 194–213.

[6] 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.

[7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The John Hopkins University Press, Baltimore, MD, 1989.

[8] D. P. LAURIE, *A numerical approach to the inverse Toeplitz eigenproblem*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 401–405.

[9] J. MAKHOUL, *On the eigenvectors of symmetric Toeplitz matrices*, IEEE Trans. Acoust. Speech Signal Process., ASSP-29 (1981), pp. 868–872.

[10] W. F. TRENCH, *Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 135–146.

[11] ———, *Spectral evolution of a one-parameter extension of a real symmetric Toeplitz matrix*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 601–611.

[12] H. J. LANDAU, *The inverse eigenvalue problem for real symmetric Toeplitz matrices*, AT&T Bell Laboratories, 1992, preprint.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 636–660, April 1994

018

PERTURBATION ANALYSIS OF A CONDITION NUMBER FOR LINEAR SYSTEMS*

ZHI-QUAN LUO† AND PAUL TSENG‡

Abstract. In 1952, A. J. Hoffman [*J. Res. Natl. Bur. Standards*, 49 (1952), pp. 263–265] published a bound on the distance from any point to the solution set of a linear system. This bound subsequently has found applications in the sensitivity analysis of linear/integer programs and the convergence analysis of descent methods for linearly constrained minimization. A certain constant in Hoffman's bound may be interpreted as a condition number for the linear system and, in this paper, the authors give simple necessary and sufficient conditions for the constant to be uniformly bounded under perturbations on the problem data. Also, these conditions are related to a uniform boundedness condition on the vertex solutions proposed by J.-S. Pang.

Key words. Hoffman's bound, condition number, linear system, perturbation analysis

AMS subject classifications. 90C31, 65F35, 15A39

1. Introduction.

For any $m \times n$ matrix A and any m-vector a ($m \geq 1, n \geq 1$), we denote the polyhedral set

$$P(A, a) = \{ x \in \Re^n \mid Ax \leq a \}.$$

In many practical settings, it is of interest to estimate the Euclidean distance from a point x to its nearest point in $P(A, a)$ when this set is nonempty. One particularly useful estimate is given by A. J. Hoffman [Hof52] who showed that this distance is bounded above by some scalar constant (depending on A only) times the Euclidean norm of the residual error

$$[Ax - a]_+,$$

where, for any vector z, $[z]_+$ denotes the positive part of z. This bound and its relatives have been studied quite extensively and important applications have been found in the sensitivity analysis of linear programs (see [Rob73a], [Rob77]) and in the convergence analysis of descent methods for linearly constrained minimization (see [Gof80], [Gül92], [IuD90], [LuT92], [TsB93], [TsL92]). Moreover, the constant in the bound provides a measure of the "condition" of the linear system $Ax \leq a$ (see [Gof80], [Man81b]).

Since real-world problems typically have inaccurate data, it is of practical as well as theoretical interest to know how Hoffman's bound behaves under perturbations on the constraint matrix A and the right-hand side a. Specifically, when would the constant in this bound be uniformly bounded under such perturbations? This question was posed to us by Professor J.-S. Pang [Pan91] who also conjectured that this uniform boundedness property would hold if and only if $P(A, a)$ contains a Slater point and its vertex set is nonempty and is in some sense uniformly bounded under local

* Received by the editors January 13, 1992; accepted for publication (in revised form) November 8, 1992.

† Room 225, Communications Research Laboratory, McMaster University, Hamilton, Ontario, L8S 4K1, Canada (luozq@mcmail.cis.mcmaster.ca). The research of this author was supported by the Natural Sciences and Engineering Research Council of Canada grant OPG0090391.

‡ Department of Mathematics, GN-50, University of Washington, Seattle, Washington 98195 (tseng@math.washington.edu). The research of this author was supported by National Science Foundation grant CCR-9103804.

perturbations on A and a. Surprisingly, the above question has been little studied and the available results mainly treat the case where a is perturbed.

In this paper, we address the above and related questions regarding the behaviour of Hoffman's bound under perturbations on the problem data. To be precise, let us define, for each $m \times n$ matrix A and m-vector a such that $P(A, a)$ is nonempty, the quantity

$$\tau(A, a) = \sup_{x \notin P(A,a)} \frac{d(x, P(A, a))}{\|[Ax - a]_+\|}. \tag{1.1}$$

(Here and throughout, we denote, for every vector x and every closed set C in some Euclidean space,

$$d(x, C) = \min_{y \in C} \|x - y\|,$$

where $\|y\|$ is the usual Euclidean norm of a vector y.) By the preceding result of Hoffman, $\tau(A, a)$ is finite and, in fact, bounded above by a scalar depending on A only. The quantity $\tau(A, a)$ may be viewed geometrically as a condition number for measuring the "sharpness" of the corners of $P(A, a)$ (see [Gof80]), i.e., the larger $\tau(A, a)$ is, the sharper is some corner of $P(A, a)$ (assuming that each row of A is normalized to have a norm of one). We note that in [Gof80] (see Theorem 4.4 therein), the L_∞-norm is used in place of the Euclidean norm but this difference is minor. For ease of reference, we will refer to $\tau(A, a)$ as the *Hoffman condition number* for the linear system $Ax \leq a$.

We say that the system of linear inequalities $Ax \leq a$ is *well-conditioned* under a set of perturbations on the problem data A and a if $Ax \leq a$ remains solvable and $\tau(A, a)$ is uniformly bounded under perturbations from this set. This criterion of well-conditionedness based on a variational principle is more stringent than the traditional criterion, which only requires that the normalized condition number $\|A\| \cdot \tau(A, a)$ be less than some threshold value. This criterion is also, in some sense, more natural since it takes into account the effects of data perturbation and is not dependent on an (arbitrarily chosen) threshold value. (Like the traditional criterion, it is independent of scaling on A and a (assuming that the set of perturbations is independent of scaling on A and a).)

We will distinguish between the following sets of perturbations on A and a. We say that a set of perturbations is *local* if the size of the perturbations is restricted to be less than some scalar (depending on A and a) and is *global* if no such restriction is made. We say that a set of perturbations is *feasible* if the perturbations are restricted to those that maintain the perturbed system to be solvable. Feasible perturbations are of interest since they arise in various practical contexts. For example, in the classical situation where a nonlinear system is linearized around one of its feasible solutions, any perturbation in the feasible solution would induce a feasible perturbation on the linearized system.

The main goal of this paper is to establish (computationally tractable) necessary and sufficient conditions for a linear system to be well-conditioned under any combination of the preceding set of perturbations. First, we show that the system $Ax \leq a$ is well-conditioned under feasible local perturbations on A and feasible global perturbations on a if and only if every row-wise submatrix of A is either strongly stable or has full column rank (see Theorem 2.2). (A matrix M is said to be strongly stable if $Mx < 0$ has a solution. See [Man81a] and [Bra88] for applications of such matrices

to the stability analysis of linear systems.) Similarly, this system is well-conditioned under feasible local perturbations on A and a if and only if every row-wise submatrix of A possessing a certain structure is either strongly stable or has full column rank (see Theorem 2.4). Second, we show that the system $Ax \leq a$ is well-conditioned under arbitrary local perturbations on A and arbitrary global perturbations on a if and only if A is strongly stable (see Theorem 3.1). Similarly, this system is well-conditioned under arbitrary local perturbations on A and a if and only if either A is strongly stable or the system $Ax \leq a$ is regular and has a bounded solution set (see Theorem 3.3). (A system $Ax \leq a$ is regular if it satisfies the Slater condition that $Ax < a$ be solvable.) The "if" part of these results have already been shown by Brady [Bra88, Chap. 5], but the "only if" part was not known. Third, we relate one of the (algebraic) characterizations of well-conditionedness to a geometric characterization, conjectured by Pang, involving a uniform boundedness condition on the vertex solutions. In particular, we show that the latter condition together with a regularity assumption on the system implies the well-conditionedness of the system under arbitrary local perturbations on the problem data, but the converse does not hold (see §4). Fourth, we show that all of the above results extend to the case where linear equalities are also present in the system (see §5), and we discuss how these results can be applied to determine whether a polyhedral set has a well-conditioned algebraic representation (see §6).

We briefly survey related results. Many estimates of the Hoffman condition number $\tau(A, a)$ have been proposed, beginning with the original work of Hoffman [Hof52] with subsequent refinements made by Robinson [Rob73a], Mangasarian [Man81b], Mangasarian and Shiau [MaS87], Cook et al. [CGST86], and, most recently, by Li [Li91] and Bergthaller and Singer [BeS91]. (Also see the early work of Rosenbloom [Ros51] for a local estimate.) Unfortunately, these estimates typically involve the solution of an optimization problem and are too complicated to be useful for a perturbation analysis. In a different direction, Robinson [Rob75a] (also see [Rob77] and Lemma 3.2 in this paper) showed that the system $Ax \leq a$ remains solvable under local perturbations on A and a if and only if it is regular. This implies that regularity is a necessary condition for a system of linear inequalities to be well-conditioned under (arbitrary) local perturbations on the problem data. However, it is not sufficient (see Theorem 3.3). Another direction in which Hoffman's bound has been extended is the sensitivity analysis of linear systems. In particular, Hoffman's bound implies that a solution of a linear system changes in a Lipschitzian manner relative to changes on the problem data, with the Lipschitz constant depending on the solution. Robinson [Rob75a] (also see [Rob72] and [Rob73b]) showed that the latter dependence can be removed to some extent if the system is regular. Daniel [Dan73], [Dan75] proposed restricting the allowable perturbations as an alternative way to remove the dependence. Further elaborations of Robinson's result are given by Brady [Bra88]. Our results can be applied to sharpen some of the results above although, for brevity, we will not discuss such applications here. Finally, Hoffman's bound has also been extended to convex programs and to systems of convex inequalities. The first result of this kind was given by Robinson [Rob75b] under a Slater condition and assuming boundedness of the solution set. This result was then extended by Mangasarian [Man85] to a system of differentiable convex inequalities in which the boundedness assumption is replaced by an asymptotic constraint qualification. Further extensions were made by Auslender and Crouzeix [AuC88], among which is the release of the differentiability assumption. It is unclear whether our analysis can be extended to the convex case.

We adopt the following notations throughout. All vectors are column vectors and superscript T denotes transpose. For any vector $x \in \Re^n$ and any $I \subseteq \{1, \ldots, n\}$, we

denote by x_i the ith component of x and x_I the vector with components $x_i \in I$ (with the x_i's arranged in the same order as in x). For any two vectors x and y of the same dimension, we denote by $\langle x, y\rangle$ the usual Euclidean inner product of x with y, i.e., $\langle x, y\rangle = \sum_i x_i y_i$. Thus, the Euclidean norm of x is given by $\|x\| = \sqrt{\langle x, x\rangle}$. For any $m \times n$ matrix M and any $I \subseteq \{1, \ldots, m\}$, we denote by M_i the ith row of M and by M_I the submatrix of M obtained by removing all rows $i \notin I$; we denote by $\|M\|$ the matrix norm of M as induced by the vector norm $\|\cdot\|$ (i.e., $\|M\| = \max_{\|x\|=1} \|Mx\|$). We say that a sequence of matrices $\{M^r\}$ converges to M ("$\{M^r\} \to M$" for short) if $\|M^r - M\| \to 0$ as $r \to \infty$. For any $I \subseteq \{1, 2, \ldots\}$, we denote by $|I|$ the number of elements of I. Finally, we say that a set of perturbations on A and a is *semilocal* if the perturbations on A are local but the perturbations on a are global.

2. Well-conditionedness under feasible perturbations In this section we study the well-conditionedness of the system $Ax \leq a$ under feasible perturbations on A and a, i.e., perturbations under which $Ax \leq a$ remains solvable. We divide the analysis into two subsections. In §2.1, we show that this system is well-conditioned under feasible local perturbations on A and feasible global perturbations on a if and only if, for every nonempty $I \subseteq \{1, \ldots, m\}$, the submatrix A_I either is strongly stable or has full column rank (see Theorem 2.2). In §2.2, we show that this system is well-conditioned under feasible local perturbations on A and a if and only if, for every nonempty $I \subseteq \{1, \ldots, m\}$ such that either $Ax \leq a, A_I x = a_I$ or $Ax \leq 0,\ A_I x = 0, x \neq 0$ is solvable, the submatrix A_I either is strongly stable or has full column rank (see Theorem 2.4). Moreover, these characterizations of well-conditionedness simplify considerably under some mild assumptions on the solution set $P(A, a)$ (see Propositions 2.3, 2.5, 2.6).

2.1. Well-conditionedness under feasible semilocal perturbations First we state and prove a key property of matrices that are either strongly stable or of full column rank. This property will be used in the proof of the main result of this subsection, namely, Theorem 2.2.

LEMMA 2.1. *Let E be an $l \times n$ matrix ($l \geq 1$).*

(a) *If there exists an n-vector u with $Eu > 0$, then for every l-vector $\theta \geq 0$, we have*

$$\|[EE^T\theta]_+\| \geq \frac{\min_i\{E_i u\}}{\|u\|}\|E^T\theta\|.$$

(b) *If E has full column rank, then for every l-vector $\theta \geq 0$, there is an $J \subseteq \{1, \ldots, l\}$ with $|J| = n$ and E_J invertible such that*

$$\|[EE^T\theta]_+\| \geq \left(\min_{\|\nu\|=1} \|(E_J)^T\nu\|\right)\|E^T\theta\|.$$

Proof. Fix any l-vector $\theta \geq 0$. If $E^T\theta = 0$, then the claims follow trivially. Thus, it suffices to consider the case where $E^T\theta \neq 0$. Using the Cauchy–Schwarz inequality and the nonnegativity of θ, we have

$$\|\theta\|\|[EE^T\theta]_+\| \geq \langle\theta, [EE^T\theta]_+\rangle \geq \langle\theta, EE^T\theta\rangle = \|E^T\theta\|^2,$$

so dividing both sides by $\|E^T\theta\|\|\theta\|$ gives

$$\frac{\|[EE^T\theta]_+\|}{\|E^T\theta\|} \geq \frac{\|E^T\theta\|}{\|\theta\|}. \tag{2.1}$$

(a) Suppose that there is an n-vector u with $Eu > 0$. Using the Cauchy–Schwarz inequality and the nonnegativity of θ, we have

$$\|u\|\|E^T\theta\| \geq \langle u, E^T\theta\rangle = \langle Eu, \theta\rangle \geq \min_i\{E_iu\}\|\theta\|.$$

Thus, $\|E^T\theta\|/\|\theta\| \geq \min_i\{E_iu\}/\|u\|$. Using this to bound the right-hand side of (2.1) yields the desired inequality.

(b) Suppose that E has rank n. Then, by Carathéodory's Theorem (see [Roc70]), there exists an n-vector $\nu \geq 0$ and a subset $J \subseteq \{1,\ldots,l\}$ with $|J| = n$ and E_J invertible such that $(E_J)^T\nu = E^T\theta$. Since (2.1) holds for any E and any nonnegative vector θ with $E^T\theta \neq 0$, it also holds when E and θ are replaced by, respectively, E_J and ν. This gives

$$\frac{\|[E_J(E_J)^T\nu]_+\|}{\|(E_J)^T\nu\|} \geq \frac{\|(E_J)^T\nu\|}{\|\nu\|}.$$

The right-hand quantity is clearly bounded below by $\min_{\|\nu\|=1}\|(E_J)^T\nu\|$; the left-hand quantity is, using $(E_J)^T\nu = E^T\theta$, equal to $\|[E_JE^T\theta]_+\|/\|E^T\theta\|$ which is clearly bounded above by $\|[EE^T\theta]_+\|/\|E^T\theta\|$. This proves the desired inequality. □

Given below is the main result of this subsection, which establishes a necessary and sufficient condition for the system $Ax \leq a$ to be well-conditioned under feasible local perturbations on A and feasible global perturbations on a. The proof of sufficiency is based on bounding $\tau(A,a)$ by $\|(A_I)^T\lambda\|/\|[A_I(A_I)^T\lambda]_+\|$ for some nonzero vector $\lambda \geq 0$ and some $I \subseteq \{1,\ldots,m\}$. Then, Lemma 2.1 is invoked to bound the latter. The proof of necessity amounts to constructing a set of feasible local perturbations on A and feasible global perturbations on a under which $\tau(A,a)$ is not uniformly bounded (see (2.5) and (2.6)).

THEOREM 2.2. (*Well-conditionedness under feasible semilocal perturbations*). *For any $m \times n$ matrix A, the following conditions are equivalent.*

(a) *For each nonempty index set $I \subseteq \{1,\ldots,m\}$, we have that either $A_Ix < 0$ is solvable or A_I has full column rank.*

(b) *There exist scalars $\delta > 0$ and $\beta > 0$ such that, for any (A',a') with $\|A'-A\| < \delta$ and $A'x \leq a'$ solvable, we have $\tau(A',a') \leq \beta$.*

Proof. (a) $\Rightarrow$ (b). Suppose that condition (a) holds. For each nonempty $I \subseteq \{1,\ldots,m\}$ such that $A_Ix < 0$ is solvable, let u^I be any n-vector with $A_Iu^I > 0$. Define the scalars:

$$\rho_1 = \min_I\left\{\min_{i\in I}\{A_iu^I\}/\|u^I\|\right\}, \qquad \rho_2 = \min_J\left\{\min_{\|\nu\|=1}\|(A_J)^T\nu\|\right\}, \tag{2.2}$$

where the minimization with respect to I is taken over all nonempty $I \subseteq \{1,\ldots,m\}$ for which $A_Ix < 0$ is solvable (with $\rho_1 = \infty$ if no such I exists); the minimization with respect to J is taken over all nonempty $J \subseteq \{1,\ldots,m\}$ for which $|J| = n$ and A_J is invertible (with $\rho_2 = \infty$ if no such J exists). Notice that ρ_1 is positive.

Consider any $m \times n$ matrix A' satisfying

$$\|A' - A\| < \min\{\rho_1,\rho_2\}/2, \tag{2.3}$$

and any m-vector a' for which $A'x \leq a'$ is solvable (so $P(A',a')$ is nonempty). Fix any n-vector $x \notin P(A',a')$ and let z denote the orthogonal projection of x onto $P(A',a')$ (so $\|x-z\| = d(x,P(A',a'))$). Let I denote the nonempty set of indices $i \in \{1,\ldots,m\}$

for which $A'_i z = a'_i$. Then, $A'_I z = a'_I$ and the Kuhn–Tucker conditions associated with this projection yield $x - z = (A'_I)^T \lambda$ for some $\lambda \geq 0$, so

$$\frac{d(x, P(A', a'))}{\|[A'x - a']_+\|} = \frac{\|x - z\|}{\|[A'x - a']_+\|} \leq \frac{\|x - z\|}{\|[A'_I x - a'_I]_+\|} = \frac{\|(A'_I)^T \lambda\|}{\|[A'_I (A'_I)^T \lambda]_+\|}. \tag{2.4}$$

Since condition (a) holds by hypothesis, it suffices to consider only the following two cases.

Case 1. $A_I x < 0$ is solvable. Then, u^I is defined and (cf. $\|A' - A\| < \rho_1/2$ and $A_I u^I > 0$) $A'_I u^I \geq A_I u^I/2 > 0$, so we can apply Lemma 2.1(a) with $E = A'_I$ to obtain

$$\frac{\|(A'_I)^T \lambda\|}{\|[A'_I (A'_I)^T \lambda]_+\|} \leq \frac{\|u^I\|}{\min_{i \in I}\{A'_i u^I\}} \leq \frac{2\|u^I\|}{\min_{i \in I}\{A_i u^I\}} \leq \frac{2}{\rho_1},$$

where the last inequality follows from (2.2).

Case 2. A_I has full column rank. Then, for any subset J of I with $|J| = n$ and A_J invertible, we have from the triangle inequality and (2.2) and (2.3) that

$$\min_{\|\nu\|=1} \|(A'_J)^T \nu\| \geq \min_{\|\nu\|=1} \|(A_J)^T \nu\| - \|(A_J)^T - (A'_J)^T\| \geq \rho_2 - \frac{\rho_2}{2} = \frac{\rho_2}{2}.$$

This shows that A'_J has full column rank, so A'_I also has full column rank. Then, we can apply Lemma 2.1(b) with $E = A'_I$ and, together with the above relation, obtain that

$$\frac{\|(A'_I)^T \lambda\|}{\|[A'_I (A'_I)^T \lambda]_+\|} \leq \frac{1}{\min_{\|\nu\|=1} \|(A'_J)^T \nu\|} \leq \frac{2}{\rho_2},$$

where J is some subset of I with $|J| = n$ and A_J invertible.

Since either the above relation or the relation in Case 1 holds, the right-hand side of (2.4) is bounded above by the maximum of $2/\rho_1$ and $2/\rho_2$. Then, (2.4) yields

$$\frac{d(x, P(A', a'))}{\|[A'x - a']_+\|} \leq \max\left\{\frac{2}{\rho_1}, \frac{2}{\rho_2}\right\}.$$

Our choice of x above was arbitrary, so the above inequality holds for all n-vectors $x \notin P(A', a')$. Taking the supremum of both sides over all $x \notin P(A', a')$ and using (1.1), we obtain $\tau(A', a') \leq \max\{2/\rho_1, 2/\rho_2\}$.

(b) $\Rightarrow$ (a). Suppose that condition (a) does not hold. Then, there exists some nonempty $I \subseteq \{1, \ldots, m\}$ such that $A_I x < 0$ has no solution and A_I lacks full column rank. Fix any m-vector a such that $Ax \leq a$ has a solution and let $\bar{x}$ be any such solution. Since $A_I x < 0$ has no solution, by the Farkas Lemma (see [Roc70] or [Sch86]), there exists a nonzero vector $\theta_I \geq 0$ with $(A_I)^T \theta_I = 0$. Also, since A_I lacks full column rank, there exists a nonzero z with $A_I z = 0$. Normalizing if necessary, we will assume that $\|\theta_I\| = 1$ and $\|z\| = 1$. Using I, θ_I, and z, we construct below a set of local perturbations on A and global perturbations on a such that the perturbed systems are solvable but whose respective Hoffman condition numbers are not uniformly bounded. The key is to perturb each row of A_I by a scalar multiple of z^T and, because z^T is orthogonal to these rows, this will induce in the solution set of the perturbed system a corner whose "sharpness" is inversely proportional to the size of the perturbation.

By reordering the rows of A and a if necessary, we can assume that

$$A = \begin{bmatrix} A_I \\ A_{\bar{I}} \end{bmatrix}, \qquad a = \begin{bmatrix} a_I \\ a_{\bar{I}} \end{bmatrix},$$

where $\tilde{I}$ denotes the complement of I relative to $\{1,\ldots,m\}$. For each scalar $\epsilon > 0$, we define the perturbed matrix A^ϵ by:

$$A_I^\epsilon = A_I + \epsilon\theta_I z^T, \qquad A_{\tilde{I}}^\epsilon = A_{\tilde{I}}, \tag{2.5}$$

and the perturbed right-hand side a^ϵ by:

$$a_I^\epsilon = A_I\bar{x} + \epsilon(\langle z, \bar{x}\rangle + \epsilon)\theta_I, \qquad a_{\tilde{I}}^\epsilon = a_{\tilde{I}} + 2\epsilon[A_{\tilde{I}}z]_+. \tag{2.6}$$

Let $x^\epsilon = \bar{x} + \epsilon z$. We claim that x^ϵ belongs to $P(A^\epsilon, a^\epsilon)$. To see this, notice that

$$\begin{aligned} A_I^\epsilon x^\epsilon &= (A_I + \epsilon\theta_I z^T)(\bar{x} + \epsilon z) \\ &= A_I\bar{x} + \epsilon(\langle z, \bar{x}\rangle + \epsilon)\theta_I \\ &= a_I^\epsilon, \end{aligned} \tag{2.7}$$

where the second equality follows from $A_I z = 0$ and $\|z\| = 1$, and the last equality follows from the definition of a_I^ϵ. Similarly, we have

$$\begin{aligned} A_{\tilde{I}}^\epsilon x^\epsilon &= A_{\tilde{I}}(\bar{x} + \epsilon z) \\ &\le a_{\tilde{I}} + 2\epsilon[A_{\tilde{I}}z]_+ \\ &= a_{\tilde{I}}^\epsilon, \end{aligned} \tag{2.8}$$

where the inequality follows from $A_{\tilde{I}}\bar{x} \le a_{\tilde{I}}$ and $A_{\tilde{I}}z \le 2[A_{\tilde{I}}z]_+$; the last equality is due to the definition of $a_{\tilde{I}}^\epsilon$.

Since $(A_I)^T\theta_I = 0$ and $\|\theta_I\| = 1$, we have from (2.5) that

$$(A_I^\epsilon)^T\theta_I = (A_I + \epsilon\theta_I z^T)^T\theta_I = \epsilon z.$$

This together with $\theta_I \ge 0$ and (2.7)–(2.8) implies that z is a normal to the polyhedral set $P(A^\epsilon, a^\epsilon)$ at x^ϵ. Thus, if we move along the direction z from x^ϵ and then project back onto $P(A^\epsilon, a^\epsilon)$, we always get x^ϵ. In particular, the projection of $x^\epsilon + \epsilon z$ onto $P(A^\epsilon, a^\epsilon)$ is x^ϵ, so that

$$d(x^\epsilon + \epsilon z, P(A^\epsilon, a^\epsilon)) = \epsilon\|z\| = \epsilon. \tag{2.9}$$

On the other hand, we have from the definitions of A^ϵ and a^ϵ (cf. (2.5) and (2.6)) that

$$\begin{aligned} A_I^\epsilon(x^\epsilon + \epsilon z) &= (A_I + \epsilon\theta_I z^T)(\bar{x} + 2\epsilon z) \\ &= A_I\bar{x} + \epsilon(\langle z, \bar{x}\rangle + 2\epsilon)\theta_I \\ &= a_I^\epsilon + \epsilon^2\theta_I, \end{aligned}$$

where the second equality follows from $A_I z = 0$ and $\|z\| = 1$. Similarly, we have

$$\begin{aligned} A_{\tilde{I}}^\epsilon(x^\epsilon + \epsilon z) &= A_{\tilde{I}}(\bar{x} + 2\epsilon z) \\ &\le a_{\tilde{I}} + 2\epsilon[A_{\tilde{I}}z]_+ \\ &= a_{\tilde{I}}^\epsilon, \end{aligned}$$

where the inequality follows from $A_{\bar{I}}\bar{x} \leq a_{\bar{I}}$ and $A_{\bar{I}}z \leq [A_{\bar{I}}z]_+$. Combining the above two relations yields

$$[A^\epsilon(x^\epsilon + \epsilon z) - a^\epsilon]_+ = \begin{bmatrix} \epsilon^2\theta_I \\ 0 \end{bmatrix},$$

so that $\|[A^\epsilon(x^\epsilon + \epsilon z) - a^\epsilon]_+\| = \epsilon^2\|\theta_I\| = \epsilon^2$. This together with (2.9) implies

$$\frac{d(x^\epsilon + \epsilon z, P(A^\epsilon, a^\epsilon))}{\|[A^\epsilon(x^\epsilon + \epsilon z) - a^\epsilon]_+\|} = \frac{\epsilon}{\epsilon^2} = \frac{1}{\epsilon}.$$

Because the right-hand side tends to infinity as $\epsilon \to 0$, it follows (cf. (1.1)) that $\tau(A^\epsilon, a^\epsilon) \to \infty$ as $\epsilon \to 0$. Since $A^\epsilon \to A$ as $\epsilon \to 0$, this shows that condition (b) cannot hold. □

In general, condition (a) in Theorem 2.2 is quite difficult to verify. The next result shows that, in the case where the system $Ax \leq a$ has an unbounded solution set, this condition simplifies considerably.

PROPOSITION 2.3. *For any $m \times n$ matrix A such that $Ax \leq 0$ has a nonzero solution, condition* (a) *in Theorem* 2.2 *holds if and only if the system $Ax < 0$ has a solution.*

Proof. The "if" direction is obvious. To prove the "only if" direction, suppose that condition (a) in Theorem 2.2 holds. By hypothesis, there is a nonzero n-vector x satisfying $Ax \leq 0$. Let I be the set of indices i such that $A_i x = 0$. Then, the submatrix A_I must lack full column rank, so condition (a) in Theorem 2.2 implies that there is an n-vector x' satisfying $A_I x' < 0$. For any scalar $\epsilon > 0$ sufficiently small, we have that $A(x + \epsilon x') < 0$. □

Checking the solvability of $Ax < 0$ amounts to solving the linear program $\min\{\ \mu \in \Re \mid Ax \leq \mu e\ \}$ with e denoting the m-vector of all 1's, for which many efficient algorithms exist. Checking whether $Ax \leq 0$ has a nonzero solution can similarly be accomplished by solving a single linear program, as was pointed out to us by one of the referees and by R. Freund. In particular, one first determines if A has full column rank. If not, then $Ax \leq 0$ trivially has a nonzero solution; if A does have full column rank, then it can be seen that the linear program $\max\{\ \langle e, u\rangle \mid Ax + u \leq 0,\ 0 \leq u \leq e\ \}$ has a positive optimal cost if and only if $Ax \leq 0$ has a nonzero solution (also see [FRT85]). Thus, the conditions in Proposition 2.3 can be checked relatively easily.

2.2. Well-conditionedness under feasible local perturbations. Below we give a necessary and sufficient condition (see condition (a) in Theorem 2.4) for the system $Ax \leq a$ to be well-conditioned under feasible local perturbations on A and a. This condition is similar to condition (a) in Theorem 2.2, but only requires that $A_I x < 0$ be solvable or A_I has full column rank for certain special index sets I (rather than for every I). (Roughly speaking, the special index sets are those I for which, under small perturbations on A and a, the perturbed system can be satisfied with the inequalities indexed by I as equalities. See the proof of (a) $\Rightarrow$ (b) in Theorem 2.4.) The proof techniques are, for the most part, similar to those used in proving Theorem 2.2. The major departure comes in proving the necessity of the condition, in which a new set of "destabilizing" perturbations must be constructed to handle the case where the solution set is unbounded (see (2.11)).

THEOREM 2.4 (*Well-conditionedness under feasible local perturbations*). *For any $m \times n$ matrix A and any m-vector a such that $Ax \leq a$ is solvable, the following conditions are equivalent.*

(a) *For each nonempty index set* $I \subseteq \{1,\ldots,m\}$ *such that either* $Ax \leq a, A_I x = a_I$ *or* $Ax \leq 0, A_I x = 0, x \neq 0$ *is solvable, we have that either* $A_I x < 0$ *is solvable or* A_I *has full column rank.*

(b) *There exist scalars* $\delta > 0$ *and* $\beta > 0$ *such that for any* (A', a') *with* $\|(A', a') - (A, a)\| < \delta$ *and* $A'x \leq a'$ *solvable, we have* $\tau(A', a') \leq \beta$.

Proof. (a) $\Rightarrow$ (b). First, we claim that there exists a scalar $\bar{\rho} > 0$ (depending on A only) such that, for any nonempty $I \subseteq \{1,\ldots,m\}$ and any (A', a') with $\|(A', a') - (A, a)\| < \bar{\rho}$, solvability of the system $A'x \leq a', A'_I x = a'_I$ implies the solvability of either $Ax \leq a, A_I x = a_I$ or $Ax \leq 0, A_I x = 0, x \neq 0$. To see this, suppose the contrary so that, for some nonempty $I \subseteq \{1,\ldots,m\}$ and some sequence $(A^1, a^1), (A^2, a^2), \ldots$ converging to (A, a), there exists a sequence $z^1, z^2, \ldots$ satisfying $A^r z^r \leq a^r$, $A^r_I z^r = a^r_I$ for all r and yet neither $Ax \leq a, A_I x = a_I$ nor $Ax \leq 0, A_I x = 0, x \neq 0$ is solvable. Since $Ax \leq a, A_I x = a_I$ is not solvable, the sequence $\{z^r\}$ must be unbounded (for otherwise any cluster point of $\{z^r\}$ would be a solution to this system). Thus, we have

$$\limsup_{r\to\infty} \frac{A^r z^r}{\|z^r\|} \leq 0, \qquad \lim_{r\to\infty} \frac{A^r_I z^r}{\|z^r\|} = 0,$$

so any cluster point of the bounded sequence $\{z^r/\|z^r\|\}$ solves $Ax \leq 0, A_I x = 0, x \neq 0$. This contradicts the hypothesis that the latter system is unsolvable.

The remainder of the proof is essentially identical to the proof of (a) $\Rightarrow$ (b) in Theorem 2.2, except that in (2.3) we take $\|A' - A\|$ to be, in addition, less than $\bar{\rho}$. The latter, together with condition (a), guarantees that either Case 1 or Case 2 therein holds.

(b) $\Rightarrow$ (a). Suppose that condition (a) does not hold. Then, there exists some nonempty $J \subseteq \{1,\ldots,m\}$ such that either $Ax \leq a$, $A_J x = a_J$ or $Ax \leq 0$, $A_J x = 0$, $x \neq 0$ is solvable, but $A_J x < 0$ is not solvable and A_J lacks full column rank. Since $A_J x < 0$ is not solvable, by the Farkas Lemma (see [Roc70] or [Sch86]), there exists a nonzero vector $\theta_J \geq 0$ with components θ_i, $i \in J$, satisfying $(A_J)^T \theta_J = 0$. Let $I \subseteq J$ be the set of indices i with $\theta_i > 0$ and let θ_I be the vector obtained by removing from θ_J all components θ_i, $i \notin I$. Then, $\theta_I > 0$ and $(A_I)^T \theta_I = 0$. Since $I \subseteq J$, we also have that A_I lacks full column rank and that either $Ax \leq a$, $A_I x = a_I$ or $Ax \leq 0$, $A_I x = 0$, $x \neq 0$ is solvable. We consider the following two cases.

Case 1. $Ax \leq a$, $A_I x = a_I$ is solvable. Then, there is an $\bar{x}$ with $A\bar{x} \leq a$ and $A_I \bar{x} = a_I$. Exactly as in the proof of Theorem 2.2, we construct matrices A^ϵ and vectors a^ϵ according to (2.5) and (2.6), with I, $\bar{x}$ and θ_I as given above, so that $A^\epsilon \to A$ and $\tau(A^\epsilon, a^\epsilon) \to \infty$ as $\epsilon \downarrow 0$. Because $A_I \bar{x} = a_I$, (2.6) implies, in addition, that $a^\epsilon \to a$. Hence condition (b) cannot hold.

Case 2. $Ax \leq a$, $A_I x = a_I$ is not solvable. Then, the system $Ax \leq 0$, $A_I x = 0$ has a nonzero solution. Let z be any such solution. Normalizing if necessary, we can assume that $\|z\| = 1$. Since $Ax \leq a$ is solvable by hypothesis, there exists an n-vector $\bar{x}$ with $A\bar{x} \leq a$. Since $Ax \leq a$, $A_I x = a_I$ is not solvable, we have $A_I \bar{x} \neq a_I$. Using I, z, and $\bar{x}$, we construct below a set of local perturbations on A and a such that the perturbed systems are solvable but whose respective Hoffman condition numbers are not uniformly bounded. As in the proof of Theorem 2.2, the key is to perturb each row of A_I by a scalar multiple of z^T. First, by reordering the rows of A and a if necessary, we will assume that

$$A = \begin{bmatrix} A_I \\ A_{\bar{I}} \end{bmatrix}, \qquad a = \begin{bmatrix} a_I \\ a_{\bar{I}} \end{bmatrix},$$

where $\tilde{I}$ denotes the complement of I relative to $\{1,\dots,m\}$. Correspondingly, we define an m-vector b given by

$$b_I = a_I - A_I\bar{x}, \qquad b_{\tilde{I}} = 0. \tag{2.10}$$

Notice that $b \geq 0$ and $b_I \neq 0$. Fix any scalar $\epsilon > 0$ with $1/\epsilon > \langle z, \bar{x}\rangle$. We associate with ϵ the perturbed matrix

$$A^\epsilon = A + \epsilon b z^T. \tag{2.11}$$

(Thus $A^\epsilon \to A$ as $\epsilon \to 0$.) Let

$$x^\epsilon = \bar{x} + \alpha z, \tag{2.12}$$

with

$$\alpha = 1/\epsilon - \langle z, \bar{x}\rangle. \tag{2.13}$$

We claim that z is a normal to the set $P(A^\epsilon, a)$ at x^ϵ. To see this, notice that

$$\begin{aligned} A^\epsilon x^\epsilon - a &= (A + \epsilon b z^T)(\bar{x} + \alpha z) - a \\ &= A\bar{x} + \epsilon(\langle z, \bar{x}\rangle + \alpha)b + \alpha A z - a \\ &= \begin{bmatrix} (1 - \epsilon(\langle z, \bar{x}\rangle + \alpha))(A_I\bar{x} - a_I) \\ A_{\tilde{I}}\bar{x} - a_{\tilde{I}} + \alpha A_{\tilde{I}} z \end{bmatrix}, \end{aligned}$$

where the last inequality follows from (2.10) and $A_I z = 0$. Since $1 = \epsilon(\langle z, \bar{x}\rangle + \alpha)$ by (2.13) and $Az \leq 0$, we obtain

$$A^\epsilon_I x^\epsilon = a_I, \qquad A^\epsilon_{\tilde{I}} x^\epsilon \leq a_{\tilde{I}}. \tag{2.14}$$

Thus, the first $|I|$ inequalities of $A^\epsilon x \leq a$ are binding at x^ϵ. Also, we have from (2.11) and $(A_I)^T\theta_I = 0$ that

$$(A^\epsilon_I)^T\theta_I = (A_I)^T\theta_I + \epsilon z\langle b_I, \theta_I\rangle = \epsilon\langle b_I, \theta_I\rangle z.$$

Since b_I is nonnegative but nonzero and $\theta_I > 0$, we have $\langle b_I, \theta_I\rangle > 0$ so z is a positive combination of the columns of $(A^\epsilon_I)^T$. This together with (2.14) shows that z is a normal to the set $P(A^\epsilon, a)$ at x^ϵ so if we move along the direction z from x^ϵ and then project back onto $P(A^\epsilon, a)$, we always get x^ϵ. In particular, the projection of $x^\epsilon + \alpha z$ onto $P(A^\epsilon, a)$ is x^ϵ, which implies

$$d(x^\epsilon + \alpha z, P(A^\epsilon, a)) = \alpha\|z\| = \alpha. \tag{2.15}$$

On the other hand, we have from using (2.14) and (cf. (2.11), $\|z\| = 1$ and $A_I z = 0$) $A^\epsilon_I z = \epsilon b_I$, and (cf. (2.11), (2.10), and $A_{\tilde{I}} z \leq 0$) $A^\epsilon_{\tilde{I}} z \leq 0$ that

$$\|[A^\epsilon(x^\epsilon + \alpha z) - a]_+\| = \left\| \begin{bmatrix} \alpha\epsilon b_I \\ 0 \end{bmatrix}_+ \right\|. \tag{2.16}$$

As $\epsilon \to 0$, we have $\alpha\epsilon \to 1$ (cf. (2.13)) so the right-hand side of (2.16) is bounded above, implying that the left-hand side of (2.16) is bounded above. On the other

hand, we also have $\alpha \to \infty$ (cf. (2.13)), so the left-hand side of (2.15) tends to ∞. Combining these two observations yields

$$\frac{d(x^\epsilon + \alpha z, P(A^\epsilon, a))}{\|[A^\epsilon(x^\epsilon + \alpha z) - a]_+\|} \to \infty,$$

so (cf. (1.1)) $\tau(A^\epsilon, a) \to \infty$, as $\epsilon \to 0$. Thus, condition (b) cannot hold. □

As with Theorem 2.2, Theorem 2.4 is not a practical result since condition (a) therein is very difficult to verify. Nonetheless, the following proposition shows that in the case where the system $Ax \leq a$ has multiple solutions, this latter condition simplifies considerably.

PROPOSITION 2.5. *For any $m \times n$ matrix A and any m-vector a such that the system $Ax \leq a$ has at least two solutions, condition* (a) *in Theorem* 2.4 *holds if and only if either $Ax < 0$ has a solution or $Ax \leq a$ is regular (i.e., there exists an x with $Ax < a$) and has a bounded solution set.*

Proof. First, we prove the "if" part. If $Ax < 0$ has a solution, then condition (a) in Theorem 2.4 holds trivially. Suppose instead that the system $Ax \leq a$ is regular and has a bounded solution set. Then, $A\bar{x} < a$ for some $\bar{x}$ and, for every nonempty $I \subseteq \{1, \ldots, m\}$, the system $Ax \leq 0, A_I x = 0, x \neq 0$ has no solution. On the other hand, for every nonempty $I \subseteq \{1, \ldots, m\}$ such that $Ax \leq a, A_I x = a_I$ has a solution, say z, we have $A_I(\bar{x} - z) < 0$ and so the system $A_I x < 0$ is solvable. This shows that condition (a) in Theorem 2.4 holds.

Next, we prove the "only if" part. Suppose that condition (a) in Theorem 2.4 holds. We first show that the system $Ax \leq a$ is regular. Let y and z be two distinct solutions of $Ax \leq a$. Let $\bar{x} = (y + z)/2$ and let I be the set of indices i such that $A_i y = A_i z = a_i$. Then, $A_I \bar{x} = a_I$ and $A_i \bar{x} < a_i$ for all $i \notin I$. Thus, if I is empty, then $\bar{x}$ solves $Ax < a$. Otherwise, we have from $A_I(y - z) = 0$ that A_I lacks full column rank, so condition (a) in Theorem 2.4 implies that $A_I u < 0$ for some u. Then, we have $A(\bar{x} + \epsilon u) < a$ for any scalar $\epsilon > 0$ sufficiently small. Hence, in either case the system $Ax \leq a$ is regular.

If the system $Ax \leq a$ has a bounded solution set, then the "only if" part immediately follows. Otherwise, there is a nonzero n-vector z satisfying $Az \leq 0$. If $Az < 0$, then again we are done. Otherwise, let I denote the set of indices i for which $A_i z = 0$. Then, z solves $Ax \leq 0, A_I x = 0$, so condition (a) in Theorem 2.4 implies that $A_I x < 0$ has a solution, say z'. (A_I cannot have full column rank since $A_I x = 0$.) It is easily seen that $A(z + \epsilon z') < 0$ for any scalar $\epsilon > 0$ sufficiently small, so the "only if" part follows. □

In the case where $Ax \leq a$ has a unique solution, condition (a) in Theorem 2.4 can again be simplified, albeit only slightly.

PROPOSITION 2.6. *For any $m \times n$ matrix A and any m-vector a such that the system $Ax \leq a$ has a unique solution, say x^*, condition* (a) *in Theorem* 2.4 *holds if and only if, for each nonempty index set $I \subseteq \{1, \ldots, m\}$ with $A_I x^* = a_I$, either $A_I x < 0$ is solvable or A_I has full column rank.*

Proof. Since $Ax \leq a$ has a unique solution, the system $Ax \leq 0, A_I x = 0$ cannot have a nonzero solution for any nonempty $I \subseteq \{1, \ldots, m\}$. Thus, in condition (a) of Theorem 2.4 it suffices to consider only those nonempty I for which $Ax \leq a$, $A_I x = a_I$ is solvable or, equivalently, $A_I x^* = a_I$. □

Although the characterization of well-conditionedness under feasible local perturbations is weaker than that under feasible semilocal perturbations (compare condition (a) in Theorems 2.2 and 2.4, respectively), the two characterizations are, surprisingly,

equivalent in some sense. In particular, notice that when $a = 0$, condition (a) in Theorem 2.4 reduces to condition (a) in Theorem 2.2 (since, for every nonempty $I \subseteq \{1, \ldots, m\}$, the system $Ax \leq a$, $A_I x = a_I$ has a solution, namely, the zero vector). Then, we immediately obtain from Theorems 2.2 and 2.4 the following result.

COROLLARY 2.7. *For any $m \times n$ matrix A, the system $Ax \leq a$ is well-conditioned under feasible local perturbations on A and feasible global perturbations on a if and only if it is well-conditioned under feasible local perturbations on both A and a when $a = 0$.*

3. Well-conditionedness under arbitrary perturbations. In this section, we continue with our study of the well-conditionedness of a system of linear inequalities under perturbations on the problem data. In contrast to the previous section, we no longer restrict the perturbations to be feasible and, as a consequence, the characterizations of well-conditionedness simplify significantly. In particular, we show that the system $Ax \leq a$ is well-conditioned under local perturbations on A and global perturbations on a if and only if A is strongly stable. Similarly, we show that this system is well-conditioned under local perturbations on A and a if and only if either A is strongly stable or $Ax \leq a$ is regular and has a bounded solution set. The proof of these results is, in great part, based on the results from the previous section.

Below we give the first main result of this section, characterizing when the system $Ax \leq a$ is well-conditioned under local perturbations on A and global perturbations on a. Its proof is based on Theorem 2.2 and the well-known fact that $Ax \leq a$ is solvable for all a if and only if A is strongly stable.

THEOREM 3.1 (*Well-conditionedness under semilocal perturbations*). *For any $m \times n$ matrix A, the following conditions are equivalent.*

(a) *$Ax < 0$ is solvable.*

(b) *There exist scalars $\delta > 0$ and $\beta > 0$ such that, for any (A', a') with $\|A' - A\| < \delta$, the system $A'x \leq a'$ is solvable and $\tau(A', a') \leq \beta$.*

Proof. (a) $\Rightarrow$ (b). Suppose that condition (a) holds. Then, condition (a) in Theorem 2.2 clearly holds, so, by Theorem 2.2, there exist scalars $\delta_1 > 0$ and $\beta > 0$ such that, for any (A', a') with $\|A' - A\| < \delta_1$ and $A'x \leq a'$ solvable, we have $\tau(A', a') \leq \beta$. Let z be any fixed n-vector satisfying $Az < 0$. Then, for any A' with $\|A' - A\| < \delta_2 = \min_i\{-A_i z\}/(2\|z\|)$, we have $A'z < 0$, so $A'x \leq a'$ is solvable (since it contains αz for any α sufficiently positive). Take δ to be the minimum of δ_1 and δ_2.

(b) $\Rightarrow$ (a). Suppose that condition (a) does not hold, so the system $Ax < 0$ has no solution. Then, for any m-vector a' with $a' < 0$, the system $Ax \leq a'$ also has no solution, so condition (b) cannot hold. □

As was noted in the introduction, the (a) $\Rightarrow$ (b) part of Theorem 3.1 is not new and was proved by Brady (see [Bra88, Thm. 5.4.13]), complete with estimates of the scalars δ and β in condition (b). We have included this part for completeness.

Although Theorem 3.1 treats the case where a is perturbed globally, it also offers some insight into the case where a is perturbed only locally. Suppose that condition (a) in Theorem 3.1 fails, so there exists a nonzero m-vector $\theta \geq 0$ with $A^T\theta = 0$. Then, if $\langle a, \theta \rangle \leq 0$, a local perturbation of a can result in $\langle a, \theta \rangle < 0$ in which case (cf. the Farkas Lemma) $Ax \leq a$ has no solution. Thus, the only unresolved case occurs when $\langle a, \theta \rangle > 0$ and this is the case on which we focus below. For this purpose, we need the following result of Robinson [Rob75a, Thm. 3] (also see [Rob77, Lemma 3] for a simpler version of the result) which uses the Slater condition to characterize solvability of the system $Ax \leq a$ under local perturbations on A and a.

LEMMA 3.2. *For any $m \times n$ matrix A and m-vector a, the following conditions are equivalent.*

(a) *The system $Ax \leq a$ is regular (i.e., there exists an x with $Ax < a$).*

(b) *There exists a scalar $\delta > 0$ such that for any (A', a') with $\|(A', a') - (A, a)\| < \delta$, the system $A'x \leq a'$ is solvable.*

By combining Lemma 3.2 with Theorems 2.4 and 3.1, we obtain the second main result of this section, characterizing when the system $Ax \leq a$ is well-conditioned under local perturbations on A and a.

THEOREM 3.3 (*Well-conditionedness under local perturbations*). *For any $m \times n$ matrix A and any m-vector a, the following conditions are equivalent.*

(a) *Either $Ax < 0$ is solvable or $Ax \leq a$ is regular and has a bounded solution set.*

(b) *There exist scalars $\delta > 0$ and $\beta > 0$ such that for any (A', a') with $\|(A', a') - (A, a)\| < \delta$, the system $A'x \leq a'$ is solvable and $\tau(A', a') \leq \beta$.*

Proof. (a) $\Rightarrow$ (b). Suppose that condition (a) holds. If $Ax < 0$ is solvable, then condition (b) holds by Theorem 3.1. Otherwise, suppose that $Ax \leq a$ is regular and has a bounded solution set. Let $\bar{x}$ be any n-vector satisfying $A\bar{x} < a$. We now verify that condition (a) in Theorem 2.4 holds. Since $Ax \leq a$ has a bounded solution set, it suffices to verify that, for each nonempty $I \subseteq \{1, \ldots, m\}$ such that $Ax \leq a$, $A_I x = a_I$ is solvable, we have that $A_I x < 0$ is solvable. For any such I and any x satisfying $A_I x = a_I$, we have $A_I(\bar{x} - x) < 0$, so $A_I x < 0$ is solvable. Hence condition (a) in Theorem 2.4 holds. Then, by Theorem 2.4, there exist scalars $\delta_1 > 0$ and $\beta > 0$ such that for any (A', a') with $\|(A', a') - (A, a)\| < \delta_1$ and $A'x \leq a'$ solvable, we have $\tau(A', a') \leq \beta$. Since $Ax < a$ is regular, Lemma 3.2 implies that there is a scalar $\delta_2 > 0$ such that $A'x \leq a'$ is solvable for all (A', a') with $\|(A', a') - (A, a)\| < \delta_2$. Take δ to be the minimum of δ_1 and δ_2.

(b) $\Rightarrow$ (a). Suppose that condition (a) does not hold, so $Ax < 0$ is not solvable and the system $Ax \leq a$ either is not regular or has an unbounded solution set. If $Ax \leq a$ is not regular, then condition (b) cannot hold by Lemma 3.2. Thus, it suffices to show that, when $Ax \leq a$ is regular and has an unbounded solution set, condition (b) cannot hold. Let z be any nonzero n-vector satisfying $Az \leq 0$. Since $Ax < 0$ is not solvable, we have, via the Farkas Lemma, that there exists a nonzero m-vector $\theta \geq 0$ with $A^T\theta = 0$. Let I denote the set of indices i for which $\theta_i > 0$. Since $0 = \langle z, A^T\theta \rangle = \langle A_I z, \theta_I \rangle$, we obtain from $A_I z \leq 0$ and $\theta_I > 0$ that $A_I z = 0$, so A_I lacks full column rank. In addition, since $(A_I)^T\theta_I = 0$, the Farkas Lemma states that $A_I x < 0$ is not solvable. Thus, we have found a nonempty $I \subseteq \{1, \ldots, m\}$ such that $Ax \leq 0$, $A_I x = 0$ has a nonzero solution but $A_I x < 0$ is not solvable and A_I lacks full column rank. Then, condition (a) in Theorem 2.4 does not hold, so, by Theorem 2.4, condition (b) therein does not hold. Since the system $Ax \leq a$ is regular so, by Lemma 3.2, condition (b) in Theorem 2.4 holds if and only if condition (b) holds, this shows that condition (b) cannot hold. □

We close this section with a comparison of the characterization of well-conditionedness under feasible perturbations on the problem data (condition (a) in Theorems 2.2 and 2.4) and under arbitrary perturbations on the problem data (condition (a) in Theorems 3.1 and 3.3). Clearly, if condition (a) in Theorem 3.1 (respectively, Theorem 3.3) holds, then condition (a) in Theorem 2.2 (respectively, Theorem 2.4) holds. Surprisingly, the converse also holds for certain linear systems. In particular, if the system $Ax \leq 0$ has a nonzero solution, then Proposition 2.3 shows that condition (a) in Theorem 2.2 holds if and only if condition (a) in Theorem 3.1 holds. In

other words, a system of linear inequalities having an unbounded solution set is well-conditioned under arbitrary semilocal perturbations on the problem data if and only if it is well-conditioned under feasible semilocal perturbations on the problem data. Similarly, if the system $Ax \leq a$ has multiple solutions, then Proposition 2.5 shows that condition (a) in Theorem 2.4 holds if and only if condition (b) in Theorem 3.3 holds. In other words, for a system of linear inequalities having multiple solutions, the release of the feasibility restriction on the local perturbations does not affect whether the system is well-conditioned.

4. Relation to uniform boundedness of vertex solutions. As was pointed out in §1, the Hoffman condition number $\tau(A, a)$ may be interpreted as a measure of the "sharpness" of the corners (or, more formally, the tangent cones) of the solution set for $Ax \leq a$. Thus, the well-conditionedness of the system $Ax \leq a$ may be interpreted geometrically as a uniform boundedness condition on the sharpness of the corners of its solution set (under perturbations on A and a). What other geometrical conditions are related to well-conditionedness? J.-S. Pang [Pan91] conjectured that the uniform boundedness of the vertex solutions may be one such condition. Intuitively, Pang's conjecture is reasonable since, in the case where the solution set is full dimensional, a corner can be unbounded in sharpness only if its vertex is unbounded. In this section, we verify that the preceding intuition is essentially sound. In particular, we show, by using Theorem 3.3, that Pang's uniform boundedness of vertices condition, together with some mild assumptions on the system, implies the well-conditionedness of the system under local perturbations on the problem data. On the other hand, we show that the implication does not go in the other direction and we indicate exactly where the failure occurs.

We say that a system $Ax \leq a$ satisfies the *uniform boundedness of the vertex solutions* (u.b.v.) condition if there exist scalars $\delta > 0$ and $\beta > 0$ such that, for any (A', a') with $\|(A', a') - (A, a)\| < \delta$, the vertices of $P(A', a')$, if any exist, all have Euclidean norm less than β. Notice that $Ax \leq a$ need not have a vertex solution to satisfy the u.b.v. condition, so long as none of the vertex solutions created by perturbation goes off to infinity.

The u.b.v. condition may be viewed as an alternative criterion for a system of linear inequalities to be well-conditioned in a variational sense. This condition, as defined above, is geometrical in description. Below we give an equivalent algebraic description of this condition which, coupled with Theorem 3.3, will enable us to show that the u.b.v. condition, together with some mild assumptions on the system, implies the well-conditionedness of the system $Ax \leq a$ under local perturbations on A and a.

THEOREM 4.1. *For any $m \times n$ matrix A and any m-vector a such that the system $Ax \leq a$ is regular and has a vertex solution, the following conditions are equivalent.*

(a) *There exists a nonzero n-vector z such that $Az \leq 0$ and the rows of A_I are linearly dependent, where $I = \{\, i \in \{1, \ldots, m\} \mid A_i z = 0 \,\}$.*

(b) *$Ax \leq a$ does not satisfy the u.b.v. condition.*

Proof. (b) $\Rightarrow$ (a). Suppose that condition (b) holds. Then, there exists a sequence $A^1, A^2, \ldots$ converging to A, a sequence $a^1, a^2, \ldots$ converging to a, and a sequence $x^1, x^2, \ldots$ with $\|x^r\| \to \infty$ and x^r being a vertex of $P(A^r, a^r)$ for all r. Thus, $A^r x^r \leq a^r$ for all r and, by passing into a subsequence if necessary, we can assume that there is a nonempty $I \subseteq \{1, \ldots, m\}$ of size at least n such that $A^r_I x^r = a^r_I$ for all r. This together with $\|x^r\| \to \infty$ implies

$$\limsup_{r \to \infty} \frac{A^r x^r}{\|x^r\|} \leq 0, \qquad \lim_{r \to \infty} \frac{A^r_I x^r}{\|x^r\|} = 0,$$

so, upon letting z be any cluster point of $\{x^r/\|x^r\|\}$ and using $A^r \to A$, we obtain $Az \leq 0$ and $A_I z = 0$. Since $z \neq 0$, A_I must have rank strictly less than n. Since A_I has at least n rows, this implies the rows must be linearly dependent.

(a) $\Rightarrow$ (b). Suppose that condition (a) holds, so there exists a nonzero n-vector z such that $Az \leq 0$ and the rows of A_I are linearly dependent, where $I = \{\, i \in \{1,\ldots,m\} \mid A_i z = 0 \,\}$. First, we claim that we can, without loss of generality, assume that A_I has a rank of exactly $n-1$ (so $|I| \geq n$). We argue this constructively. Since $Ax \leq a$ has a vertex solution, the polyhedral cone $\{\, x \in \Re^n \mid Ax \leq 0 \,\}$ cannot contain a straight line. Thus, this cone can be generated by a finite number of extreme rays, which we denote by $z^1,\ldots,z^k$ (see [Sch86, §8.8]). Since z clearly belongs to this cone, we can express it as $z = \sum_{i=1}^k \theta_i z^i$, for some nonnegative scalars $\theta_1,\ldots,\theta_k$, not all zero. Fix any index j with $\theta_j > 0$. Then, $0 = A_I z = \sum_{i=1}^k \theta_i A_I z^i \leq \theta_j A_I z^j \leq 0$, so that $A_I z^j = 0$. Let J denote the set of indices i for which $A_i z^j = 0$. Then, $I \subseteq J$. Since z^j is an extreme ray, the rank of A_J is exactly $n-1$ (see [Sch86, §8.7]). Now replace z by z^j and I by J.

Normalizing if necessary, we will assume that $\|z\| = 1$. Also, since $Ax \leq a$ is regular, there exists an n-vector $\bar{x}$ with $A\bar{x} < a$. Finally, since the rows of A_I are linearly dependent, there exist an index $j \in I$ and scalars γ_i, $i \in I$ with $i \neq j$, such that

$$A_j = \sum_{i \in I, i \neq j} \gamma_i A_i. \tag{4.1}$$

Then, we perturb A and a almost exactly as in the proof of Theorem 2.4 with z, $\bar{x}$, and I as given above, except for a slight difference in the jth row. More precisely, let b be given by (2.10) and, for each scalar $\epsilon > 0$ with $1/\epsilon > \langle z, \bar{x}\rangle$, we define x^ϵ by (2.12)–(2.13) and A^ϵ and a^ϵ by

$$A^\epsilon = A + \epsilon b z^T + \epsilon^2 e^j z^T, \qquad a^\epsilon = a + \epsilon e^j, \tag{4.2}$$

where e^j denotes the m-vector whose jth component is 1 and whose other components are all 0. Then, it is readily seen that, as $\epsilon \to 0$, we have $A^\epsilon \to A$, $a^\epsilon \to a$ and (cf. (2.12) and (2.13)) $\|x^\epsilon\| \to \infty$. Thus it only remains to show that x^ϵ is a vertex solution of $A^\epsilon x \leq a^\epsilon$ for all $\epsilon > 0$ sufficiently small. Now, it can be verified by following the proof of Theorem 2.4 (compare with (2.14)) that

$$A^\epsilon_I x^\epsilon = a^\epsilon_I, \qquad A^\epsilon_{\tilde{I}} x^\epsilon \leq a^\epsilon_{\tilde{I}}$$

for all $\epsilon > 0$, so it suffices to show that A^ϵ_I has rank n for all $\epsilon > 0$ sufficiently small. It is straightforward to verify, using (4.1) and (4.2), that

$$A^\epsilon_j - \sum_{i \in I, i \neq j} \gamma_i A^\epsilon_i = \epsilon \left[\epsilon + \left(b_j - \sum_{i \in I, i \neq j} \gamma_i b_i \right) \right] z^T$$

for all $\epsilon > 0$. Since the quantity inside the brackets is nonzero for all ϵ sufficiently small, this implies that z^T is in the space spanned by the rows of A^ϵ_I for all $\epsilon > 0$ sufficiently small. Since A^ϵ_I is obtained from A_I by adding to every row some scalar multiple of z^T (cf. (4.2)), this implies that the rows of A_I are also in the space spanned by the rows of A^ϵ_I for all $\epsilon > 0$ sufficiently small. Since the rows of A_I together with z^T have rank n, this completes the proof. □

Combining Theorems 3.3 and 4.1, we obtain the following corollary showing that the u.b.v. condition, together with regularity and the existence of a vertex solution, implies the well-conditionedness of the system $Ax \leq a$ under local perturbations on A and a.

COROLLARY 4.2. *For any $m \times n$ matrix A and any m-vector a, if the system $Ax \leq a$ is regular, has a vertex solution, and satisfies the u.b.v. condition, then condition* (b) *in Theorem* 3.3 *holds.*

Proof. Since $Ax \leq a$ is regular, by Theorem 3.3, it suffices to show that either there exists an n-vector x with $Ax < 0$ or there does not exist a nonzero n-vector z with $Az \leq 0$. We argue by contradiction. Suppose the contrary, so there exists a nonzero m-vector $\theta \geq 0$ with $A^T\theta = 0$ (cf. the Farkas Lemma) and a nonzero n-vector z with $Az \leq 0$. Let I denote the set of indices i with $\theta_i > 0$. Since $0 = \langle z, A^T\theta \rangle = \langle A_I z, \theta_I \rangle$, we obtain from $A_I z \leq 0$ and $\theta_I > 0$ that $A_I z = 0$. Also, since $(A_I)^T\theta = 0$, the rows of A_I are linearly dependent. Then, by Theorem 4.1, the system $Ax \leq a$ does not satisfy the u.b.v. condition, a contradiction. □

Theorem 4.1 and Corollary 4.2 help to pinpoint where the u.b.v. condition fails to be necessary for the well-conditionedness of the system $Ax \leq a$ under local perturbations on A and a. First, this system can be well-conditioned under such perturbations without having a vertex solution. (Consider the system $x_1 + 0 \cdot x_2 \leq 0$.) Second, even when this system is regular and has a vertex solution, it can be well-conditioned under such perturbations without satisfying the u.b.v. condition. From Theorems 3.3 and 4.1 we see that this would happen precisely when, for every z specified by condition (a) in Theorem 4.1, the rows of A_I fail to contain the origin in their convex hull, where $I = \{ i \in \{1, \ldots, m\} \mid A_i z = 0 \}$. An example of such a system is $x_1 \leq 1,\ x_1 \leq 0,\ x_2 \leq 0$.

5. Systems with equalities and inequalities. In this section, we extend the results of previous sections to linear systems in which equalities are also present. In particular, we consider a system of the form

$$Ax \leq a, \quad Bx = b, \tag{5.1}$$

where A is an $m \times n$ matrix ($m \geq 1,\ n \geq 1$), B is an $l \times n$ matrix ($l \geq 1$), a is an m-vector and b is an l-vector. Accordingly, we let

$$P(A, B, a, b) = \{ x \in \Re^n \mid Ax \leq a, Bx = b \},$$

and define the Hoffman condition number for (5.1) as the quantity

$$\tau(A, B, a, b) = \sup_{x \notin P(A,B,a,b)} \frac{d(x, P(A, B, a, b))}{\left\| \begin{matrix} [Ax - a]_+ \\ Bx - b \end{matrix} \right\|}, \tag{5.2}$$

whenever $P(A, B, a, b)$ is nonempty. Analogous to the case when only linear inequalities are present, we are interested in finding necessary and sufficient conditions for (5.1) to be well-conditioned (or, equivalently, $\tau(A, B, a, b)$ to be uniformly bounded) under a given set of perturbations on A, B, a, and b. We show below that the results from the previous sections extend in a fairly straightforward manner to this general situation.

The idea of our analysis is to reduce the system (5.1) to one without linear equalities, at which time the results of §§2 and 3 can be applied. To motivate our

reduction, suppose that B has full row rank. It will turn out that, excepting some simple cases, this is a necessary assumption for well-conditionedness (see Lemma 5.2 and the proof of Theorem 5.5). Then, by rearranging columns if necessary, we can write

$$A = [A_r \quad A_s], \qquad B = [B_r \quad B_s], \tag{5.3}$$

for some matrices A_r, A_s, B_r, and B_s with B_r invertible. We partition

$$x = \begin{bmatrix} x_r \\ x_s \end{bmatrix},$$

accordingly. Then, by using equation $Bx = b$ to eliminate x_r, we see that an $\bar{x}$ solves the system (5.1) if and only if $\bar{x}_s$ solves the reduced system

$$(A_s - A_r(B_r)^{-1}B_s)x_s \leq a - A_r(B_r)^{-1}b, \tag{5.4}$$

and $\bar{x}_r = (B_r)^{-1}(b - B_s\bar{x}_s)$. Below we show that regularity conditions and the Hoffman condition number for the two systems (5.1) and (5.4) are intimately related. By exploiting this relationship, we can then extend the results of §§2 and 3 to the general system (5.1).

LEMMA 5.1. *Consider any $m \times n$ matrix A and any $l \times n$ matrix B with full row rank. Partition A and B according to* (5.3) *with B_r invertible. Let $\bar{A} = A_s - A_r(B_r)^{-1}B_s$ and $\bar{a} = a - A_r(B_r)^{-1}b$. Then, the following results hold.*

(a) *For any m-vector a and l-vector b and any (possibly empty) index set $I \subseteq \{1, \ldots, m\}$, the system $Ax \leq a$, $A_I x = a_I$, $Bx = b$ has a solution if and only if the system $\bar{A}x_s \leq \bar{a}$, $\bar{A}_I x_s = \bar{a}_I$ has a solution. The same holds when all inequalities are replaced by strict inequalities.*

(b) *For any (possibly empty) index set $I \subseteq \{1, \ldots, m\}$, the system $Ax \leq 0$, $A_I x = 0$, $Bx = 0$ has a nonzero solution if and only if the reduced system $\bar{A}x_s \leq 0$, $\bar{A}_I x_s = 0$ has a nonzero solution.*

(c) *For any (possibly empty) index set $I \subseteq \{1, \ldots, m\}$, the system $A_I x < 0$, $Bx = 0$ has a solution if and only if the reduced system $\bar{A}_I x_s < 0$ has a solution.*

(d) *For any (possibly empty) index set $I \subseteq \{1, \ldots, m\}$, the matrix*

$$\begin{bmatrix} A_I \\ B \end{bmatrix}$$

has full column rank if and only if $\bar{A}_I$ has full column rank.

(e) *There holds*

$$\frac{1}{\|(B_r)^{-1}B_s\| + 1}\left[\frac{\tau(A, B, a, b)}{2(1 + \|A_r(B_r)^{-1}\|)} - \|(B_r)^{-1}\|\right] \leq \tau(\bar{A}, \bar{a}) \leq \tau(A, B, a, b). \tag{5.5}$$

Proof. (a) It is easily verified using (5.3) and the definition of $\bar{A}$ and $\bar{a}$ that the vector

$$\begin{bmatrix} \bar{x}_r \\ \bar{x}_s \end{bmatrix}$$

solves the system $Ax \leq a$, $A_I x = a_I$, $Bx = b$ if and only if $\bar{x}_s$ solves the system $\bar{A}x_s \leq \bar{a}$, $\bar{A}_I x_s = \bar{a}_I$, and $\bar{x}_r = (B_r)^{-1}(b - B_s\bar{x}_s)$. The same argument applies when the inequalities in the systems are all replaced by strict inequalities.

(b) and (c) The argument is entirely analogous to that for part (a).

(d) It is easily seen using (5.3) and the definition of $\bar{A}$ that the vector

$$\begin{bmatrix} \bar{x}_r \\ \bar{x}_s \end{bmatrix}$$

solves the system $A_I x = 0$, $Bx = 0$ if and only if $\bar{x}_s$ solves the system $\bar{A}_I x_s = 0$ and $\bar{x}_r = -(B_r)^{-1}B_s\bar{x}_s$. Thus, one system has a nonzero solution if and only if the other does.

(e) Fix any n-vector

$$x = \begin{bmatrix} x_r \\ x_s \end{bmatrix} \notin P(A, B, a, b).$$

By letting $\bar{x}_s$ be the element of $P(\bar{A}, \bar{a})$ nearest to x_s and using the fact that

$$\begin{bmatrix} (B_r)^{-1}(b - B_s\bar{x}_s) \\ \bar{x}_s \end{bmatrix}$$

solves $Ax \leq a, Bx = b$, we obtain

$$\begin{aligned} d(x, P(A, B, a, b)) &\leq \left\| \begin{matrix} x_r - (B_r)^{-1}(b - B_s\bar{x}_s) \\ x_s - \bar{x}_s \end{matrix} \right\| \\ &= \left\| \begin{matrix} (B_r)^{-1}(Bx - b + B_s(\bar{x}_s - x_s)) \\ x_s - \bar{x}_s \end{matrix} \right\| \\ &\leq \|(B_r)^{-1}\|\|Bx - b\| + (\|(B_r)^{-1}B_s\| + 1)\|\bar{x}_s - x_s\| \\ &= \|(B_r)^{-1}\|\|Bx - b\| + (\|(B_r)^{-1}B_s\| + 1)d(x_s, P(\bar{A}, \bar{a})). \end{aligned} \tag{5.6}$$

Also, simple algebra using (5.3) gives

$$Ax - a + A_r(B_r)^{-1}(b - Bx) = (A_s - A_r(B_r)^{-1}B_s)x_s - (a - A_r(B_r)^{-1}b) = \bar{A}x_s - \bar{a},$$

so that

$$\begin{aligned} \|[\bar{A}x_s - \bar{a}]_+\| &= \|[Ax - a + A_r(B_r)^{-1}(b - Bx)]_+\| \\ &\leq \|[Ax - a]_+\| + \|A_r(B_r)^{-1}\|\|Bx - b\|. \end{aligned}$$

Adding $\|Bx - b\|$ to both sides and using the fact $\|\alpha\| + \|\beta\| \leq \sqrt{2}\left\|\begin{matrix}\alpha \\ \beta\end{matrix}\right\|$ for any two vectors α and β, we obtain

$$\|Bx - b\| + \|[\bar{A}x_s - \bar{a}]_+\| \leq \sqrt{2}(1 + \|A_r(B_r)^{-1}\|) \left\| \begin{matrix} [Ax - a]_+ \\ Bx - b \end{matrix} \right\|. \tag{5.7}$$

Dividing the left-hand side of (5.6) by the right-hand side of (5.7) and then using (5.6), (5.7), and the fact $(\alpha\chi+\beta)/(\chi+\delta) \le \alpha+\beta/\delta$ for any four nonnegative scalars $\alpha, \beta, \chi, \delta$ such that the two fractions are defined (with the convention $0/0 = 0$), we obtain

$$\frac{1}{2(1+\|A_r(B_r)^{-1}\|)} \frac{d(x, P(A,B,a,b))}{\left\| \begin{bmatrix} [Ax-a]_+ \\ Bx-b \end{bmatrix} \right\|} \le \|(B_r)^{-1}\| + (\|(B_r)^{-1}B_s\|+1) \frac{d(x_s, P(\bar{A},\bar{a}))}{\|[\bar{A}x_s - \bar{a}]_+\|}.$$

The choice of x above was arbitrary, so the above relation holds for all $x \notin P(A,B,a,b)$. Taking the supremum of both sides over all $x \notin P(A,B,a,b)$ and observing that the x_s corresponding to x is not in $P(\bar{A},\bar{a})$, we obtain (cf. (1.1) and (5.2))

$$\frac{\tau(A,B,a,b)}{2(1+\|A_r(B_r)^{-1}\|)} \le \|(B_r)^{-1}\| + (\|(B_r)^{-1}B_s\|+1)\tau(\bar{A},\bar{a}).$$

Rearranging terms and the first inequality in (5.5) is proved.

To prove the second inequality in (5.5), fix any $x_s \notin P(\bar{A},\bar{a})$. By letting $\bar{x}$ be the element of $P(A,B,a,b)$ nearest to

$$x = \begin{bmatrix} (B_r)^{-1}(b - B_s x_s) \\ x_s \end{bmatrix}$$

and observing that $\bar{x}_s$ solves (5.4) so it is in $P(\bar{A},\bar{a})$, we obtain

$$d(x_s, P(\bar{A},\bar{a})) \le \|x_s - \bar{x}_s\| \le \|x - \bar{x}\| = d(x, P(A,B,a,b)). \tag{5.8}$$

Also, using the definition of x and (5.3), we have $Bx = b$ so

$$\left\| \begin{bmatrix} [Ax-a]_+ \\ Bx-b \end{bmatrix} \right\| = \|[Ax-a]_+\|$$

$$= \|[(A_s - A_r(B_r)^{-1}B_s)x_s - (a - A_r(B_r)^{-1}b)]_+\| \tag{5.9}$$

$$= \|[\bar{A}x_s - \bar{a}]_+\|. \tag{5.10}$$

Dividing the left-hand side of (5.8) by the right-hand side of (5.10), we obtain

$$\frac{d(x_s, P(\bar{A},\bar{a}))}{\|[\bar{A}x_s - \bar{a}]_+\|} \le \frac{d(x, P(A,B,a,b))}{\left\| \begin{bmatrix} [Ax-a]_+ \\ Bx-b \end{bmatrix} \right\|}.$$

The choice of x_s above was arbitrary, so the above relation holds for all $x_s \notin P(\bar{A},\bar{a})$. Taking the supremum of both sides over all $x_s \notin P(\bar{A},\bar{a})$ and observing that the x corresponding to x_s is not in $P(A,B,a,b)$, we obtain (cf. (1.1) and (5.2))

$$\tau(\bar{A},\bar{a}) \le \tau(A,B,a,b).$$

This proves the second inequality in (5.5). □

5.1. Well-conditionedness under feasible perturbations. Lemma 5.1, though essential to our analysis, covers only the case where B has full row rank. We need in addition the following lemma to cover the case where B lacks full row rank. According to this lemma, that B has either full row rank or full column rank is a necessary condition for the system (5.1) to be well-conditioned under feasible local perturbations on the problem data. The proof of this is patterned after that of (b) $\Rightarrow$ (a) in Theorem 2.2.

LEMMA 5.2. *Suppose that the system $Ax \leq a, Bx = b$ has a solution. For there to exist scalars $\delta > 0$ and $\beta > 0$ such that $\tau(A', B', a', b') \leq \beta$ for all (A', a') and (B', b') with $\|(A', a') - (A, a)\| + \|(B', b') - (B, b)\| < \delta$ and $A'x \leq a', B'x = b'$ solvable, it is necessary that B has either full row rank or full column rank.*

Proof. Suppose that B has neither full row rank nor full column rank. We will show that there cannot exist any scalars $\delta > 0$ and $\beta > 0$ with the stated properties.

Since B lacks full row rank and full column rank, there exist a nonzero l-vector θ and a nonzero n-vector z satisfying $B^T\theta = 0$ and $Bz = 0$. Normalizing if necessary, we will assume that $\|\theta\| = \|z\| = 1$. Let $\bar{x}$ be any solution of the system $Ax \leq a, Bx = b$ and let I denote the set of indices i for which $A_i\bar{x} = a_i$. Then, $A_I\bar{x} = a_I$, $A_{\tilde{I}}\bar{x} < a_{\tilde{I}}$ and $B\bar{x} = b$, where $\tilde{I}$ denotes the complement of I relative to $\{1, \ldots, m\}$. By reordering the rows of A and a if necessary, we can assume that

$$A = \begin{bmatrix} A_I \\ A_{\tilde{I}} \end{bmatrix}, \qquad a = \begin{bmatrix} a_I \\ a_{\tilde{I}} \end{bmatrix}.$$

For each scalar $\epsilon > 0$, we define the perturbed right-hand side a^ϵ by

$$a_I^\epsilon = a_I + 2\epsilon[A_I z]_+, \qquad a_{\tilde{I}}^\epsilon = a_{\tilde{I}},$$

and, similarly, we define

$$B^\epsilon = B + \epsilon\theta z^T, \qquad b^\epsilon = b + \epsilon(\langle \bar{x}, z\rangle + \epsilon)\theta.$$

Clearly, $B^\epsilon \to B$, $a^\epsilon \to a$ and $b^\epsilon \to b$ as $\epsilon \to 0$. Let $x^\epsilon = \bar{x} + \epsilon z$. Then,

$$A_I x^\epsilon = A_I\bar{x} + \epsilon A_I z \leq a_I + 2\epsilon[A_I z]_+ = a_I^\epsilon \tag{5.11}$$

and, for ϵ sufficiently small,

$$A_{\tilde{I}} x^\epsilon = A_{\tilde{I}}\bar{x} + \epsilon A_{\tilde{I}} z < a_{\tilde{I}}. \tag{5.12}$$

By a similar calculation, we also have that $B^\epsilon x^\epsilon = b^\epsilon$. Thus, the vector x^ϵ belongs to $P(A, B^\epsilon, a^\epsilon, b^\epsilon)$ for all ϵ sufficiently small. Finally, we have from $(B^\epsilon)^T\theta = (B + \epsilon\theta z^T)^T\theta = \epsilon z$ that z is a linear combination of the columns of $(B^\epsilon)^T$, which, together with (5.11) and (5.12), implies that z is a normal to the polyhedral set $P(A, B^\epsilon, a^\epsilon, b^\epsilon)$ at x^ϵ, for all ϵ sufficiently small. Then, if we move along the direction z from x^ϵ and then project back onto this set, we always get x^ϵ. In particular, the projection of $x^\epsilon + \epsilon z$ onto $P(A, B^\epsilon, a^\epsilon, b^\epsilon)$ is x^ϵ, so that

$$d(x^\epsilon + \epsilon z, P(A, B^\epsilon, a^\epsilon, b^\epsilon)) = \|\epsilon z\|.$$

By a calculation similar to that used in (5.11) and (5.12), we see that $A(x^\epsilon + \epsilon z) \leq a^\epsilon$ for all ϵ sufficiently small, in which case $[A(x^\epsilon + \epsilon z) - a^\epsilon]_+ = 0$. Also, we have from

$Bz = 0$ and $\|z\| = 1$ that $B^\epsilon(x^\epsilon + \epsilon z) = b^\epsilon + \epsilon^2\theta$. Combining these two results with the relation above and using $\|z\| = \|\theta\| = 1$, we obtain

$$\frac{d(x^\epsilon + \epsilon z, P(A, B^\epsilon, a^\epsilon, b^\epsilon))}{\left\| \begin{matrix} [A(x^\epsilon + \epsilon z) - a^\epsilon]_+ \\ B^\epsilon(x^\epsilon + \epsilon z) - b^\epsilon \end{matrix} \right\|} = \frac{\|\epsilon z\|}{\|\epsilon^2\theta\|} = \frac{1}{\epsilon},$$

for all ϵ sufficiently small. By (5.2), the left-hand side in the above relation is bounded above by $\tau(A, B^\epsilon, a^\epsilon, b^\epsilon)$. Thus, $\tau(A, B^\epsilon, a^\epsilon, b^\epsilon) \geq 1/\epsilon \to \infty$ as $\epsilon \to 0$. □

By combining Lemmas 5.1 and 5.2 with Theorem 2.2, we obtain the first main result of this section, which may be viewed as an extension of Theorem 2.2 to the general system (5.1).

THEOREM 5.3 (*Well-conditionedness under feasible semilocal perturbations*). *For any $m \times n$ matrix A and any $l \times n$ matrix B, the following conditions are equivalent.*

(a) *B has either full column rank or full row rank and, if B has full row rank, then, for each nonempty index set $I \subseteq \{1, \ldots, m\}$, we have that either $A_I x < 0$, $Bx = 0$ is solvable or*

$$\begin{bmatrix} A_I \\ B \end{bmatrix}$$

has full column rank.

(b) *There exist scalars $\delta > 0$ and $\beta > 0$ such that, for any (A', a') and (B', b') with $\|A' - A\| + \|B' - B\| < \delta$ and $A'x \leq a'$, $B'x = b'$ solvable, we have $\tau(A', B', a', b') \leq \beta$.*

Proof. We consider two separate cases. First, suppose that B has full row rank. Partition A and B according to (5.3) with B_r invertible, and let $\bar{A} = A_s - A_r(B_r)^{-1}B_s$. Then, the following statements are equivalent.

Condition (a) holds $\iff$ $\bar{A}$ satisfies condition (a) in Theorem 2.2, (by Lemma 5.1(c)–(d)),
$\iff$ $\tau(\bar{A}, \bar{a})$ is uniformly bounded under feasible local perturbations on $\bar{A}$ and feasible global perturbations on $\bar{a}$, (by Theorem 2.2),
$\iff$ $\tau(A, B, a, b)$ is uniformly bounded under feasible local perturbations on A, B and feasible global perturbations on a, b, (by Lemma 5.1(e)),
$\iff$ condition (b) holds.

The third equivalence also uses the observation that any feasible local perturbation on $\bar{A}$ and feasible global perturbation on $\bar{a}$ translates into a feasible local perturbation on A, B and a feasible global perturbation on a, b (more precisely, we associate with each perturbed version of $\bar{A}$ and $\bar{a}$, say $\bar{A}'$ and $\bar{a}'$, the following perturbed versions of A, B, a, b respectively: $A' = [\, A_r \quad A_s + (\bar{A}' - \bar{A})\,]$, $B' = B$, $a' = a + (\bar{a}' - \bar{a})$, and $b' = b$); and vice versa. Second, suppose that B lacks full row rank. If condition (b) holds, then, by Lemma 5.2, B must have full column rank. Conversely, if B has full column rank, then, upon letting I be any nonempty subset of $\{1, \ldots, l\}$ with B_I invertible, we see that $P(A, B, a, b)$ comprises the single point $(B_I)^{-1}b_I$ whenever it

is nonempty, in which case

$$d(x, P(A,B,a,b)) = \|x - (B_I)^{-1}b_I\| \leq \|(B_I)^{-1}\|\|Bx - b\| \quad \forall x.$$

This implies $\tau(A,B,a,b) \leq \|(B_I)^{-1}\|$ (cf. (5.2)), so $\tau(A,B,a,b)$ is uniformly bounded under feasible local perturbations on A,B and feasible global perturbations on a,b. □

Analogous to Proposition 2.3, in the case where the system (5.1) has an unbounded solution set, condition (a) in Theorem 5.3 simplifies to: B has full row rank and the system $Ax < 0$, $Bx = 0$ has a solution. For brevity, we omit the proof.

By combining Lemmas 5.1 and 5.2 with Theorem 2.4, we obtain the second main result of this section, which may be viewed as an extension of Theorem 2.2 to the general system (5.1).

THEOREM 5.4 (*Well-conditionedness under feasible local perturbations*). *For any $m \times n$ matrix A and any $l \times n$ matrix B such that the system $Ax \leq a$, $Bx = b$ is solvable, the following conditions are equivalent.*

(a) *B has either full column rank or full row rank and, if B has full row rank, then, for each nonempty index set $I \subseteq \{1,\ldots,m\}$ such that either $Ax \leq a$, $A_I x = a_I$, $Bx = b$ or $Ax \leq 0$, $A_I x = 0$, $Bx = 0$, $x \neq 0$ is solvable, we have that either $A_I x < 0$, $Bx = 0$ is solvable or*

$$\begin{bmatrix} A_I \\ B \end{bmatrix}$$

has full column rank.

(b) *There exist scalars $\delta > 0$ and $\beta > 0$ such that for any (A',a') and (B',b') with $\|(A',a') - (A,a)\| + \|(B',b') - (B,b)\| < \delta$ and $A'x \leq a', B'x = b'$ solvable, we have $\tau(A',B',a',b') \leq \beta$.*

Proof. The proof is essentially identical to that of Theorem 5.3, except that "feasible global perturbations," "Theorem 2.2," and "Lemma 5.1(c)–(d)" are replaced by, respectively, "feasible local perturbations," "Theorem 2.4," and "Lemma 5.1(a)–(d)." □

Analogous to Proposition 2.5, in the case where the system (5.1) has multiple solutions, condition (a) in Theorem 5.4 simplifies to: either B has full row rank and the system $Ax < 0$, $Bx = 0$ is solvable or the system (5.1) is regular (i.e., B has full row rank and $Ax < a$, $Bx = b$ is solvable) and has a bounded solution set. In the case where (5.1) has a unique solution, this condition can again be simplified as in Proposition 2.6, albeit only slightly.

5.2. Well-conditionedness under arbitrary perturbations. By combining Lemma 5.1 with Theorem 3.1, the third main result of this section, which may be viewed as an extension of Theorem 3.1 to the general system (5.1), readily follows.

THEOREM 5.5 (*Well-conditionedness under semilocal perturbations*). *For any $m \times n$ matrix A and any $l \times n$ matrix B, the following conditions are equivalent.*

(a) *B has full row rank and the system $Ax < 0$, $Bx = 0$ is solvable.*

(b) *There exist scalars $\delta > 0$ and $\beta > 0$ such that for any (A',a') and (B',b') with $\|A'-A\|+\|B'-B\| < \delta$, the system $A'x \leq a', B'x = b'$ is solvable and $\tau(A',B',a',b') \leq \beta$.*

Proof. First, suppose that B has full row rank. Then, by applying the argument used in the proof of Theorem 5.3 with the word "feasible" removed and with

"Theorem 2.2" and "Lemma 5.1(c)–(d)" replaced by, respectively, "Theorem 3.1" and "Lemma 5.1(c) with $I = \{1, \ldots, m\}$," we see that condition (a) holds if and only if condition (b) holds. Second, suppose that B lacks full row rank. Then, there is some l-vector b' for which $Bx = b'$ has no solution, so that $Ax \leq a', Bx = b'$ has no solution for any m-vector a'. Hence, both conditions (a) and (b) fail to hold. □

As with Theorem 3.1, the (a) ⇒ (b) part of Theorem 5.5 is not new (see [Bra88, Thm. 5.4.13]) and is included for completeness only. Also, we note in passing that condition (a) in Theorem 5.5 is simply the Mangasarian–Fromovitz constraint qualification condition applied to the system (5.1) (see [MaF67]). This shows yet another application of this well-known constraint qualification condition.

By combining Lemma 5.1 with Theorem 3.3, we obtain the fourth main result of this section, which may be viewed as an extension of Theorem 3.3 to the general system (5.1).

THEOREM 5.6 (*Well-conditionedness under local perturbations*). *For any $m \times n$ matrix A, any $l \times n$ matrix B, any m-vector a and l-vector b, the following conditions are equivalent.*

(a) *Either B has full row rank and the system $Ax < 0$, $Bx = 0$ is solvable or the system* (5.1) *is regular and has a bounded solution set.*

(b) *There exist scalars $\delta > 0$ and $\beta > 0$ such that for any (A', a') and (B', b') with $\|(A', a') - (A, a)\| + \|(B', b') - (B, b)\| < \delta$, the system $A'x \leq a', B'x = b'$ is solvable and $\tau(A', B', a', b') \leq \beta$.*

Proof. The proof is essentially identical to that of Theorem 5.5, except that "global perturbations," "Theorem 3.1" and "Lemma 5.1(c)" are replaced by, respectively, "local perturbations," "Theorem 3.3," and "Lemma 5.1(a)–(c)." □

Analogous to the definition given in §4, let us say that the system (5.1) satisfies the u.b.v. condition if there exist scalars $\delta > 0$ and $\beta > 0$ such that, for any (A', a'), (B', b') with $\|(A', a') - (A, a)\| + \|(B', b') - (B, b)\| < \delta$, the vertices of $P(A', B', a', b')$, if any exist, all have Euclidean norm less than β. By combining parts (a)–(b), (d) of Lemma 5.1 with Theorem 4.1, we readily obtain the final result of this section, which is an extension of Theorem 4.1 to the general system (5.1).

THEOREM 5.7. *For any $m \times n$ matrix A, any $l \times n$ matrix B, any m-vector a and any l-vector b such that the system* (5.1) *is regular and has a vertex solution, the following conditions are equivalent.*

(a) *There exists a nonzero n-vector z satisfying $Az \leq 0, Bz = 0$ and the rows of*

$$\begin{bmatrix} A_I \\ B \end{bmatrix}$$

are linearly dependent, where $I = \{\, i \in \{1, \ldots, m\} \mid A_i z = 0 \,\}$.

(b) *The system* (5.1) *does not satisfy the u.b.v. condition.*

As a corollary of Theorems 5.6 and 5.7, we have that the system (5.1) is well-conditioned under local perturbations on (A, a) and (B, b), whenever it is regular, has a vertex solution, and satisfies the u.b.v. condition.

6. Conclusion and extensions. In this paper, we have studied in detail the well-conditionedness of a linear system under each of four sets of perturbations on the problem data. In particular, we gave a necessary and sufficient condition for the system to be well-conditioned under each set of perturbations. We also related

well-conditionedness (under one of the sets of perturbations) to Pang's uniform boundedness condition on the vertex solutions.

We are presently investigating applications of our results to a number of problems, amongst which are (i) the sensitivity analysis of convex programs/complementarity problems, (ii) the convergence analysis of descent methods for nonlinearly constrained minimization, and (iii) the derivation of local error bounds for nonlinearly constrained problems. Also, a question closely related to our work concerns the classification of those polyhedral sets that admit a well-conditioned algebraic representation. Some preliminary results in this direction have been obtained, but much remains to be done.

Acknowledgments. We thank Professors J.-S. Pang and O. L. Mangasarian for several stimulating discussions on the subject of this paper. We are particularly indebted to Professor Pang for suggesting the main problem of this study. Also, we thank an anonymous referee for making a careful reading of this paper and offering many helpful comments on the presentation.

REFERENCES

[AuC88] A. A. AUSLENDER AND J.-P. CROUZEIX, *Global regularity theorems*, Math. Oper. Res., 13 (1988), pp. 1–11.

[BeS91] C. BERGTHALLER AND I. SINGER, *The distance to a polyhedron*, Linear Algebra Appl., 169 (1992), pp. 111–129.

[Bra88] L. BRADY, *Condition Constants for Solutions of Convex Inequalities*, Ph.D. thesis, Department of Computer Sciences, University of Wisconsin, Madison, WI, 1988.

[CGST86] W. COOK, A. M. H. GERARDS, A. SCHRIJVER, AND É. TARDÖS, *Sensitivity results in integer linear programming*, Math. Prog., 34 (1986), pp. 251–264.

[Dan73] J. W. DANIEL, *On perturbations in systems of linear inequalities*, SIAM J. Numer. Anal., 10 (1973), pp. 299–307.

[Dan75] ———, *Remarks on perturbations in linear inequalities*, SIAM J. Numer. Anal., 12 (1975), pp. 770–772.

[FRT85] R. M. FREUND, R. ROUNDY, AND M. J. TODD, *Identifying the Set of Always Active Constraints in a System of Linear Inequalities by a Single Linear Program*, Sloan School of Management Sciences Working Paper No. 1674–1685, Massachusetts Institute of Technology, Cambridge, MA, 1985.

[Gof80] J. L. GOFFIN, *The relaxation method for solving systems of linear inequalities*, Math. Oper. Res., 5 (1980), pp. 388–414.

[Gül92] O. GÜLER, *Augmented lagrangian algorithms for linear programming*, J. Optim. Theory Appl., 75 (1992), pp. 445–478.

[Hof52] A. J. HOFFMAN, *On approximate solutions of systems of linear inequalities*, J. Res. Natl. Bur. Standards, 49 (1952), pp. 263–265.

[IuD90] A. N. IUSEM AND A. R. DE PIERRO, *On the convergence properties of Hildreth's quadratic programming algorithm*, Math. Prog., 47 (1990), pp. 37–51.

[Li91] W. LI, *Lipschitz Continuity and Strong Uniqueness in Linear Programs and Polyhedral Approximations*, Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA, 1991.

[LuT92] Z.-Q. LUO AND P. TSENG, *On the linear convergence of descent methods for convex essentially smooth minimization*, SIAM J. Control Optim., 30 (1992), pp. 408–425.

[Man81a] O. L. MANGASARIAN, *A stable theorem of the alternative: An extension of the Gordan theorem*, Linear Algebra Appl., 41 (1981), pp. 209–223.

[Man81b] ———, *A Condition Number for Linear Inequalities and Linear Systems*, in Methods of Operations Research 43, Proc. of 6, Symposium über Operations Research, Universität Augsburg, September 7–9, 1981, G. Bamberg and O. Opitz, eds., Verlagsgruppe Athenäum/Hain/Scriptor/Hanstein, Konigstein, 1981, pp. 3–15.

[Man85] ———, *A condition number for differentiable convex inequalities*, Math. Oper. Res., 10 (1985), pp. 175–179.

[MaF67] O. L. MANGASARIAN AND S. FROMOVITZ, *The Fritz John necessary optimality conditions in the presence of equality and inequality constraints*, J. Math. Anal. Appl., 17 (1967), pp. 37–47.

[MaS87] O. L. MANGASARIAN AND T.-H. SHIAU, *Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems*, SIAM J. Control Optim., 25 (1987), pp. 583–595.

[Pan91] J.-S. PANG, Private communication, May 1991.

[Rob72] S. M. ROBINSON, *Normed convex processes*, Trans. Amer. Math. Soc., 174 (1972), pp. 127–140.

[Rob73a] ———, *Bounds for error in the solution set of a perturbed linear program*, Linear Algebra Appl., 6 (1973), pp. 69–81.

[Rob73b] ———, *Perturbations in Finite-Dimensional Systems of Linear Inequalities and Equations*, Mathematical Research Center Technical Summary Report No. 1357, University of Wisconsin at Madison, August 1973.

[Rob75a] ———, *Stability theory for systems of inequalities, Part* I: *Linear systems*, SIAM J. Numer. Anal., 12 (1975), pp. 754–769.

[Rob75b] ———, *An application of error bounds for convex programming in a linear space*, SIAM J. Control, 13 (1975), pp. 271–273.

[Rob76] ———, *Stability theory for systems of inequalities, Part* II: *Differentiable nonlinear systems*, SIAM. J. Numer. Anal., 13 (1976), pp. 497–513.

[Rob77] ———, *A characterization of stability in linear programming*, Oper. Res., 25 (1977), pp. 435–447.

[Roc70] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[Ros51] P. C. ROSENBLOOM, *Quelques Classes de Problèmes Extrémaux*, Bulletin of the Société Mathématique de France, 79 (1951), pp. 1–58.

[Sch86] A. SCHRIJVER, *Theory of Linear and Integer Programming*, John Wiley & Sons, New York, 1986.

[TsB93] P. TSENG AND D. P. BERTSEKAS, *On the convergence of the exponential multiplier method for convex programming*, Math. Prog., 60 (1993), pp. 1–19.

[TsL92] P. TSENG AND Z.-Q. LUO, *On the convergence of the affine scaling algorithm*, Math. Prog., 56 (1992), pp. 301–319.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 661–671, April 1994

019

A SECOND-ORDER PERTURBATION EXPANSION FOR THE SVD*

RICHARD J. VACCARO†

Abstract. Let A be a rank-deficient matrix and let N be a matrix whose norm is small compared with that of A. The left singular vectors of A can be grouped into two matrices U_1 and U_2 whose columns provide orthonormal bases for the p-dimensional column space of A and for its $n-p$ dimensional orthogonal complement. The left singular vectors of $\tilde{A} = A + N$ can also be partitioned into the first p columns, $\tilde{U}_1$, and the last $n-p$ columns $\tilde{U}_2$. When analyzing a variety of signal processing algorithms, it is useful to know how different the spaces spanned by U_1 and $\tilde{U}_1$ (or U_2 and $\tilde{U}_2$) are. This question can be answered by developing a perturbation expansion for the subspace spanned by a set of singular vectors. A first-order expansion of this type has recently been developed and used to analyze the performance of direction-finding algorithms in array signal processing. In this paper, a new second-order expansion is derived and the result is illustrated with two examples.

Key words. singular value decomposition, perturbation expansion, singular subspaces

AMS subject classifications. 15A18, 15A52, 15A60

1. Introduction.

Let A be an $m \times n$ matrix of rank p, where $p < \min(m,n)$. The singular value decomposition (SVD) of A can be partitioned as follows:

$$A = [U_s \quad U_\perp] \begin{bmatrix} \Sigma_s & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_s^H \\ V_\perp^H \end{bmatrix}, \tag{1}$$

where the columns of U_s and $U_\perp$ provide a basis for the column-space of A and its orthogonal complement, respectively. The matrix A could be complex valued, and the superscript H means complex conjugate transpose. The column-space of A is called the *signal subspace* in the signal processing literature, and we will use the subscript s to refer to this subspace. The signal subspace S of the matrix A is the span of the p columns of U_s. We will refer to the orthogonal complement of the signal subspace by the abbreviated title *orthogonal subspace* in this paper. The orthogonal subspace $S^\perp$ is the span of the $n-p$ columns in $U_\perp$. Thus,

$$S \stackrel{\text{def}}{=} \text{span}(U_s) \quad \text{and} \quad S^\perp \stackrel{\text{def}}{=} \text{span}(U_\perp).$$

In practice, the matrix A is not available, but only a noise-corrupted matrix

$$\tilde{A} = A + N.$$

An SVD of the perturbed matrix $\tilde{A}$ can be partitioned similar to that in (1)

$$\tilde{A} = [\tilde{U}_s \quad \tilde{U}_\perp] \begin{bmatrix} \tilde{\Sigma}_s & 0 \\ 0 & \tilde{\Sigma}_\perp \end{bmatrix} \begin{bmatrix} \tilde{V}_s^H \\ \tilde{V}_\perp^H \end{bmatrix}, \tag{2}$$

where $\tilde{U}_s$ has p columns and $\tilde{U}_\perp$ has $n-p$ columns. Although $\tilde{A}$ will generally have full rank, it will be "close" to a matrix of rank p (if N is small), and the p-dimensional subspace spanned by the columns of $\tilde{U}_s$ will be called the *perturbed signal subspace*

$$\tilde{S} \stackrel{\text{def}}{=} \text{span}(\tilde{U}_s). \tag{3}$$

* Received by the editors December 30, 1991; accepted for publication (in revised form) November 4, 1992. This research was supported in part by the Office of Naval Research under grant N00014-90-J1283.

† Department of Electrical Engineering, The University of Rhode Island, Kingston, Rhode Island 02881 (vaccaro@ele.uri.edu).

We also define the *perturbed orthogonal subspace* $\tilde{S^{\perp}}$ as

$$\tilde{S^{\perp}} \stackrel{\text{def}}{=} \text{span}(\tilde{U}_{\perp}). \tag{4}$$

We would like to know how different the perturbed subspace $\tilde{S}$ ($\tilde{S^{\perp}}$) is from the unperturbed subspace S ($S^{\perp}$) as a function of the noise matrix N.

In a series of papers, the performance of algorithms for estimating the directions-of-arrival of plane waves impinging on an array of sensors was studied [2]–[6]. Many algorithms in array signal processing can be expressed in terms of an SVD of a matrix formed from array data. The performance of such algorithms can be analyzed using a perturbation expansion for the SVD. This was done in [2]–[6] using a first-order expansion to derive expressions for the variance of estimated directions-of-arrival. However, a first-order expansion will always predict zero bias when the additive noise is zero mean, because the resulting perturbation terms are linear in the noise matrix. To analyze bias, a second-order perturbation expansion is necessary. This is one motivation for the second-order expansion that is derived in this paper.

For the purpose of *performance analysis*, the matrices A and N are individually known. In a statistical performance analysis, only the statistics of N are known (e.g., its elements are independent and identically distributed (i.i.d.)). In any case, the perturbation formulas derived in this paper, which are functions of the perturbation matrix N, can be used directly for performance analysis. A second use for the perturbation formulas is the development of new signal processing algorithms. In this case only the matrix $\tilde{A}$ is observed, but the matrix N is described statistically, up to a scale factor. The first-order perturbation expansion has been used to derive algorithms for signal estimation [7] and for direction-finding in array signal processing [11].

2. A second-order perturbation expansion.

2.1. Preliminaries. To be useful, the perturbed subspaces must not be "too far" from the unperturbed subspaces. This will be true if the noise matrix N is "small enough." In this case, basis vectors for the perturbed signal and orthogonal subspaces can be found by appropriately combining the unperturbed singular vectors, as shown in the following lemma.

LEMMA. *Let $\tilde{A} = A + N$ with SVDs of A and $\tilde{A}$ given by* (1) *and* (2), *respectively. Assume that $\|N\|_2$ is less than the smallest nonzero singular value of A. Let the perturbed signal and orthogonal subspaces be defined by* (3) *and* (4), *respectively. Then $\tilde{S^{\perp}}$ is spanned by the columns of $U_{\perp} + U_s Q$ and $\tilde{S}$ is spanned by the columns of $U_s + U_{\perp} R$, where Q and R are matrices whose norms are of the order of $\|N\|$.*

The proof of this lemma can be found in [5], and this result is also implicit in a paper by Stewart [8]. In the lemma the assumption that $\|N\|_2$ is less than the smallest nonzero singular value of A is equivalent to assuming a high signal-to-noise ratio. If the signal-to-noise ratio is not high enough, the subspaces of the perturbed and unperturbed matrices A and $\tilde{A}$ may be quite different (e.g., they may not have the same dimension), and so the perturbation expansions given in the lemma are not useful. The case when $\|N\|$ is greater than the smallest nonzero singular value of A is treated using other techniques in [10].

The main result of this paper is the derivation of expressions for the coefficient matrices Q and R, which are correct up to second-order terms in the noise matrix N. Before presenting that derivation, we first show two preliminary calculations.

The lemma above gives bases for the perturbed signal and orthogonal subspaces. However, it is easy to show that the given bases are not orthonormal. In particular,

for the orthogonal subspace, we have

$$(U_\perp^H + Q^H U_s^H)(U_\perp + U_s Q) = (I + Q^H Q).$$

The above equation shows how the basis for the perturbed orthogonal subspace can be normalized, and a similar equation holds for the perturbed signal subspace. The result is that an orthonormal basis for the perturbed orthogonal subspace is given by

$$(U_\perp + U_s Q)(I + Q^H Q)^{-1/2} \tag{5}$$

and an orthonormal basis for the perturbed signal subspace is given by

$$(U_s + U_\perp R)(I + R^H R)^{-1/2}.$$

We now show that the coefficient matrices Q and R are related in a simple way. Because the perturbed signal and orthogonal subspaces are orthogonal to each other, it must be true that the (unnormalized) basis vectors given in the lemma are orthogonal. That is,

$$\begin{aligned}(U_\perp^H + Q^H U_s^H)(U_s + U_\perp R) &= 0,\\ R + Q^H &= 0,\end{aligned}$$

or

$$R = -Q^H.$$

Thus in the derivation that follows, we only need to compute Q to some desired accuracy, and then use the above equation to get R.

2.2. Derivation of the perturbation expansion. Let the matrices A and $\tilde{A}$ have SVDs given by (1) and (2), respectively. We can expand the SVD of $\tilde{A}$ to get

$$\tilde{A} = A + N = \tilde{U}_s \tilde{\Sigma}_s \tilde{V}_s^H + \tilde{U}_\perp \tilde{\Sigma}_\perp \tilde{V}_\perp^H.$$

Premultiply the above equation by $(U_\perp + U_s Q)^H$ to get

$$(U_\perp^H + Q^H U_s^H)(U_s \Sigma_s V_s^H + N) = (U_\perp^H + Q^H U_s^H)(\tilde{U}_\perp \tilde{\Sigma}_\perp \tilde{V}_\perp^H). \tag{6}$$

From the lemma we know that $(U_\perp + U_s Q)$ and $\tilde{U}_\perp$ span the same space, and thus their columns are related by a nonsingular matrix X as follows:

$$\tilde{U}_\perp = (U_\perp + U_s Q)X. \tag{7}$$

We can now simplify (6) using (7) to yield

$$\begin{aligned}U_\perp^H N + Q^H(\Sigma_s V_s^H + U_s^H N) &= (U_\perp^H + Q^H U_s^H)(\tilde{U}_\perp \tilde{\Sigma}_\perp \tilde{V}_\perp^H)\\ &= X\tilde{\Sigma}_\perp \tilde{V}_\perp^H + Q^H Q X \tilde{\Sigma}_\perp \tilde{V}_\perp^H\\ &\overset{2}{=} X\tilde{\Sigma}_\perp \tilde{V}_\perp^H.\end{aligned}$$

We use the notation "$\overset{i}{=}$" to mean "equal up to terms of order $\|N\|^i$" for i=0,1, or 2. The last line in the above equation is obtained by noting that Q and $\tilde{\Sigma}_\perp$ are

first-order terms, and so the omitted term from the second line is third order. After taking conjugate transposes of both sides of the above equation and letting

$$M = N^H U_s + V_s \Sigma_s, \tag{8}$$

we can solve for Q to obtain

$$\begin{aligned} Q &\overset{2}{=} -M^\dagger (N^H U_\perp + \tilde{V}_\perp \tilde{\Sigma}_\perp^H X^H) \\ &\overset{2}{=} -M^\dagger N^H U_\perp - M^\dagger \tilde{V}_\perp \tilde{\Sigma}_\perp^H X^H \\ &\overset{2}{=} -T_1 - T_2, \end{aligned} \tag{9}$$

where T_1 and T_2 are implicitly defined in the above equation. Since T_1 has N as a factor, a first-order expression for Q is obtained by using a zeroth-order expansion of $M^\dagger$ in the term T_1. Similarly, since $\tilde{\Sigma}_\perp$ is a first-order term, a first-order expression for Q is obtained by using zeroth-order expansions for $M^\dagger$ and $\tilde{\Sigma}_\perp$ in T_2. The zeroth-order expansion of $M^\dagger$ is obtained by setting N to zero in the definition of M and computing

$$M^\dagger = (M^H M)^{-1} M^H \overset{0}{=} [(V_s \Sigma_s)^H (V_s \Sigma_s)]^{-1} \Sigma_s V_s^H = \Sigma_s^{-1} V_s^H.$$

Substituting this into (9) yields the previously known result (see [4], [6])

$$Q \overset{1}{=} -\Sigma_s^{-1} V_s^H N^H U_\perp.$$

For future reference, we define this first-order expression for Q to be Q_1; that is,

$$Q_1 = -\Sigma_s^{-1} V_s^H N^H U_\perp. \tag{10}$$

We can use (9) to generate a perturbation expansion of second order by using a first-order expansion of $M^\dagger$ in (9), and also considering the term T_2. It can be shown that T_2 contains only two second-order terms, and these terms sum to zero. Thus

$$T_2 \overset{2}{=} 0. \tag{11}$$

The derivation of (11) is tedious, but it is similar to the calculation of the T_1, which we give below.

The required first-order expansion of $M^\dagger$ is obtained as follows

(12)

$$\begin{aligned} M^\dagger &= [(U_s^H N + \Sigma_s V_s^H)(N^H U_s + V_s \Sigma_s)]^{-1} (U_s^H N + \Sigma_s V_s^H) \\ &= [U_s^H N N^H U_s + U_s^H N V_s \Sigma_s + \Sigma_s V^H N^H U_s + \Sigma_s^2]^{-1} (U_s^H N + \Sigma_s V_s^H) \\ &= [\Sigma_s^{-2} U_s^H N N^H U_s + \Sigma_s^{-2} U_s^H N V_s \Sigma_s + \Sigma_s^{-1} V^H N^H U_s + I]^{-1} \Sigma_s^{-2} (U_s^H N + \Sigma_s V_s^H). \end{aligned}$$

Consider approximating the expression in the last line of the equation above. Since we only need a first-order expansion of $M^\dagger$, the first term in brackets can be omitted. The inverse can be represented by the Neumann expansion

$$(I + Y)^{-1} = I - Y + Y^2 - \cdots. \tag{13}$$

If we let Y be the terms linear in N in the first bracketed term of (12), keep up to the linear term in (13), and substitute the resulting expression for $M^\dagger$ into T_1 in (9), we get

$$Q \stackrel{2}{=} -[I - \Sigma_s^{-1} V_s^H N^H U_s - \Sigma_s^{-2} U_s^H N V_s \Sigma_s] \Sigma_s^{-2} [U_s^H N + \Sigma_s V_s^H] N^H U_\perp.$$

The above equation is correct up to second-order terms, but it also contains some third-order terms that can be deleted. The result is, after some simplification, a second-order expression for Q which is denoted Q_2

$$Q_2 = Q_1 + (-\Sigma_s^{-2} U_s^H N V_\perp V_\perp^H + \Sigma_s^{-1} V_s^H N^H U_s \Sigma_s^{-1} V_s^H) N^H U_\perp. \tag{14}$$

From the above development, we know that

$$\tilde{S^\perp} \stackrel{2}{=} \text{span}(U_\perp + U_s Q_2) \quad \text{and} \quad \tilde{S} \stackrel{2}{=} \text{span}(U_s - U_\perp Q_2^H).$$

However, the basis vectors in the above equation are not orthonormal up to second order. They can be normalized as shown in (5). The normalization only has to be computed up to second-order terms. The result for the orthogonal subspace is

$$(I + Q^H Q)^{-1/2} \stackrel{2}{=} I - \frac{1}{2} Q_1^H Q_1.$$

Applying this normalization and dropping terms higher than second order yields an orthogonal basis for the perturbed orthogonal subspace

$$\tilde{S^\perp} \stackrel{2}{=} \text{span}\left[U_\perp \left(I - \frac{1}{2} Q_1^H Q_1\right) + U_s Q_2\right], \tag{15}$$

where Q_1 is defined in (10) and Q_2 is defined in (14). The corresponding result for the perturbed signal subspace is

$$\tilde{S} \stackrel{2}{=} \text{span}\left[U_s \left(I - \frac{1}{2} Q_1 Q_1^H\right) - U_\perp Q_2^H\right].$$

3. Statistics of estimated projection matrices. Once the signal and orthogonal subspaces of a data matrix are estimated, many signal processing tasks require projection onto an estimated subspace. Thus it is useful to characterize the estimated orthogonal projection matrix.

To be specific, we consider the orthogonal subspace in this section. From (2), the projection onto the estimated orthogonal subspace is given by

$$\tilde{P}_\perp = \tilde{U}_\perp \tilde{U}_\perp^H.$$

For future reference, we define the projection matrices for the unperturbed signal and orthogonal subspaces as

$$P_s = U_s U_s^H \quad \text{and} \quad P_\perp = U_\perp U_\perp^H,$$

respectively. Suppose we want to characterize $\tilde{P}_\perp$ in terms of the unperturbed projection matrices and the noise matrix N. For example, if we assume that the elements of N are i.i.d. random variables with variance σ^2, we can calculate the expected value of $\tilde{P}_\perp$. Using the subspace perturbation expansions developed in the previous

sections, we can approximate the perturbed projection matrices as follows. Using the first-order expression, we get

$$\tilde{P}_\perp \stackrel{1}{=} (U_\perp + U_s Q_1)(U_\perp + U_s Q_1)^H \stackrel{\text{def}}{=} P_1.$$

This expression is exact to first order and contains some, but not all, of the second-order terms, as shown by expanding the expression as

$$P_1 = U_\perp U_\perp^H + U_\perp Q_1^H U_s^H + U_s Q_1 U_\perp^H + U_s Q_1 Q_1^H U_s^H. \tag{16}$$

A similar approximation for the perturbed projection matrix can be obtained using the second-order perturbation expansion as follows

$$\tilde{P}_\perp \stackrel{2}{=} \left[\left(I - \frac{1}{2}Q_1^H Q_1\right) + U_s Q_2\right]\left[\left(I - \frac{1}{2}Q_1^H Q_1\right) + U_s Q_2\right]^H.$$

If we expand this equation, keeping up to second-order terms, the result, which we denote P_2, is

$$P_2 = U_\perp U_\perp^H - U_\perp Q_1^H Q_1 U_\perp^H + U_\perp Q_2^H U_\perp^H + U_s Q_2 U_\perp^H + U_s Q_1 Q_1^H U_s^H. \tag{17}$$

To take expected values of P_1 and P_2, we need to be able to compute the expectation of the following matrices:

$$\begin{aligned} Q_1 &= -\Sigma_s^{-1} V_s^H N^H U_\perp, \\ Q_1^H Q_1 &= U_\perp^H N V_s \Sigma_s^{-2} V_s^H N^H U_\perp, \\ Q_1 Q_1^H &= \Sigma_s^{-1} V_s^H N^H U_\perp U_\perp^H N V_s \Sigma_s^{-1}, \\ Q_2 &= Q_1 - (\Sigma_s^{-2} U_s^H N V_\perp V_\perp^H + \Sigma_s^{-1} V_s^H N^H U_s \Sigma_s^{-1} V_s^H) N^H U_\perp. \end{aligned} \tag{18}$$

The required expectations can be derived using the following result from [3]

$$E[N\beta\beta^H N^H] = \|\beta\|^2 \sigma^2 I \tag{19}$$

for any constant vector β and matrix N consisting of i.i.d. random variables with variance σ^2. (If N contains complex elements, we assume that the real and imaginary parts are i.i.d. with variance $\sigma^2/2$.) It is also shown in [3] that

$$E[N^H \alpha\alpha^H N^H] = 0. \tag{20}$$

We can also show that for real-valued noise samples,

$$E[N\alpha\beta^T N] = \sigma^2 \beta\alpha^T. \tag{21}$$

It is easy to see that the expected value of Q_1 is zero. Equations (19)–(21) can be used to calculate the expected values of the other matrices in (18). As an example, we derive the expected value of Q_2 as follows:

$$E[Q_2] = -\Sigma_s^{-2} U_s^H E[N V_\perp V_\perp^H N^H] U_\perp - \Sigma_s^{-1} V_s E[N^H U_s \Sigma_s^{-1} V_s^H N^H] U_\perp. \tag{22}$$

The first expectation on the right-hand side of the above equation using (19) is

$$\begin{aligned} E[N V_\perp V_\perp^H N^H] &= \sum_{i=1}^{q} E[N v_{\perp i} v_{\perp i}^H N^H] \\ &= \sum_{i=1}^{q} \|v_{\perp i}\|^2 \sigma^2 I = q\sigma^2 I, \end{aligned} \tag{23}$$

where $v_{\perp i}$ is the ith column of $V_\perp$, and $q = n - p$ (see (1)). The second expectation on the right-hand side of (22) is zero if the noise samples are zero-mean i.i.d. complex random variables from (20). For real-valued noise samples, this expectation can be evaluated using (21) as

$$\begin{aligned} E[N^T U_s \Sigma_s^{-1} V_s^T N^T] &= \sum_{i=1}^{p} \frac{1}{\sigma_{si}} E[N^T u_{si} v_{si}^T N^T] \\ &= \sum_{i=1}^{p} \frac{\sigma^2}{\sigma_{si}} v_{si} u_{si}^T \\ &= \sigma^2 V_s \Sigma_s^{-1} U_s^T, \end{aligned} \tag{24}$$

where σ_{si} is the ith singular value of A and u_{si} and v_{si} are the ith columns of U_s and V_s, respectively. Substituting (23) (and (24) if N is real-valued) into (22) yields

$$E[Q_2] = 0. \tag{25}$$

The expectations of $Q_1^H Q_1$ and $Q_1 Q_1^H$ are determined in a similar fashion to be

$$E[Q_1^H Q_1] = \frac{\sigma^2}{x} I, \quad \text{and} \quad E[Q_1 Q_1^H] = q\sigma^2 \Sigma_s^{-2}, \tag{26}$$

where x equals the sum of the squares of the singular values of A. Using (25) and (26), the expected values of the first- and second-order expressions (16) and (17) can be computed to yield

$$\begin{aligned} E[P_1] &= U_\perp U_\perp^H + q\sigma^2 U_s \Sigma_s^{-2} U_s^H \stackrel{\text{def}}{=} \bar{P}_1(\sigma) \\ E[P_2] &= U_\perp U_\perp^H \left(1 - \frac{\sigma^2}{x}\right) + q\sigma^2 U_s \Sigma_s^{-2} U_s^H \stackrel{\text{def}}{=} \bar{P}_2(\sigma). \end{aligned} \tag{27}$$

In the next section, we give an example to compare the accuracy of these expressions. The results of this section can be extended to handle noise matrices with correlated elements using results from [9].

4. Examples and discussion. We give two examples in this section to demonstrate the validity of our results. In the first example, we compare the results predicted by (27) with sample expectations in a simulation experiment. In the second example, we compute the distance between the actual perturbed subspace, and the perturbed subspace generated by a first- or second-order perturbation expansion.

4.1. Expected value of a projection matrix. To demonstrate the results of the previous section, consider the following matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} = [U_s \quad U_\perp] \begin{bmatrix} 9.4868 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_s^H \\ V_\perp^H \end{bmatrix},$$

where

$$U_s = \begin{bmatrix} 0.5774 \\ 0.5774 \\ 0.5774 \end{bmatrix}, \quad \text{and} \quad U_\perp = \begin{bmatrix} -0.8165 & 0.0000 \\ 0.4082 & 0.7071 \\ 0.4082 & -0.7071 \end{bmatrix}.$$

The projection matrices for the signal and orthogonal subspaces are

$$P_s = U_s U_s^T = \begin{bmatrix} 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 \end{bmatrix}$$

and

$$P_\perp = U_\perp U_\perp^T = \begin{bmatrix} 0.6667 & -0.3333 & -0.3333 \\ -0.3333 & 0.6667 & -0.3333 \\ -0.3333 & -0.3333 & 0.6667 \end{bmatrix}.$$

Suppose that instead of the matrix A, only the perturbed matrix

$$\tilde{A} = A + \sigma N \tag{28}$$

were available, where the elements of N are i.i.d. Gaussian random variables with zero-mean and unit variance. An SVD for $\tilde{A}$ gives bases for the estimated signal and orthogonal subspaces

$$\tilde{A} = [\tilde{U}_s \quad \tilde{U}_\perp]\tilde{\Sigma}\tilde{V}^T, \tag{29}$$

and the projection matrix for the estimated orthogonal subspace is

$$\tilde{P}_\perp = \tilde{U}_\perp \tilde{U}_\perp^T. \tag{30}$$

The following simulation was performed. Values of σ from 0.2–2 in increments of 0.2 were generated and used to form matrices $\tilde{A}$ as in (28). For each value of σ, 10,000 realizations of the noise matrix N were generated, and the corresponding projection matrices were computed using (29) and (30). The estimated projection matrices were averaged to produce a matrix $\bar{\tilde{P}}_\perp(\sigma)$ for each value of σ. The experimentally determined matrices $\bar{\tilde{P}}_\perp(\sigma)$ were compared with the theoretical expressions in (27) by computing the following error norms:

$$e_1(\sigma) = \|\bar{\tilde{P}}_\perp(\sigma) - \bar{P}_1(\sigma)\|_2,$$

$$e_2(\sigma) = \|\bar{\tilde{P}}_\perp(\sigma) - \bar{P}_2(\sigma)\|_2.$$

The error norms $e_1(\sigma)$ and $e_2(\sigma)$ result from using the first- and second-order perturbation expansions, respectively, for the perturbed orthogonal subspace. For comparison, we also consider an error norm corresponding to a zeroth-order perturbation expansion, namely,

$$e_0(\sigma) = \|\bar{\tilde{P}}_\perp(\sigma) - P_\perp\|_2.$$

Note that the error norms $e_i(\sigma)$ are the distances between the actual average perturbed subspace and the average subspace predicted by a perturbation expansion of order i.

We can gain some insight into the functional form of the error norms by looking at (16) and (17). Specifically, we see from (16) that the zeroth-order projection matrix $U_\perp U_\perp^T$ does not model terms that are first- and second-order in N (recall that Q_1 is linear in N). However, since first-order terms in N have zero mean, the zeroth-order projection matrix incurs errors of second order. From the manner in which the data

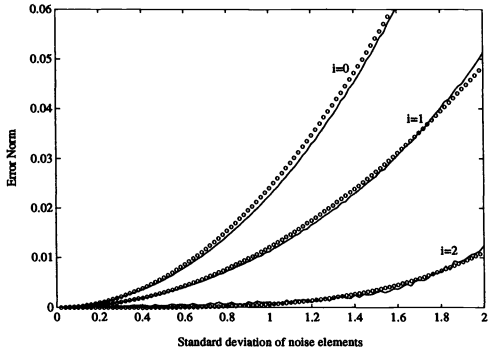


FIG. 1. *The error norms $e_i(\sigma)$ defined by* (55) *and* (56) *for $i = 1, 2, 3$. The solid lines are experimentally computed curves; the circles are least-squares polynomial fits.*

is constructed in (28), we see that the norm of N is proportional to σ. Thus the functional form of $e_0(\sigma)$ is essentially quadratic in σ

$$e_0(\sigma) = \alpha_0 \sigma^2 + O(\sigma^4).$$

In a similar way we can compare (17) and (16) and see that the first-order approximation P_1 contains some but not all of the terms that are second order in N. Thus $e_1(\sigma)$ will also be approximately a quadratic function of σ

$$e_1(\sigma) = \alpha_1 \sigma^2 + O(\sigma^4).$$

We would suspect that $\alpha_1 < \alpha_0$ since P_1 contains some second-order terms while P_0 does not.

Finally, we recall that P_2 is exact up to second-order terms. Since third- and fifth-order terms have zero mean, the error norm $e_2(\sigma)$ should be approximately proportional to σ^4

$$e_2(\sigma) = \alpha_2 \sigma^4 + O(\sigma^6).$$

The above conclusions on the functional form of the error norms are verified in the simulation results in Fig. 1. In this figure, the solid lines are the experimentally computed curves $e_i(\sigma)$ for $i = 0, 1, 2$. The circles show the functional forms deduced above with the optimal amplitudes. The amplitudes for a least-squares fit were found to be

$$\alpha_0 = 2.4 \times 10^{-2} \quad \text{quadratic},$$

$$\alpha_1 = 1.2 \times 10^{-2} \quad \text{quadratic},$$

$$\alpha_2 = 7.0 \times 10^{-4} \quad \text{quartic}.$$

We can draw several conclusions from Fig. 1. First we see that both $e_1(\sigma)$ and $e_2(\sigma)$ are proportional to σ^2. The only difference is that $e_2(\sigma)$ has a smaller constant.

Thus the advantage of using a first-order perturbation expansion over a zeroth-order expansion is a scale factor in the error norm, which for this example equals 0.5. However, since $e_2(\sigma)$ is proportional to σ^4, there is a substantial benefit to using a second-order expansion. For example, $e_2(\sigma) < 0.01$ for $\sigma < 1.9$ whereas the same error bound for $e_1(\sigma)$ requires $\sigma < 0.92$, and for $e_1(\sigma)$ requires $\sigma < 0.66$. We also see from (27) that the expressions resulting from a second-order expansion are not necessarily more complicated than those from a first-order expansion.

4.2. Distance between subspaces. For each realization of the noise matrix N, the results of a first- or second-order perturbation expansion can be used to obtain a basis for the perturbed orthogonal (or signal) subspace. That is

$$\tilde{U}_\perp \overset{0}{=} \operatorname{span}(U_\perp) \overset{\text{def}}{=} \tilde{U}_\perp^0,$$

$$\tilde{U}_\perp \overset{1}{=} \operatorname{span}(U_\perp + U_s Q_1) \overset{\text{def}}{=} \tilde{U}_\perp^1,$$

$$\tilde{U}_\perp \overset{2}{=} \operatorname{span}(U_\perp + U_s Q_2) \overset{\text{def}}{=} \tilde{U}_\perp^2,$$

where Q_1 and Q_2 are given as functions of the noise matrix N in (10) and (14), respectively.

For each realization of N, a basis for the actual orthogonal subspace $\tilde{U}_\perp$ can be obtained from an SVD of $\tilde{A}$. In this section we compute the distance between the actual perturbed subspaces computed by an SVD of $\tilde{A}$ and the subspaces generated by perturbation expansions of order zero, one, and two. These distances are denoted by

$$\text{(31)} \qquad \operatorname{dist}_i(\sigma) \overset{\text{def}}{=} \operatorname{dist}(\tilde{U}_\perp, \tilde{U}_\perp^i), \qquad i = 0, 1, 2.$$

The distance between two subspaces S_1 and S_2 is defined to be [1]

$$\operatorname{dist}(S_1, S_2) \overset{\text{def}}{=} \|P_1 - P_2\|,$$

where P_i is the orthogonal projection matrix onto S_i.

We plot $E[\operatorname{dist}_i(\sigma)]$ versus σ in Fig. 2. Since the zeroth-order expansion neglects linear terms, we expect $E[\operatorname{dist}_0(\sigma)]$ to be linear. Since first- and second-order expansions neglect second- and third-order terms, respectively, we expect $E[\operatorname{dist}_1(\sigma)]$ to be quadratic and $E[\operatorname{dist}_2(\sigma)]$ to be cubic. That is in fact what we see in Fig. 2. The circles show the functional forms deduced above with the optimal amplitudes. The amplitudes for a least-squares fit were found to be

$$\alpha_0 = 1.4 \times 10^{-1} \quad \text{linear},$$

$$\alpha_1 = 2.8 \times 10^{-2} \quad \text{quadratic},$$

$$\alpha_2 = 7.2 \times 10^{-3} \quad \text{cubic}.$$

4.3. Discussion. We are currently using the second-order expansion to analyze the performance of sensor array processing algorithms for direction-of-arrival estimation. The analysis in [2]–[6] used the first-order expansion. By using the second-order expansion derived in this paper, we hope to be able to develop performance expressions that are accurate at lower signal-to-noise ratios and that predict bias.

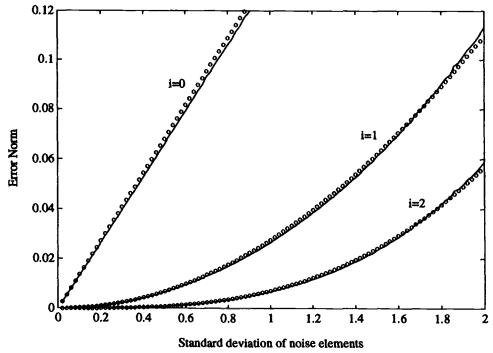


FIG. 2. *The distances between the actual perturbed subspace and the subspaces generated by perturbation expansions of orders* $i = 0, 1, 2$. *The solid lines are experimentally computed curves; the circles are least-squares polynomial fits.*

REFERENCES

[1] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[2] F. LI AND R. J. VACCARO, *MUSIC performance prediction by matrix approximation at high SNR*, in Proc. 1989 Conf. Information Sciences and Systems, The Johns Hopkins University, Baltimore, MD, March 1989, pp. 447–481.

[3] ———, *Analysis of MUSIC and Min-Norm for Arbitrary Array Geometry*, IEEE Trans. Aerosp., Electron. Syst., AES-26 (1990), pp. 976–985.

[4] ———, *SVD and Signal Processing*, II, Analytical Performance Prediction of Subspace-Based Algorithms for DOA Estimation, R.J. Vaccaro, ed., Elsevier Science Publishers, 1991, pp. 243–260.

[5] ———, *Unified analysis for DOA estimation algorithms in array signal processing*, Signal Processing, 25 (1991), pp. 147–169.

[6] ———, *Sensitivity analysis of DOA estimation algorithms to sensor errors*, IEEE Trans. Aerosp., Electron. Syst., AES-27 (1992), pp. 708–717.

[7] A. A. SHAH AND D. W. TUFTS, *Estimation of the signal component of a data vector*, in Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing, March, 1992, pp. 393–396.

[8] G. W. STEWART, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15-33 (1973), pp. 727–764.

[9] ———, *Stochastic perturbation theory*, SIAM Rev., 32 (1990), pp. 579–610.

[10] D. W. TUFTS, A. C. KOT, AND R. J. VACCARO, *SVD and Signal Processing*, II, The Threshold Effect in Signal Processing Algorithms Which Use an Estimated Subspace, R.J. Vaccaro, ed., Elsevier Science Publishers, New York, 1991, pp. 300–321.

[11] R. J. VACCARO AND Y. DING, *Optimal subspace-based parameter estimation*, in Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing, Minneapolis, MN, 1993, pp. IV-368–IV-371.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 672–691, April 1994

020

PROBABILISTIC BOUNDS ON THE EXTREMAL EIGENVALUES AND CONDITION NUMBER BY THE LANCZOS ALGORITHM*

J. KUCZYŃSKI† AND H. WOŹNIAKOWSKI‡

Abstract. The authors analyze the Lanczos algorithm with a random start for approximating the extremal eigenvalues of a symmetric positive definite matrix. They present some bounds on the Lebesgue measure (probability) of the sets of these starting vectors for which the Lanczos algorithm gives at the kth step satisfactory approximations to the largest and smallest eigenvalues. Combining these bounds gets similar estimates for the condition number of a matrix.

Key words. extreme eigenvalues, Lanczos algorithm, condition number, random start

AMS subject classification. 65

1. Introduction.

There are many algorithms for approximating the extremal eigenvalues of $n \times n$ symmetric positive definite matrices. Algorithms based on a factorization of A usually require $\Theta(n^3)$ arithmetic operations and if n is large then such algorithms are too expensive. On the other hand, large matrices are usually sparse and the matrix-vector multiplication Av can be performed cheaply. This suggests that algorithms based on vectors Av_i for some vectors v_i may be efficient.

The Lanczos algorithm is probably the most popular algorithm that uses a few vectors Av_i for approximating eigenvalues. In particular, at the kth step it gives approximations to the largest eigenvalue λ_1 and to the smallest eigenvalue λ_n of A as the maximal and minimal Rayleigh quotient $(Ax, x)/(x, x)$ over nonzero vectors x from the kth Krylov subspace $A_k = \text{span}(b, Ab, \ldots, A^{k-1}b)$, where $b \neq 0$. Of course, the quality of such approximations strongly depends both on the matrix A and on the starting vector b. It is known (see [15]) that a poor choice of the vector b can cause a bad behaviour of the Lanczos algorithm. The analysis of the Lanczos algorithm for a fixed vector b may be found in many books and papers, particularly, [2],[5],[7],[8],[10]–[12], and [14]–[16]. The analysis of the Lanczos algorithm for a random vector b can be found in [9], where average case and probabilistic estimates on the largest eigenvalue are given. These estimates are independent of the distribution of the eigenvalues of A.

In §2, we first translate the estimates of [9] for approximating the smallest eigenvalue λ_n of A. We estimate the average relative error of the Lanczos algorithm over all starting vectors b. We also provide an upper bound on the Lebesgue measure (probability) of the set of those b's from the unit ball for which the Lanczos algorithm fails to give an ε-approximation to the smallest eigenvalue at the kth step.

In §3, we present new probabilistic estimates for the Lanczos algorithm for approximating the largest and smallest eigenvalue of A. These estimates depend on the distribution of the eigenvalues of A.

In §4, we apply estimates on the smallest and largest eigenvalue to approximate the condition number, $\text{cond}\, A$, of A in the two-norm,

$$\text{cond}\, A = \frac{\lambda_1}{\lambda_n}.$$

* Received by the editors February 25, 1992; accepted for publication (in revised form) July 16, 1992.

† Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland.

‡ Department of Computer Science, Columbia University, New York, New York 10027 (henryk@ground.cs.columbia.edu) and Institute of Applied Mathematics, University of Warsaw, Warsaw, Poland. Supported in part by National Science Foundation contract IRI-89-07215.

An approximation to the condition number of a matrix is often wanted when one deals with matrix calculations; see [1] and [4]. We provide bounds on the probability of the set of these b's for which the Lanczos algorithm gives at the kth step the approximation κ_k of the condition number of A such that

$$\kappa_k \leq \operatorname{cond} A \leq \alpha \kappa_k$$

for any $\alpha > 1$. In many cases, only a rough approximation of $\operatorname{cond} A$ is needed. Then α may be quite large, say, $\alpha = 10$.

A similar problem of approximating the condition number has been considered by Dixon [3] for arbitrary matrices and by using modified power and inverse power methods. His results are computationally applicable if the vector $A^{-1}z$ can be computed cheaply for any vector z. This is the case when a factorization of A is given. However, if n is large and the matrix A is given only via the subroutine performing a matrix-vector multiplication, the vector $A^{-1}z$ may be approximated by an iterative method but its computation may be expensive. In this case, the Lanczos algorithm is preferable since it does not use the vector $A^{-1}z$.

2. Lanczos algorithm for approximating extremal eigenvalues. Let $A = A^T > 0$ be an $n \times n$ symmetric positive definite matrix with eigenvalues $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A) > 0$. The kth step of the Lanczos algorithm produces approximations to the largest and smallest eigenvalues of A defined as

$$\hat{\xi}_k = \hat{\xi}(A, b, k) = \max \left\{ \frac{(Ax, x)}{(x, x)} : x \in A_k, x \neq 0 \right\}$$

and

$$\xi_k = \xi(A, b, k) = \min \left\{ \frac{(Ax, x)}{(x, x)} : x \in A_k, x \neq 0 \right\},$$

where $A_k = \operatorname{span}(b, Ab, \ldots, A^{k-1}b)$ for some nonzero vector b. Without loss of generality we may assume that $b \in S_n = \{b \in \mathcal{R}^n : ||b|| = 1\}$, where $||\cdot||$ is the Euclidean norm of vectors.

First we discuss the error of the Lanczos approximation ξ_k to the smallest eigenvalue λ_n. The most natural error criterion is probably the relative error $(\xi_k - \lambda_n)/\lambda_n$. It is shown in Remark 1, §6, that unfortunately even the *average* value of

$$\frac{\xi(A, b, k) - \lambda_n(A)}{\lambda_n(A)},$$

with respect to the uniform distribution of b, is arbitrary large for some matrices A.

Therefore we have to switch to a different error criterion. One may consider the relative error with respect to the largest eigenvalue $(\xi_k - \lambda_n)/\lambda_1$, or the gap ratio $(\xi_k - \lambda_n)/(\lambda_1 - \lambda_n)$. In Remark 2, §6, it turns out that the results both for the average error and probabilistic failure are the same for these two error criteria. Therefore we only consider the relative error with respect to the largest eigenvalue, i.e.,

$$e(A, b, k) = \frac{\xi(A, b, k) - \lambda_n(A)}{\lambda_1(A)}.$$

It is easy to see that there exist vectors b and matrices A for which the error $e(A, b, k)$ can be arbitrary close to 1. Indeed, $\hat{\xi}(A, b, k) \leq \lambda_1(A)$ and $\lambda_n(A) \leq \xi(A, b, k)$ and therefore

$$e(A, b, k) < \xi(A, b, k)/\hat{\xi}(A, b, k) \leq 1.$$

In general, this bound cannot be improved since if b is an eigenvector belonging to the largest eigenvalue $\lambda_1(A)$, then for all k, $\xi(A,b,k) = \hat{\xi}(A,b,k) = \lambda_1(A)$, and yet $\lambda_n(A)$ can be arbitrary close to 0.

Thus, we cannot guarantee that the error of the Lanczos algorithm is smaller than 1 for all matrices A and all vectors b. Moreover, if b is orthogonal to the eigenspace generated by the smallest eigenvalue, then the Lanczos error $e(A,b,k) > 0$ for all k. On the other hand, if b is not orthogonal to this subspace, then $e(A,b,k) = 0$ for $k \geq m$, where m denotes the number of distinct eigenvalues of A.

It is clear that if we pick the vector b randomly according to the uniform distribution over the unit sphere, then with probability one b is *not* orthogonal to the eigenspace of λ_n. This suggests that for any symmetric positive definite matrix, the average relative error of the Lanczos algorithm with respect to vectors b should be small for large k. Also, the measure of the set of these b for which the Lanczos algorithm fails to compute a good approximation to the smallest eigenvalue should be small.

Let μ be the uniform distribution over the unit sphere S_n with $\mu(S_n) = 1$. By the average relative error of the kth step of the Lanczos algorithm, we mean

$$e^{\text{avg}}(A,k) = \int_{S_n} e(A,b,k)\, \mu(db).$$

Let

$$f^{\text{prob}}(A,k,\varepsilon) = \mu\{b \in S_n : e(A,b,k) > \varepsilon\}$$

denote the probability that the kth step of the Lanczos algorithm fails to approximate the smallest eigenvalue with relative error at most ε. For brevity we call this the probabilistic failure.

We now present an upper bound on the average error of the Lanczos algorithm. It turns out that this bound is the same as the one obtained in [9] for the Lanczos algorithm approximating the largest eigenvalue λ_1. For simplicity, as in [9], we assume that $n \geq 8$ and $k \geq 4$.

THEOREM 1. *Let A be a symmetric positive definite matrix.*

(a) *Let m denote the number of distinct eigenvalues of A. Then for $k \geq m$,*

$$e^{\text{avg}}(A,k) = 0,$$

for $k \in [4, m-1]$,

$$e^{\text{avg}}(A,k) \leq 0.103 \left(\frac{\ln\left(n(k-1)^4\right)}{k-1}\right)^2 \leq 2.575 \left(\frac{\ln n}{k-1}\right)^2.$$

(b) *Let $p, p < n$, denote the multiplicity of the smallest eigenvalue λ_n, and let λ_{n-p} be the second smallest eigenvalue of A. Then*

$$e^{\text{avg}}(A,k) \leq 2.589\sqrt{n} \left(\frac{1 - \sqrt{(\lambda_{n-p} - \lambda_n)/(\lambda_1 - \lambda_n)}}{1 + \sqrt{(\lambda_{n-p} - \lambda_n)/(\lambda_1 - \lambda_n)}}\right)^{k-1}.$$

Proof. For the proof it is enough to apply Theorem 3.2 of [9] to the matrix $B = \lambda_1 I - A$. □

Theorem 1 states that $e^{\text{avg}}(A,k) = 0$ for $k \geq m$, which means that the Lanczos algorithm converges in $m, m \leq n$, steps. For $k < m$ the average relative error is bounded by $2.6 \ln^2 n/(k-1)^2$. We do not know if this bound is sharp. Numerical tests suggest that for some matrices A we have

$$e^{\text{avg}}(A,k) = \Theta(k^{-2}).$$

Part (b) of Theorem 1 contains nonasymptotic bounds in terms of the gap ratio of the matrix A.

We now turn to the bound on the probabilistic failure of the Lanczos algorithm for the smallest eigenvalue.

THEOREM 2. *Let A be a symmetric positive definite matrix.*

(a) *Let m denote the number of distinct eigenvalues of A. Then for any $\varepsilon \in (0,1]$,*

$$f^{\text{prob}}(A,k,\varepsilon) = 0 \quad \textit{for } k \geq m,$$

$$f^{\text{prob}}(A,k,\varepsilon) \leq 1.648\sqrt{n}\, e^{-(2k-1)\sqrt{\varepsilon}} \quad \textit{for any } k\,.$$

(b) *Let $p, p < n$, denote the multiplicity of the smallest eigenvalue λ_n and let λ_{n-p} be the second smallest eigenvalue of A. Then for $\varepsilon > 0$,*

$$f^{\text{prob}}(A,k,\varepsilon) \leq 1.648\sqrt{n/\varepsilon}\left(\frac{1-\sqrt{(\lambda_{n-p}-\lambda_n)/(\lambda_1-\lambda_n)}}{1+\sqrt{(\lambda_{n-p}-\lambda_n)/(\lambda_1-\lambda_n)}}\right)^{k-1}.$$

Proof. For the proof it is enough to apply Theorem 4.2 of [9] for the matrix $B = \lambda I - A$. □

Theorem 2 gives upper bounds on the probabilistic failure of the Lanczos algorithm for approximating the smallest eigenvalue. For $k \geq m$, $f^{\text{prob}}(A,k,\varepsilon) = 0$, which means that the Lanczos algorithm converges in m steps, where $m \leq n$. For $k < m$, the probabilistic failure is bounded by $1.65\sqrt{n}\, e^{-(2k-1)\sqrt{\varepsilon}}$.

Suppose we wish to find an ε-approximation to the smallest eigenvalue with a δ-failure, i.e., $f^{\text{prob}}(A,k,\varepsilon) \leq \delta$. Then from Theorem 2 we conclude that we have to perform at most roughly

$$k = \lceil \ln(3n/\delta^2) \,/\, (4\sqrt{\varepsilon}) \rceil$$

steps. Note a weak dependence on δ and the strong dependence on ε.

3. Bounds dependent on distribution of eigenvalues. In this section we provide new probabilistic bounds on extemal eigenvalues that depend on distribution of eigenvalues of the matrix A. For any symmetric positive definite matrix A, consider two sets

$$L_k = \{b \in S_n : \hat{\xi}(A,b,k) \leq \lambda_1(A) \leq \theta\hat{\xi}(A,b,k)\} \quad \text{for} \quad \theta > 1$$

and

$$M_k = \{b \in S_n : \eta\,\xi(A,b,k) \leq \lambda_n(A) \leq \xi(A,b,k)\} \quad \text{for} \quad \eta < 1.$$

Obviously, the first inequality in the set L_k and the second one in M_k hold for any vector $b \in S_n$. We now find lower bounds on $\mu(L_k)$ and $\mu(M_k)$. We begin with $\mu(L_k)$.

THEOREM 3. *For any symmetric positive definite matrix A let m denote the number of distinct eigenvalues of A. Then for $k \geq m$, or $\theta \geq \operatorname{cond} A$,*

$$\mu(L_k) = 1,$$

and for $k < m$ and $1 < \theta < \operatorname{cond} A$,

$$\begin{aligned}\mu(L_k) &\geq 1 - \frac{2\,\Gamma((n-j)/2+1)}{\sqrt{\pi}\,\Gamma((n-j+1)/2)} \frac{1}{\sqrt{\theta-1}\,U_{2(k-1)}(\sqrt{\theta})} \\ &\geq 1 - \frac{2\,\Gamma(n/2)}{\sqrt{\pi}\,\Gamma((n-1)/2)} \frac{1}{\sqrt{\theta-1}\,U_{2(k-1)}(\sqrt{\theta})},\end{aligned}$$

where the index j is defined by $\lambda_j \leq \lambda_1/\theta < \lambda_{j-1}$, $2 \leq j \leq n$, U_k is the Chebyshev polynomial of the second kind of degree k, and Γ is the Euler Γ-function.

Proof. The idea of the proof is similar to the proof of Theorem 4.2 in [9]. Let $b = \sum_{i=1}^n b_i v_i$, where $v_i, i = 1, \ldots, n$ are orthonormal eigenvectors of A. Consider the set

$$L_k' = S_n - L_k = \{b \in S_n : \lambda_1(A) > \theta \hat{\xi}(A, b, k)\}.$$

Then we have

$$L_k' = \left\{ b \in S_n : \lambda_1 > \theta \sup_{P \in \mathcal{P}_k} \frac{\sum_{i=1}^n \lambda_i P^2(\lambda_i) b_i^2}{\sum_{i=1}^n b_i^2 P^2(\lambda_i)} \right\},$$

where $\mathcal{P}_k$ denotes the set of nonzero polynomials of degree smaller than k.

Assume that $k \geq m$. Then the set $\{\lambda_1, \ldots, \lambda_n\}$ contains m distinct elements $\{t_1, \ldots, t_m\}$, $\lambda_1 = t_1 > \cdots > t_m$ and for the polynomial $P(x) = \prod_{i=2}^m (x - t_i)$ the supremum takes the value λ_1 for $b_1 \neq 0$. Thus, since $\theta \geq 1$, the set L_k' is empty with probability one. Thus $\mu(L_k) = 1$ as claimed.

Assume now that $k < m$. Then after simple calculations, we get

$$L_k' = \left\{ b \in S_n : \sup_{P \in \mathcal{P}_k} \frac{\sum_{i=1}^n b_i^2 P^2(\lambda_i)(\theta\lambda_i - \lambda_1)}{\sum_{i=1}^n b_i^2 P^2(\lambda_i)} < 0 \right\}.$$

Observe that the supremum is negative if and only if the enumerator is negative. Thus

$$L_k' = \left\{ b \in S_n : \sup_{P \in \mathcal{P}_k} \sum_{i=1}^n b_i^2 P^2(\lambda_i)(\theta\lambda_i - \lambda_1) < 0 \right\}.$$

Assume that $\theta \geq \operatorname{cond} A$. Then $\theta\lambda_i - \lambda_1 \geq 0$ for all $i = 1, 2, \ldots, m$ and the set L_k' is empty. Thus $\mu(L_k) = 1$ as claimed.

Assume now that $\theta < \operatorname{cond} A$. Note that using a continuity argument, we may restrict ourselves to polynomials for which $P(\lambda_1) \neq 0$. Let $x_i = \lambda_i/\lambda_1$ and $Q(x) = P(\lambda_1 x)/P(\lambda_1)$. Then $x_i \in (0, 1]$ and

$$L_k' = \left\{ b \in S_n : \sup_{Q \in \mathcal{P}_k(1)} \sum_{i=1}^n b_i^2 Q^2(x_i)(\theta x_i - 1) < 0 \right\},$$

where $\mathcal{P}_k(\alpha) = \{w \in \mathcal{P}_k : w(\alpha) = 1\}$.

Clearly, for any polynomial $Q \in \mathcal{P}_k(1)$, we have

$$L_k' \subset \left\{ b \in S_n : \sum_{i=1}^{n} b_i^2 Q^2(x_i)(\theta x_i - 1) < 0 \right\}.$$

From the definition of the index j, we have $\theta x_j \leq 1 < \theta x_{j-1}$ and

$$\begin{aligned}
\mu(L_k') &\leq \mu \left\{ b \in S_n : \sum_{i=j}^{n} b_i^2 Q^2(x_i)(1 - \theta x_i) > \sum_{i=1}^{j-1} b_i^2 Q^2(x_i)(\theta x_i - 1) \right\} \\
&\leq \mu \left\{ b \in S_n : \sum_{i=j}^{n} b_i^2 Q^2(x_i)(1 - \theta x_i) > b_1^2(\theta - 1) \right\} \\
&\leq \mu \left\{ b \in S_n : \sum_{i=j}^{n} b_i^2 \max_{x \in [0, x_j]} [Q^2(x)(1 - \theta x)] > b_1^2(\theta - 1) \right\}.
\end{aligned}$$

Let

$$\omega_k(x_j, \theta) = \inf_{Q \in \mathcal{P}_k(1)} \max_{x \in [0, x_j]} Q^2(x)(1 - \theta x).$$

We now find an upper bound on $\omega_k(x_j, \theta)$. It is easy to check that

$$\omega_k(x_j, \theta) = \inf_{Q \in \mathcal{P}_k(\theta)} \max_{x \in [0, \gamma]} Q^2(x)(1 - x),$$

where $\gamma = \theta x_j \leq 1$. Let

$$\hat{Q}(x) = \frac{U_{2(k-1)}(\sqrt{x})}{U_{2(k-1)}(\sqrt{\theta})},$$

where $U_{2(k-1)}$ is the Chebyshev polynomial of the second kind of degree $2(k-1)$. Since $U_{2(k-1)}$ is even, $\hat{Q}$ is a polynomial of degree $k-1$. Obviously $\hat{Q}(\theta) = 1$, so $\hat{Q} \in \mathcal{P}_k(\theta)$. Then

$$\omega_k(x_j, \theta) \leq \max_{x \in [0, \gamma]} \frac{U_{2(k-1)}^2(\sqrt{x})}{U_{2(k-1)}^2(\sqrt{\theta})} (1 - (\sqrt{x})^2) = \frac{1}{U_{2(k-1)}^2(\sqrt{\theta})} =: \sigma,$$

since $U_{2(k-1)}(t)\sqrt{1-t^2} \leq 1$ and this inequality is sharp (see, e.g., [13]). We note in passing that if $\gamma > \cos^2(\pi/(2(2k-1)))$, then $\omega_k(x_j, \theta) = \sigma$ (see the proof of Theorem 4.2 of [9]). Clearly,

$$\mu(L_k') \leq \mu \left\{ b \in S_n : \sum_{i=j}^{n} b_i^2 > \beta^{-2} b_1^2 \right\},$$

where $\beta^2 = \sigma/(\theta - 1)$.

Let c_i be the Lebesgue measure of the unit ball in $\mathcal{R}^i$, $c_i = \pi^{i/2}/\Gamma(1 + i/2)$. As in [9], Remark 7.2, instead of integrating over the unit sphere S_n, we may integrate over the unit ball $||b|| \leq 1$ with respect to normalized Lebesgue measure,

$$\mu \left\{ b \in S_n : \sum_{i=j}^{n} b_i^2 > \beta^{-2} b_1^2 \right\} = \frac{1}{c_n} \int_{||b|| \leq 1} \chi \left(\sum_{i=j}^{n} b_i^2 > \beta^{-2} b_1^2 \right) db,$$

where $\chi(Z)$ is the characteristic function of the set Z. Then

$$\mu(L_k') \le \frac{1}{c_n} \int_{[-1,1]} \int_{\sum_{i=j}^n b_i^2 > \beta^{-2} b_1^2} \int_{\sum_{i=2}^{j-1} b_i^2 \le 1 - b_1^2 - \sum_{i=j}^n b_i^2} db \, .$$

The last integral is the measure of the ball in $\mathcal{R}^{j-2}$, and

$$\begin{aligned}
\mu(L_k') &\le \frac{2\,c_{j-2}}{c_n} \int_{[0,1]} \int_{\sum_{i=j}^n b_i^2 > \beta^{-2} b_1^2} \left(1 - b_1^2 - \sum_{i=j}^n b_i^2\right)_+^{(j-2)/2} \\
&= \frac{2\,c_{j-2}}{c_n} \int_{[0,1]} \int_{\sum_{i=j}^n b_i^2 < 1} \chi\left(\sum_{i=j}^n b_i^2 > \beta^{-2} b_1^2\right) \left(1 - b_1^2 - \sum_{i=j}^n b_i^2\right)_+^{(j-2)/2}
\end{aligned}$$

Changing variables by $t = \sqrt{\sum_{i=j}^n b_i^2}$ and $x = b_1$, formula 4.642 of [6] yields

$$\mu(L_k') \le \frac{2\,c_{j-2}\,c_{n-j+1}}{c_n}(n-j+1)$$

$$\cdot \int_0^1 \int_0^1 \chi(t^2 > \beta^{-2} x^2)\,(1 - x^2 - t^2)_+^{(j-2)/2}\, t^{n-j}\, dx\, dt$$

$$\begin{aligned}
&\le \frac{2\,c_{j-2}\,c_{n-j+1}}{c_n}(n-j+1) \int_0^1 \int_0^{\beta t} (1 - x^2 - t^2)_+^{(j-2)/2}\, t^{n-j}\, dx\, dt \\
&\le \frac{2\,c_{j-2}\,c_{n-j+1}}{c_n}(n-j+1)\beta \int_0^1 (1-t^2)^{j/2-1}\, t^{n-j+1}\, dt \\
&= \frac{c_{j-2}\,c_{n-j+1}}{c_n}(n-j+1)\beta B((n-j+2)/2, j/2),
\end{aligned}$$

where $B(i,j) = \Gamma(i)\Gamma(j)/\Gamma(i+j)$ is the beta function. Finally,

$$\mu(L_k') \le \frac{\beta\,(n-j+1)\Gamma((n-j)/2+1)}{\sqrt{\pi}\,\Gamma((n-j+1)/2+1)}$$

and

$$\mu(L_k') \le \frac{2\Gamma((n-j)/2+1)}{\sqrt{\pi}\,\Gamma((n-j+1)/2)\,\sqrt{\theta-1}\,U_{2(k-1)}(\theta)},$$

which completes the first inequality of Theorem 3.

Let

$$s_j = \frac{2\Gamma((n-j)/2+1)}{\sqrt{\pi}\,\Gamma((n-j+1)/2)}.$$

Then

$$\mu(L_k) \ge 1 - s_j/\sqrt{\theta-1}/U_{2(k-1)}(\sqrt{\theta}),$$

and since s_j is a decreasing function of j, the proof is completed. □

We now explain how Theorem 3 is related to Theorem 4.2 of [9]. Observe that using formula (13) of [9] for $n \geq 8$, we get

$$s_n = \frac{2}{\pi} < s_{n-1} = 1 < \cdots < s_2 = \frac{2\Gamma(n/2)}{\sqrt{\pi}\,\Gamma((n-1)/2)} \leq 0.824\sqrt{n-1}.$$

Take $j = 2$. Then from Theorem 3 it follows that for any $\theta > 1$, we have

$$\mu\{b \in S_n : \lambda_1(A) \leq \theta\,\hat{\xi}(A,b,k)\} \geq 1 - 0.824\,\frac{\sqrt{n-1}}{\sqrt{\theta-1}\,U_{2(k-1)}(\sqrt{\theta})}.$$

Let $\theta = 1/(1-\varepsilon)$ for some $\varepsilon \in (0,1)$. Then

$$\mu(L_k) = \mu\left\{b \in S_n : \frac{\lambda_1(A) - \hat{\xi}(A,b,k)}{\lambda_1(A)} < \varepsilon\right\}$$
$$\geq 1 - 0.824\,\frac{\sqrt{(n-1)\,\varepsilon}}{\sqrt{1-\varepsilon}\,U_{2(k-1)}(1/\sqrt{1-\varepsilon})}.$$

Observe that

$$U_{2(k-1)}(1/\sqrt{1-\varepsilon}) = 0.5\sqrt{(1-\varepsilon)/\varepsilon}\left(\left(\frac{1+\sqrt{\varepsilon}}{\sqrt{1-\varepsilon}}\right)^{2k-1} - \left(\frac{1-\sqrt{\varepsilon}}{\sqrt{1-\varepsilon}}\right)^{2k-1}\right),$$

and after some calculations, we obtain

$$U_{2(k-1)}(1/\sqrt{1-\varepsilon}) = 0.5\,\sqrt{(1-\varepsilon)/\varepsilon}\,c^{-1}\,(1-c^2),$$

where $c = ((1-\sqrt{\varepsilon})/(1+\sqrt{\varepsilon}))^{k-1/2}$. Thus we get

$$\mu\left\{b \in S_n : \frac{\lambda_1(A) - \hat{\xi}(A,b,k)}{\lambda_1(A)} < \varepsilon\right\} \geq 1 - 1.648\,\sqrt{n-1}\,c\,(1-c^2)^{-1}.$$

Recall that Theorem 4.2 of [9] yields

$$\mu\left\{b \in S_n : \frac{\lambda_1(A) - \hat{\xi}(A,b,k)}{\lambda_1(A)} < \varepsilon\right\} \geq 1 - 1.648\,\sqrt{n}\,c \geq 1 - 1.648\,\sqrt{n}\,e^{-(2k-1)\sqrt{\varepsilon}}.$$

Thus for $j = 2$, Theorem 3 gives essentially the same bound as Theorem 4.2. For $j > 2$, the bound of Theorem 3 is better. Usually we do not know the index j. However, for large θ or for many eigenvalues close to the largest eigenvalue λ_1, the index j is large and s_j becomes independent on n. In this case, $\mu(L_k)$ goes quickly to 1 even for very large n.

We now find a lower bound on the probability of the set M_k. Recall that

$$M_k = \{b \in S_n : \eta\,\xi(A,b,k) \leq \lambda_n(A) \leq \xi(A,b,k)\} \quad \text{for} \quad \eta \leq 1.$$

THEOREM 4. *For any symmetric positive definite matrix A, let m denote the number of distinct eigenvalues. Then we have*

$$\mu(M_k) = 1 \quad \textit{for } k \geq m \quad \textit{or} \quad \eta \leq 1/\operatorname{cond} A$$

and for any $k < m$ *and any* $1/\operatorname{cond} A < \eta < 1$,

$$\mu(M_k) \geq 1 - \frac{2\Gamma(j/2)}{\sqrt{\pi}\,\Gamma((j-1)/2)}$$

$$\cdot \sqrt{\frac{\eta \operatorname{cond} A - 1}{1-\eta}}\, U^{-1}_{2(k-1)}\left(\sqrt{1 + (1-\eta)/(\eta \operatorname{cond} A - 1)}\right)$$

$$\geq 1 - \frac{2\Gamma(n/2)}{\sqrt{\pi}\,\Gamma((n-1)/2)} \sqrt{\frac{\eta \operatorname{cond} A - 1}{1-\eta}}\, U^{-1}_{2(k-1)}\left(\sqrt{1 + (1-\eta)/(\eta \operatorname{cond} A - 1)}\right),$$

where the index j *is defined by* $\lambda_n/\lambda_{j-1} < \eta \leq \lambda_n/\lambda_j$, $2 \leq j \leq n$.

Proof. The proof of Theorem 4 is similar to the proof of Theorem 3. As before, we consider the set M_k',

$$M_k' = S_n - M_k = \{b \in S_n : \lambda_n(A) < \eta\, \xi(A, b, k)\}.$$

Using the same notation and reasoning as in the proof of Theorem 3, we can easily get

$$\mu(M_k') = \mu\left\{ b \in S_n : \inf_{Q \in \mathcal{P}_k(1)} \sum_{i=1}^{n} b_i^2 Q^2(x_i)(\eta x_i - 1) > 0 \right\},$$

where now $x_i = \lambda_i/\lambda_n$, $1 = x_n \leq x_{n-1} \leq \cdots \leq x_1 = \operatorname{cond} A$. Clearly, if $k \geq m$ or if $\eta \leq \lambda_n/\lambda_1 = 1/x_1$, then $\mu(M_k') = 0$ and the first part of Theorem 4 is proved.

Assume now that $k < m$ and $1/x_1 < \eta < 1$. We have $1/x_{j-1} < \eta \leq 1/x_j$ and, as in Theorem 3, we can show that

$$\mu(M_k') \leq \mu\left\{ b \in S_n : \sum_{i=1}^{j-1} b_i^2 Q^2(x_i)(\eta x_i - 1) > b_n^2 Q^2(1)(1-\eta) \right\}$$

for any $Q \in \mathcal{P}_k(1)$. Hence

$$\mu(M_k') \leq \mu\left\{ b \in S_n : \sigma \sum_{i=1}^{j-1} b_i^2 > b_n^2(1-\eta) \right\},$$

where

$$\sigma = \inf_{Q \in \mathcal{P}_k(1)} \max_{x \in [1/\eta, \operatorname{cond} A]} (Q^2(x)(\eta x - 1)).$$

Changing variables by $y = (\operatorname{cond} A - x)/(\operatorname{cond} A - 1/\eta)$, we get

$$\sigma = (\xi - 1) \inf_{Q \in \mathcal{P}_k(\delta)} \max_{y \in [0,1]} (Q^2(y)(1-y)),$$

where $\xi = \eta \operatorname{cond} A$ and $\delta = (\xi - \eta)/(\xi - 1) > 1$. Using the estimate of $\omega_k(x_j, \theta)$ from the proof of Theorem 3, we obtain

$$\sigma \leq \frac{\xi - 1}{U^2_{2(k-1)}(\sqrt{\delta})},$$

and consequently

$$\mu(M_k') \le \mu\left\{ b \in S_n : \sum_{i=1}^{j-1} b_i^2 > U_{2(k-1)}^2(\sqrt{\delta})\,(1-\eta)/(\xi-1)\; b_n^2 \right\}.$$

As in the proof of Theorem 3, we thus conclude that

$$\mu(M_k') \le \frac{2\,\Gamma(j/2)}{\sqrt{\pi}\,\Gamma((j-1)/2)} \sqrt{\frac{\eta\,\mathrm{cond}\,A - 1}{1-\eta}}\, U_{2(k-1)}^{-1}\left(\sqrt{1 + \frac{1-\eta}{\eta\,\mathrm{cond}\,A - 1}}\right),$$

which completes the first inequality of Theorem 4. To get the second inequality, it is enough to observe that

$$s_j' = \frac{2\,\Gamma(j/2)}{\sqrt{\pi}\,\Gamma((j-1)/2)} < \frac{2\,\Gamma(n/2)}{\sqrt{\pi}\,\Gamma((n-1)/2)} = s_n'$$

for $j = 2, 3, \ldots, n$. Hence the proof is completed. □

We now show how Theorem 4 is related to Theorem 2. Clearly, for $n \ge 8$, we have

$$s_2' = \frac{2}{\pi} < s_3' = 1 < \cdots < s_n' = \frac{2\,\Gamma(n/2)}{\sqrt{\pi}\,\Gamma((n-1)/2)} \le 0.824\sqrt{n-1}.$$

Take $j = n$. Then from Theorem 4 it follows that for any η: $0 < \eta < 1$, we have

$$\mu\{b \in S_n : \eta\,\xi(A,b,k) \le \lambda_n(A)\} \ge 1 - 0.824\sqrt{(n-1)\,z}\;/U_{2(k-1)}(\sqrt{1+1/z}\,),$$

where $z = (\eta\,\mathrm{cond}\,A - 1)/(1-\eta)$. Let $\eta = 1/(1+\varepsilon\,\mathrm{cond}\,A)$ for some $\varepsilon \in (0,1)$. Then

$$\mu(M_k) = \mu\left\{ b \in S_n : \frac{\xi(A,b,k) - \lambda_n(A)}{\lambda_1(A)} < \varepsilon \right\}$$

$$\ge 1 - 0.824\sqrt{(n-1)(1-\varepsilon - 1/\mathrm{cond}\,A)/\varepsilon}\;/U_{2(k-1)}\left(\sqrt{1 + \varepsilon/(1-\varepsilon - 1/\mathrm{cond}\,A)}\right).$$

Since

$$U_{2(k-1)}\left(\sqrt{1+1/z}\right) = 0.5\sqrt{z}\,c^{-1}\,(1-c^2),$$

where now

$$c = \left(\left(1 - \sqrt{\varepsilon/(1-1/\mathrm{cond}\,A)}\right) / \left(1 + \sqrt{\varepsilon/(1-1/\mathrm{cond}\,A)}\right)\right)^{k-1/2}$$

(see the proof of Theorem 4), we get

$$\mu\left\{ b \in S_n : \frac{\xi(A,b,k) - \lambda_n(A)}{\lambda_1(A)} < \varepsilon \right\} \ge 1 - 1.648\sqrt{n-1}\,c\,(1-c^2)^{-1}.$$

Recall that Theorem 2 yields

$$\mu\left\{ b \in S_n : \frac{\xi(A,b,k) - \lambda_n(A)}{\lambda_1(A)} < \varepsilon \right\} \ge 1 - 1.648\sqrt{n}\left(\frac{1-\sqrt{\varepsilon}}{1+\sqrt{\varepsilon}}\right)^{k-1/2}$$

$$\ge 1 - 1.648\sqrt{n}\,e^{-(2k-1)\sqrt{\varepsilon}}.$$

Thus even for $j = n$, Theorem 4, generally gives a better bound than Theorem 2. Let us stress that if we know that j is small, which is the case for small η or cluster eigenvalues, the bound of Theorem 4 is even more attractive than that of Theorem 2.

The bound given in Theorem 4 depends on the condition number of A, which is not known. We can avoid this difficulty if we know an upper bound of the condition number of the matrix A. The following corollary holds (the proof is the same as the proof of Theorem 4).

COROLLARY 1. *Let M be any number, $M > 1$. For any symmetric positive definite matrix A of size $n \geq 8$ such that* cond $A \leq M$ *let m denote the number of distinct eigenvalues. Then we have*

$$\mu(M_k) = 1 \quad \text{for } k \geq m \quad \text{or} \quad \eta \leq 1/M$$

and

$$\mu(M_k) \geq 1 - \frac{2\Gamma(n/2)}{\sqrt{\pi}\,\Gamma((n-1)/2)} \sqrt{\frac{\eta M - 1}{1-\eta}}\, U^{-1}_{2(k-1)}\left(\sqrt{1 + (1-\eta)/(\eta M - 1)}\right)$$

$$\geq 1 - 0.824\sqrt{n} \sqrt{\frac{\eta M - 1}{1-\eta}}\, U^{-1}_{2(k-1)}\left(\sqrt{1 + (1-\eta)/(\eta M - 1)}\right)$$

for $k < m$ and $1/M < \eta < 1$.

4. Estimating the condition number λ_1/λ_n. We now apply the results of the previous section and Theorem 4.2 of [9] to estimate the condition number of a matrix. Since $\hat{\xi}_k = \hat{\xi}(A, b, k)$ and $\xi_k = \xi(A, b, k)$ are the Lanczos approximations to the largest and to the smallest eigenvalues of A correspondingly, we estimate the condition number cond A by $\kappa_k = \hat{\xi}_k/\xi_k$, $\kappa_k \leq$ cond A. Clearly, the quality of this estimation depends on the starting vector b. We now give a lower bound on the measure of the set of vectors b for which cond $A \leq \alpha\kappa_k$ for $\alpha > 1$. Let

$$Z_k = \{b \in S_n : \text{cond } A \leq \alpha\kappa_k\}.$$

Then we have the following theorem.

THEOREM 5. *Let A be a symmetric positive definite matrix A and let m denote the number of distinct eigenvalues of A. Then*

$$\mu(Z_k) = 1 \quad \text{for } k \geq m,$$

for any k and any $\alpha > 1$

$$\mu(Z_k) \geq 1 - 1.648\sqrt{n} \inf_{0 < t < \frac{1-1/\alpha}{\kappa_k}} \phi(t),$$

where

$$\phi(t) = e^{-(2k-1)\sqrt{1-t\kappa_k-1/\alpha}} + e^{-(2k-1)\sqrt{t}}.$$

Proof. The proof for $k \geq m$ follows immediately from the corresponding parts of Theorem 4.2 in [9] and Theorem 2. Assume now that k is any integer. Consider the sets

$$W = \{b \in S_n : \lambda_1 \leq \hat{\xi}_k/(1-\varepsilon_1)\}$$

and

$$\hat{Y} = \{b \in S_n : \lambda_n \geq \xi_k - \varepsilon_2 \lambda_1\}$$

for any $\varepsilon_1, \varepsilon_2 \in (0,1)$. From Theorem 4.2 of [9] it follows that

$$\mu(W) > 1 - \phi_1(\varepsilon_1) = 1 - 1.648\sqrt{n}\, e^{-\sqrt{\varepsilon_1}(2k-1)},$$

while from Theorem 2 we have

$$\mu(\hat{Y}) > 1 - \phi_2(\varepsilon_2) = 1 - 1.648\sqrt{n}\, e^{-(2k-1)\sqrt{\varepsilon_2}}.$$

Let

$$Y = \left\{ b \in S_n : \lambda_n \geq \xi_k - \varepsilon_2 \frac{\hat{\xi}_k}{1-\varepsilon_1} \right\}$$

for ε_1 and ε_2 such that $\xi_k - \varepsilon_2 \frac{\hat{\xi}_k}{1-\varepsilon_1} > 0$. Then $W \cap \hat{Y} \subset Y$ ($W \cap \hat{Y} \subset W \cap Y$) and

$$\mu(W \cap Y) \geq \mu(W \cap \hat{Y}) = \mu(W) + \mu(\hat{Y}) - \mu(W \cup \hat{Y})$$

$$> 1 - \phi_1(\varepsilon_1) + 1 - \phi_2(\varepsilon_2) - 1 = 1 - (\phi_1(\varepsilon_1) + \phi_2(\varepsilon_2)).$$

Thus, since $\xi_k - \varepsilon_2 \frac{\hat{\xi}_k}{1-\varepsilon_1} > 0$, we have

$$\mu\left\{ b \in S_n : \operatorname{cond} A \leq \frac{1}{1 - \varepsilon_1 - \varepsilon_2 \kappa_k} \kappa_k \right\} > 1 - (\phi_1(\varepsilon_1) + \phi_2(\varepsilon_2)).$$

Maximizing the right-hand side of the last inequality under constraints

$$0 < \varepsilon_1 < 1, \quad 0 < \varepsilon_2 < 1, \quad 0 < \varepsilon_2/(1-\varepsilon_1) < 1/\kappa_k, \quad 1 - \varepsilon_1 - \varepsilon_2\kappa_k = 1/\alpha,$$

we complete the proof by taking $\varepsilon_2 = t$ and $\varepsilon_1 = 1 - 1/\alpha - t\kappa_k$. □

The first part of Theorem 5 states that the Lanczos algorithm at the mth step recovers the condition number of a matrix with probability one. This confirms our intuition that for sufficiently large k the Lanczos algorithm may fail only on a set of measure zero.

The second part of Theorem 5 gives a lower bound on the probability that the condition number of a matrix is not greater than $\alpha\kappa_k$ for $\alpha > 1$. It may happen that this bound is negative, especially when κ_k grows too rapidly with respect to k (see Table 1 where the approximated minimum of the function ϕ is shown) or for α close to one or n huge. However, for $k \geq n$, $\kappa_k = \operatorname{cond} A$ with probability one no matter what was happening with κ_k for $k < n$.

It is difficult to calculate the exact value of the infimum of the function ϕ in the interval $(0, (1 - 1/\alpha)/\kappa_k)$ (note that k, α, and κ_k are known), but it is quite easy to find a satisfactory approximation of it numerically. Indeed, having calculated κ_k we can perform a few steps of the bisection method applied to the first derivative ϕ' on the interval $(0, (1 - 1/\alpha)/\kappa_k)$. Table 1 displays an approximation of the minimum of the function ϕ for $\alpha = 10$ for some values of k and κ_k. The first column contains values of k. The next four columns show an approximation of the minimum of $\phi, \phi_{\min}(k, \kappa_k)$ for $\kappa_k = \sqrt{k}$, $\kappa_k = k$, $\kappa_k = k^2$, and $\kappa_k = k^3$, respectively.

TABLE 1

k	$\phi_{\min}(k, \sqrt{k})$	$\phi_{\min}(k, k)$	$\phi_{\min}(k, k^2)$	$\phi_{\min}(k, k^3)$
10	0.00025	0.00612	0.184	0.585
20	0.0000002	0.00039	0.163	0.667
30	0	0.00005	0.158	0.714
40	0	0.000009	0.156	0.745
50	0	0.000002	0.154	0.768
60	0	0.0000006	0.153	0.785
70	0	0.0000002	0.153	0.799
80	0	0.0000001	0.152	0.810

To apply Theorem 5 for $\mu(Z_k)$ we have to compute κ_k. It is possible to estimate $\mu(Z_k)$ without computing κ_k. It can be done by combining Theorems 3 and 4. Indeed, observe that for $\alpha = \theta/\eta$, we have $L_k \cap M_k \subseteq Z_k$, and

$$\mu(Z_k) \geq \mu(L_k \cap M_k) = \mu(L_k) + \mu(M_k) - \mu(L_k \cup M_k) \geq \mu(L_k) + \mu(M_k) - 1.$$

Thus, to find a lower bound on $\mu(Z_k)$, we need to find lower bounds on $\mu(L_k)$ and $\mu(M_k)$. Applying Theorems 3 and 4 we get the following theorem.

THEOREM 6. *For any symmetric positive definite matrix A let m denote the number of distinct eigenvalues. Let α be any number, $\alpha > 1$. Then for any numbers θ and η such that $\theta > 1$, $0 < \eta < 1$ and $\alpha = \theta/\eta$, we have*

$$\mu(Z_k) = 1 \quad \text{for } k \geq m \quad \text{or} \quad (\theta \geq \operatorname{cond} A \quad \text{and} \quad \eta \leq 1/\operatorname{cond} A)$$

and

$$\mu(Z_k) \geq 1 - \frac{2\Gamma((n-j)/2+1)}{\sqrt{\pi}\,\Gamma((n-j+1)/2)\,\sqrt{\theta-1}\,U_{2(k-1)}(\sqrt{\theta})} - \frac{2\Gamma(i/2)}{\sqrt{\pi}\,\Gamma((i-1)/2)} \sqrt{\frac{\eta \operatorname{cond} A - 1}{1-\eta}}\, U^{-1}_{2(k-1)}\left(\sqrt{1+(1-\eta)/(\eta \operatorname{cond} A - 1)}\right),$$

where $\theta < \operatorname{cond} A$ or $\eta > 1/\operatorname{cond} A$ and j and i satisfy:

$$\lambda_j \leq \lambda_1/\theta < \lambda_{j-1}, \quad 2 \leq j \leq n \quad \text{and} \quad \lambda_n/\lambda_{i-1} < \eta \leq \lambda_n/\lambda_i, \quad 2 \leq i \leq n.$$

Theorem 6 yields a bound on the measure of the set Z_k that depends on the indices i and j as well as $\operatorname{cond} A$. Usually they are unknown. However, the indices i, j can be replaced by n, and $\operatorname{cond} A$ can be replaced by an upper bound M. Using the second inequality of Theorem 3 and Corollary 1, we then get the following theorem.

THEOREM 7. *Let A be any symmetric positive definite matrix of the size $n \geq 8$ such that $\operatorname{cond} A \leq M$. Then under assumptions of Theorem 6, we have*

$$\mu(Z_k) = 1 \quad \text{for } k \geq m \quad \text{or} \quad (\theta \geq M \quad \text{and} \quad \eta \leq 1/M)$$

and

$$\begin{aligned}\mu(Z_k) &\geq 1 - \frac{2\Gamma(n/2)}{\sqrt{\pi}\,\Gamma((n-1)/2)} \left((\theta-1)^{-1/2} U^{-1}_{2(k-1)}(\sqrt{\theta}) + \sqrt{\frac{M\eta-1}{1-\eta}}\, U^{-1}_{2(k-1)}\left(\sqrt{1+(1-\eta)/(M\eta-1)}\right)\right) \\ &\geq 1 - 0.824\sqrt{n}\left((\theta-1)^{-1/2} U^{-1}_{2(k-1)}(\sqrt{\theta}) + \sqrt{z}\, U^{-1}_{2(k-1)}(\sqrt{1+1/z}\,)\right)\end{aligned}$$

for $1 < \theta < M$ *or* $1/M < \eta < 1$, *where* $z = (M\eta - 1)/(1 - \eta)$.

Note that the better bound M on $\operatorname{cond} A$ is given, and the better bound on the measure of the set Z_k is obtained.

Obviously, any number $\alpha > 1$ can be decomposed as $\alpha = \theta/\eta$, $M > \theta > 1$, $1/M < \eta < 1$ in many ways. To get the best probability bound one can maximize it over θ and η under the constrains $\alpha = \theta/\eta$, $M > \theta > 1$, $1/M < \eta < 1$. It can easily be done numerically.

5. Numerical results. We tested several matrices with many pseudorandom starting vectors b. Without loss of generality [9] we restricted ourselves only to diagonal matrices. Vectors b were generated in the same way as in [9]. The tests were performed on XT and 486 personal computers with the round-off unit of order 10^{-7}. All calculations were done in the single precision. The diagonal form of matrices reduced the effect of round-off errors.

The purpose of the numerical tests was, in particular, to verify the sharpness of the bound of Theorem 1 and to check the quality of the bounds of Theorems 5 and 7. We first report our results for the matrix of dimension $n = 250$ with eigenvalues

$$\lambda_k = 1 + \cos\frac{(2k-1)\pi}{2n}, \qquad k = 1, \ldots, 250.$$

Hence, λ_k's are the shifted zeros of the Chebyshev polynomial of the first kind of degree 250. This distribution of eigenvalues is difficult for the Lanczos algorithm [15]. We have selected several values of ε and ten pseudorandom vectors b and for each of them we run the Lanczos algorithm for $k = 1, \ldots, k^*$, where k^* was the minimal k for which the relative error $e(A, b, k)$ was no greater than ε. We compared the relative error with k^{-2}. For each b and k we observed

$$0.20 \le e(A, b, k)\, k^2 \le 1.23.$$

In Table 2 we report the average errors achieved after $(k-1)$ steps of the Lanczos algorithm for some values of k displayed in the first column. The second column shows the average errors defined as

$$\varepsilon^{\text{ave}} = \frac{1}{10}\sum_{i=1}^{10} e(A, b_i, k),$$

where b_i is the ith pseudorandom vector. The third column contains upper bounds on the Lanczos errors from Theorem 1, i.e.,

$$\varepsilon^{up} = 0.103\left(\frac{\ln(n(k-1)^4)}{k-1}\right)^2.$$

The fourth column shows the ratio $r_1 = \varepsilon^{\text{up}}/\varepsilon^{\text{ave}}$, while the last column displays how r_1 is related to the (probably unnecessary) factor $r_2 = r_1/\ln^2(n(k-1)^4)$ in the theoretical bound.

TABLE 2

$k-1$	ε^{ave}	ε^{up}	r_1	r_2
10	0.005276	0.2235	42.36	0.195
20	0.001379	0.0789	57.22	0.187
30	0.000688	0.0419	60.87	0.166
40	0.000387	0.0265	68.53	0.167
50	0.000244	0.0185	75.76	0.184
60	0.000165	0.0137	83.23	0.174
70	0.000145	0.0107	73.95	0.146

Small changes in the last column of Table 2 may suggest that the error of the Lanczos algorithm for the matrix with Chebyshevian distribution of eigenvalues behaves like k^{-2} and the factor roughly $\ln^2(n(k-1)^4)$ is probably an overestimate in the upper bound.

Table 3 shows for five values of ε how many steps were needed to achieve the relative error no greater than ε. The first row contains the values of ε, the second row the average number k^{ave} of performed steps with $k^{\text{ave}} = \sum_{i=1}^{10} k(A, b_i)/10$, where $k(A, b_i)$ was the number of steps needed for the pseudorandom vector b_i. The third row indicates the minimal $k = k^{\text{up}}$ such that

$$0.103 \left(\frac{\ln(n(k-1)^4)}{k-1} \right)^2 \leq \varepsilon,$$

which is one of the two theoretical bounds of Theorem 1 for the Lanczos algorithm. The fourth row presents the ratio $r = k^{\text{up}}/k^{\text{ave}}$.

TABLE 3

ε	$1_{10}-3$	$7.5_{10}-4$	$5_{10}-4$	$2.5_{10}-4$	$2.0_{10}-4$
k^{ave}	35.5	41	49.9	65.5	69.2
k^{up}	287	339	429	638	724
r	8.08	8.27	8.6	9.74	10.46

Note that the theoretical bound exceeds the actual value by a factor of at most 11. Observe also that all k^{up}'s are greater than $n = 250$ and the second bound of Theorem 1 gives a better estimate.

We also tested matrices of dimension 250 with other distributions of eigenvalues in the interval $(0, 2)$. The following distributions were tested:

- the quadratic, $\lambda_i = 2\,(1 - i/251)^2$ and $\lambda_i = 2\,(1 - (i/251)^2)$;
- the uniform, $\lambda_i = 2\,(1 - i/251)$;
- the logarithmic, $\lambda_i = 2\,\log(252 - i)/\log 252$;
- the expotential, $\lambda_i = 2\,e^{-\sqrt[3]{i}}, \lambda_i = 1 + e^{-i}, i = 1, \ldots, 250$.

For all these matrices we have observed faster convergence to the smallest eigenvalue than for the matrix with Chebyshevian distribution. For some matrices the difference was significant.

We were also interested in checking the quality of the bound on the condition number presented in Theorems 5 and 7. As before, for each matrix we selected several pseudorandom vectors b and ran the Lanczos algorithm both for the smallest and for the largest eigenvalues with $k = 1, 2, \ldots, k^*$, where k^* was the minimal k for which this algorithm gave approximations with a prescribed accuracy. In all tests, i.e., for all matrices, all starting vectors and all k, the number α was fixed to be equal to 10. We now describe some tests on the bound of Theorem 5. The infimum of the function ϕ from Theorem 5 was computed numerically. The matrices most intensively tested were the matrices that arise as discretizations of a two-dimensional Laplace operator. We first report some results for the matrix A of the size 256 (16 points in each direction). The condition number of this matrix is $116.461\ldots$. Table 4 shows some results obtained for ten pseudorandom vectors $b_i, i = 1, 2, \ldots, 10$. The first column contains twelve values of k. The next one indicates the average of the condition numbers obtained for vectors b_i, i.e.,

$$\text{cond}^{\text{ave}} = \frac{1}{10} \sum_{i=1}^{10} \hat{\xi}(A, b_i, k)/\xi(A, b_i, k),$$

while the third one shows the standard deviation $\sigma_{10}^{\text{cond}}$ of the empirical condition number. The last two columns concern the bound on the probability given in Theorem 5, namely, the average, prob^{ave}, and the standard deviation, $\sigma_{10}^{\text{prob}}$, of the bound of probability for the vectors b_i.

TABLE 4

$k-1$	cond^{ave}	$\sigma_{10}^{\text{cond}}$	prob^{ave}	$\sigma_{10}^{\text{prob}}$
5	21.533	5.997	−2.692	0.913
10	57.481	19.465	−1.178	0.885
15	85.442	21.391	−0.230	0.448
20	102.793	19.770	0.349	0.199
25	111.690	12.074	0.701	0.062
30	116.163	0.782	0.866	0.002
35	116.451	0.018	0.944	0
40	116.457	0.0017	0.977	0
45	116.457	0.0017	0.990	0
50	116.457	0.0017	0.996	0
55	116.457	0.0017	0.998	0
60	116.457	0.0017	0.999	0

Observe that the first three elements of the fourth and fifth columns should be replaced by zeros, but we think it is interesting to see how the bound of Theorem 5 increases with k.

Table 4 shows that the average bound on the probability is close to 1 for k larger than 30. Note that for these k's, the average condition number is nearly equal to the exact condition number of this matrix. We stress that the corresponding standard deviations are very small. For k smaller than 20, the average bound of the probability was quite small (or even negative) although for $k \geq 7$ and all pseudorandom vectors b_i, the computed condition number, $\hat{\xi}_k/\xi_k$ satisfied: $\hat{\xi}_k/\xi_k \leq \text{cond}\, A \leq 10\,\hat{\xi}_k/\xi_k$.

For the matrix A of the size 1024 (32 points in each direction) the corresponding table is (the condition number of this matrix is 440.6886...).

TABLE 5

$k-1$	cond^{ave}	$\sigma_{10}^{\text{cond}}$	prob^{ave}	$\sigma_{10}^{\text{prob}}$
5	24.195	1.625	−7.338	0.496
10	66.358	8.380	−4.223	0.743
15	123.881	30.943	−3.053	1.256
20	181.720	56.008	−2.115	1.312
25	233.467	71.980	−1.347	1.093
30	279.179	80.741	−0.752	0.834
35	321.968	82.043	−0.314	0.588
40	375.354	62.747	−0.049	0.316
45	420.228	32.238	0.191	0.120
50	434.603	11.736	0.452	0.033
55	439.324	2.548	0.643	0.005
60	440.310	0.684	0.771	0.001
65	440.556	0.156	0.854	0
70	440.622	0.042	0.907	0

Note that for $k > 8$ and all pseudorandom vectors b_i, the computed condition number, $\hat{\xi}_k/\xi_k$ satisfied: $\hat{\xi}_k/\xi_k \leq \text{cond}\, A \leq 10\,\hat{\xi}_k/\xi_k$, while the corresponding probability is positive for $k > 42$. This suggests that the lower bound on the probability given in Theorem 5 is rather poor for this matix.

We have also tested matrices with distributions of eigenvalues described earlier. For instance, for Chebyshevian distribution the bound on the probability was negative (usually between -7 and -3). Note that for this matrix the approximations $\hat{\xi}_k/\xi_k$ of the condition number satisfying: $\hat{\xi}_k/\xi_k \leq \text{cond}\, A = 101371 \leq 10\,\hat{\xi}_k/\xi_k$ were observed

(on average) for k larger than 70. Let us stress that for all pseudorandom vectors b_i, the numbers k^* were smaller than 80. Similar results were observed for the quadratic distribution of eigenvalues $\lambda_i = 2(1 - i/251)^2$, $i = 1, \ldots, 250$ (for this matrix the condition number is equal to 62500).

For other distributions we obtained better results. Note that the condition numbers were much smaller than the condition number for the Chebyshevian distribution. For instance, for the logarithmic distribution the bound on the probability was on average greater than 0.9 for k larger than 9. The condition number of this matrix is 7.9715 and for each k we got the empirical condition number $\hat{\xi}_k/\xi_k$ satisfying $\hat{\xi}_k/\xi_k \leq \operatorname{cond} A = 7.9715 \leq 10\,\hat{\xi}_k/\xi_k$.

We now turn to the bounds for the condition number of a matrix given by Theorem 7. As before, the number α was fixed to 10 and θ and η were chosen to maximize (numerically) the second bound of Theorem 7. Table 6 shows some results for the matrix A of the size 256 that arises as a discretization of a two-dimensional Laplace operator and for ten pseudorandom vectors $b_i, i = 1, \ldots, 10$ (compare with Table 4). The first three columns contain values defined in the description of Table 4. The last two columns display the bound on the probability given in Theorem 7 with $M = \operatorname{cond} A$ (prob(1)) and $M = 100 \operatorname{cond} A$ (prob(100)), respectively. To speed our tests, for $k-1 \geq 75$ we did not compute approximations to the extreme eigenvalues. Instead, we use bounds on $\mu(Z_k)$ from Theorem 7 with κ_{69}. Thus, we slightly decreased the actual bound on $\mu(Z_k)$ and this is denoted by "$\geq$" in the second column and by "$\leq$" in the third column.

TABLE 6

$k-1$	cond^{ave}	$\sigma_{10}^{\text{cond}}$	prob(1)	prob(100)
5	21.307	6.444	−1.401	−45.042
10	55.979	19.397	0.800	−21.575
15	84.451	20.874	0.984	−13.191
20	103.419	19.902	0.9988	−8.792
25	111.676	12.069	0.99991	−6.057
30	116.162	0.781	0.99999	−4.202
35	116.447	0.020	1	−2.881
40	116.456	0.013	1	−1.915
45	116.456	0.013	1	−1.198
50	116.456	0.013	1	−0.661
55	116.456	0.013	1	−0.256
60	116.456	0.013	1	−0.050
65	116.456	0.013	1	0.2390
70	116.456	0.013	1	0.4236
75	$\geq$ 116.456	$\leq$ 0.013	1	0.5869
80	$\geq$ 116.456	$\leq$ 0.013	1	0.6871
85	$\geq$ 116.456	$\leq$ 0.013	1	0.7627
90	$\geq$ 116.456	$\leq$ 0.013	1	0.8202
95	$\geq$ 116.456	$\leq$ 0.013	1	0.8636
100	$\geq$ 116.456	$\leq$ 0.013	1	0.8968
110	$\geq$ 116.456	$\leq$ 0.013	1	0.9407
120	$\geq$ 116.456	$\leq$ 0.013	1	0.9660
130	$\geq$ 116.456	$\leq$ 0.013	1	0.9804
140	$\geq$ 116.456	$\leq$ 0.013	1	0.9888
150	$\geq$ 116.456	$\leq$ 0.013	1	0.9935

We observed that the probability bound of Theorem 7 with $M = \operatorname{cond} A$ is larger than 0.9 for $k \geq 12$. The fact that relatively early we can be almost sure that the Lanczos algorithm produces a good approximation to the condition number of this matrix is due to the fact that we have used a perfect bound for the condition number, namely, the condition number itself. To contrast this, we see that the probability

bound gets much worse (the fifth column) if we use Theorem 7 with $M = 100 \operatorname{cond} A$. Let us stress that for all $k \geq 7$, the approximations $\kappa_k = \hat{\xi}_k/\xi_k$ to the condition number satisfied

$$\kappa_k \leq \operatorname{cond} A \leq 10\kappa_k.$$

For the matrix A of the size 1024 (32 points in each direction) the corresponding table (Table 7) is as follows (compare with Table 5).

TABLE 7

$k-1$	$\operatorname{cond}^{\text{ave}}$	$\sigma_{10}^{\text{cond}}$	prob(1)	prob(100)
10	66.358	8.380	−2.695	−88.952
20	181.720	56.008	0.758	−42.527
30	279.179	80.741	0.984	−26.218
40	375.354	62.747	0.999	−17.680
50	434.603	11.736	1	−12.381
60	440.310	0.684	1	−8.801
70	440.622	0.042	1	−6.256
80	440.641	0.023	1	−4.410
90	440.641	0.023	1	−3.047
100	≥ 440.641	≤ 0.023	1	−2.035
110	≥ 440.641	≤ 0.023	1	−1.278
120	≥ 440.641	≤ 0.023	1	−0.710
130	≥ 440.641	≤ 0.023	1	−0.283
140	≥ 440.641	≤ 0.023	1	0.035
150	≥ 440.641	≤ 0.023	1	0.276
160	≥ 440.641	≤ 0.023	1	0.455
170	≥ 440.641	≤ 0.023	1	0.591
180	≥ 440.641	≤ 0.023	1	0.693
190	≥ 440.641	≤ 0.023	1	0.769
200	≥ 440.641	≤ 0.023	1	0.827
210	≥ 440.641	≤ 0.023	1	0.870
220	≥ 440.641	≤ 0.023	1	0.902
230	≥ 440.641	≤ 0.023	1	0.926

We now briefly report some results for other matrices. The size of the matrices was chosen to be equal to 250. For the quadratic distribution of eigenvalues $\lambda_i = 2(1-i/251)^2$, $i = 1, \ldots, 250$ in the interval $(0, 2)$ we obtained much worse probability bounds. For $M = \operatorname{cond} A$, the bounds were negative up to the 140th step and they reached 0.786 at the 201st step. For $M = 100 \operatorname{cond} A$, the bounds were all negative and at the 201st step they reached −25.02. Note that for all $k > 71$, the approximated condition numbers κ_k satisfy

$$\kappa_k \leq \operatorname{cond} A = 62500 \leq 10\kappa_k.$$

Slightly worse results were obtained for the shifted Chebyshev distribution of the eigenvalues. Better results were obtained for matrices with the quadratic distribution of eigenvalues

$$\lambda_i = 2(1 - (i/251)^2), \qquad i = 1, ..., 250.$$

The condition number of this matrix is equal to 125.749. Only five steps were needed to get the desired approximations to the condition number of this matrix. The probability bounds for $M = \operatorname{cond} A$ and $M = 100 \operatorname{cond} A$ were positive after 9 and 62 steps. They were greater than 0.5 after 11 and 76 steps, and greater than 0.9 after 15 and 106 steps, respectively.

6. Final remarks. *Remark* 1. We now report a result for the average relative error with respect to the smallest eigenvalue. That is, define

$$e^{\text{avg}}_{\min}(A,k) = \int_{S_n} \frac{\xi(A,b,k) - \lambda_n}{\lambda_n}\, \mu(db).$$

We will show that this error is unbounded in the class of all symmetric positive definite matrices. Indeed, using the same notation as in the proof of Theorem 3, we get

$$e^{\text{avg}}_{\min}(A,k) = \frac{1}{c_n\,\lambda_n} \int_{B_n} \inf_{P\in\mathcal{P}_k} \frac{\sum_{i=1}^{n-1}(\lambda_i - \lambda_n) b_i^2 P^2(\lambda_i)}{\sum_{i=1}^{n} b_i^2 P^2(\lambda_i)}\, db.$$

Assume that the matrix A has n distinct eigenvalues and let $\lambda_i - \lambda_n \geq \delta > 0$ for $i = 1,\ldots,n-1$. Then we have

$$e^{\text{avg}}_{\min}(A,k) \geq \frac{\delta}{c_n\,\lambda_n} \int_{B_n} \left(1 - \sup_{P\in\mathcal{P}_k} \frac{b_n^2 P^2(\lambda_n)}{\sum_{i=1}^{n} b_i^2 P^2(\lambda_i)}\right) db = \frac{\delta}{c_n\,\lambda_n}\, \phi(A,k),$$

where

$$\phi(A,k) = \int_{B_n} \left(1 - \sup_{P\in\mathcal{P}_k} \frac{b_n^2 P^2(\lambda_n)}{\sum_{i=1}^{n} b_i^2 P^2(\lambda_i)}\right) db.$$

Note that the function ϕ is shift invariant, i.e., $\phi(A+\alpha I,k) = \phi(A,k)$ for all $\alpha \in \mathcal{R}$, and $\phi(A,k) > 0$ for $k \leq n-1$. Take now the matrix $A - \alpha I$, where $\alpha < \lambda_n$. Then for $\alpha \to \lambda_n$, we have

$$e^{\text{avg}}_{\min}(A-\alpha I,k) \geq \frac{\delta}{c_n(\lambda_n - \alpha)}\, \phi(A,k) \to +\infty.$$

Thus for any k no greater than $n-1$ and any positive number M there exists a symmetric positive definite matrix A such that

$$e^{\text{avg}}_{\min}(A,k) > M.$$

Remark 2. We now consider a gap ratio as the error criterion instead of the relative error with respect to the largest eigenvalue, i.e., we want to find ξ such that

$$\frac{\xi - \lambda_n(A)}{\lambda_1(A) - \lambda_n(A)} < \varepsilon.$$

Note that the gap ratio for the Lanczos algorithm approximating the smallest eigenvalue is shift invariant, i.e.,

$$\frac{\xi(A+\alpha I,b,k) - \lambda_n(A+\alpha I)}{\lambda_1(A+\alpha I) - \lambda_n(A+\alpha I)} = \frac{\xi(A,b,k) - \lambda_n(A)}{\lambda_1(A) - \lambda_n(A)}.$$

Using a continuity argument, we can conclude that the bounds given in Theorems 1 and 2 hold, since

$$\frac{\xi(A) - \lambda_n(A)}{\lambda_1(A) - \lambda_n(A)} = \frac{\xi(B) - \lambda_n(B)}{\lambda_1(B) - \lambda_n(B)},$$

where $B = A - \lambda_n I$, $B = B^T \geq 0$ with $\lambda_n(B) = 0$.

Acknowledgments. We thank A. Kiełbasiński, J. F. Traub, and A. G. Werschulz for their valuable comments.

REFERENCES

[1] C. H. BISCHOF, *Incremental condition estimation*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 644–659.

[2] J. K. CULLUM AND R. A. WILLOUGHBY, *Lanczos algorithms for large symmetric eigenvalue computation*, Progress in Scientific Computing, 3,4, Birkhauser, Boston, 1985.

[3] J. H. DIXON, *Estimating extremal eigenvalues and condition numbers of matrices*, SIAM J. Numer. Anal., 20 (1983), pp. 812–814.

[4] W. R. FERNG, G. H. GOLUB, AND R. J. PLEMMONS, *Adaptive Lanczos methods for recursive condition estimation*, Numer. Algorithms, 1 (1991), pp. 1–19.

[5] G. H. GOLUB AND C. VAN LOAN, *Matrix Computations*, Johns Hopkins Press, Baltimore, MD, 1989.

[6] I. S. GRADSHTEYN AND I. W. RYZHIK, *Table of Integrals, Series and Products*, Academic Press, New York, 1980

[7] W. KAHAN AND B. N. PARLETT, *How far should we go with the Lanczos process?*, in Sparse Matrix Computations, J. Bunch and D.Rose, eds., Academic Press, New York, 1976, pp. 131–144.

[8] S. KANIEL, *Estimates for some computational techniques in linear algebra*, Math. Comp., 20 (1966), pp. 369–378.

[9] J. KUCZYŃSKI AND H. WOŹNIAKOWSKI, *Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1094–1122.

[10] C. C. PAIGE, *The computation of eigenvalues and eigenvectors of very large sparse matrix*, Ph. D. thesis, University of London, London, 1971.

[11] ———, *Computational variants of the Lanczos method for the eigenproblem*, J. Inst. Math. Appl., 10 (1972), pp. 373–381.

[12] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[13] S. PASZKOWSKI, *Zastosowania numeryczne wielomianów i szeregów Czebyszewa*, Państwowe Wydawnictwo Naukowe, Warsaw, Poland, 1975.

[14] Y. SAAD, *On the rates of convergence of the Lanczos and the block Lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[15] D. S. SCOTT, *Analysis of the Symmetric Lanczos Process*, Ph. D. thesis, Memorandum NCB/ERLM 78/40 University of California at Berkeley, Berkeley, 1978.

[16] J. H. WILKINSON, *The algebraic eigenvalue problem*, Oxford University Press, London, New York, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 2, pp. 692–713, April 1994

021

AN ERROR MODEL FOR SWARZTRAUBER'S PARALLEL TRIDIAGONAL EQUATION SOLVER*

NAI-KUAN TSAO†

Abstract. Error models for two algorithms based on Cramer's rule are given for tridiagonal systems. The results show that if the exact x_i of the solution vector x is expressed as det p/ det q where p and q are matrices formed from the original coefficient matrix and the right-hand side vector, then the computed x_i can be expressed as det $\hat{p}$/ det $\hat{q}$, where det $\hat{p}$ and det $\hat{q}$ are perturbed det p and det q, respectively, using either the parallel or the sequential algorithm. The relative error of each product in the determinantal expansion of det $\hat{p}$ and det $\hat{q}$ can also be bounded easily.

Key words. tridiagonal matrix, determinant, parallel algorithms, linear equations, error analysis

AMS subject classifications. 65F05, 68A20, 15A15

1. Introduction.

In [1], Swarztrauber described a parallel algorithm that is based on an efficient implementation of Cramer's rule for solving general tridiagonal equations. His basic algorithm can be described as follows. For a given tridiagonal system

$$Ax = y, \quad A = \begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & c_{n-1} \\ & & & & a_n & b_n \end{bmatrix} \equiv [\, a_j \quad b_j \quad c_j \,], \quad a_1 = c_n = 0,$$

$$x = [\, x_1 \quad x_2 \quad \dots \quad x_n \,]^T, \qquad y = [\, y_1 \quad y_2 \quad \dots \quad y_n \,]^T,$$

let us denote by d_i the determinant of the leading i by i submatrix of A, f_i the determinant of the trailing $(n-i+1)$ by $(n-i+1)$ submatrix of A, t_i the determinant of the leading i by i submatrix of A with column i replaced by the first i components of y, and u_i the determinant of the trailing $(n-i+1)$ by $(n-i+1)$ submatrix of A with column i replaced by the last $(n-i+1)$ components of y. Then the following theorem can be proved [1].

THEOREM 1.1. *Given the sequences d_i, f_i, t_i, and u_i defined above, then the solution x_i is given by*

$$x_1 = u_1/d_n, \quad x_i = (d_{i-1}u_i - a_i f_{i+1} t_{i-1})/d_n, \quad 1 < i < n, \quad x_n = t_n/d_n,$$

where

$$d_i = b_i d_{i-1} - a_i c_{i-1} d_{i-2}, \quad d_0 = 1, \; d_1 = b_1, \quad i > 1,$$

$$t_i = -a_i t_{i-1} + d_{i-1} y_i, \quad t_1 = y_1, \quad i > 1,$$

$$f_i = b_i f_{i+1} - a_{i+1} c_i f_{i+2}, \quad f_{n+1} = 1, \quad f_n = b_n, \quad i < n,$$

*Received by the editors April 15, 1992; accepted for publication (in revised form) November 10, 1992.

†Department of Computer Science, Wayne State University, Detroit, Michigan 48202 (tsao@pandora.cs.wayne.edu).

$$u_i = -c_i u_{i+1} + f_{i+1} y_i, \quad u_n = y_n, \quad i < n.$$

A sequential algorithm based on the above theorem is given below.

ALGORITHM Sw1
$d_1 = b_1$; $e_1 = c_1$
for $i = 2$ to n do
  $d_i = fl(b_i d_{i-1} - a_i e_{i-1})$; $e_i = fl(c_i d_{i-1})$
$f_n = b_n$; $g_n = a_n$
for $i = n - 1$ downto 1 do
  $f_i = fl(b_i f_{i+1} - c_i g_{i+1})$; $g_i = fl(a_i f_{i+1})$
$t_1 = y_1$
for $i = 2$ to n do
  $t_i = fl(-a_i t_{i-1} + d_{i-1} y_i)$
$u_n = y_n$
for $i = n - 1$ downto 1 do
  $u_i = fl(-c_i u_{i+1} + f_{i+1} y_i)$
$x_1 = fl(u_1/d_n)$; $x_n = fl(t_n/d_n)$
for $i = 2$ to $n - 1$ do
  $x_i = fl((-g_i t_{i-1} + d_{i-1} u_i)/d_n)$

where $fl(.)$ is used to denote the computed result of the enclosed argument.

For $n = 2^k$, $k > 0$, Stone's recursive doubling technique [2] can be used to parallelize the calculation of the vectors d, e, f, g, t, and u using the following algorithm.

ALGORITHM Sw2
for $i = 1$ to n do {in parallel}
  $\alpha_i^{(i)} = -a_i$; $\gamma_i^{(i)} = -c_i$; $Q_i^{(i)} = \begin{bmatrix} b_i & c_i \\ -a_i & 0 \end{bmatrix}$
for $j = 1$ to k do
  $l = 2^{j-1}$
  for $i = 0$ to $n - 2^j$ step 2^j do {in parallel}
    $\alpha_{i+1}^{(i+2l)} = fl\left(\alpha_{i+1}^{(i+l)} \alpha_{i+l+1}^{(i+2l)}\right)$; $\gamma_{i+1}^{(i+2l)} = fl\left(\gamma_{i+1}^{(i+l)} \gamma_{i+l+1}^{(i+2l)}\right)$
    $Q_{i+1}^{(i+2l)} = fl\left(Q_{i+1}^{(i+l)} Q_{i+l+1}^{(i+2l)}\right)$
for $j = 0$ to k do {in parallel}
  $l = 2^j$; $[d_l \; e_l] = [1 \; 0] Q_1^{(l)}$; $\begin{bmatrix} f_{n-l+1} \\ g_{n-l+1} \end{bmatrix} = Q_{n-l+1}^{(n)} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
for $j = k - 1$ downto 1 do
  $l = 2^{j-1}$
  for $i = 2^j$ to $n - 2^j$ step 2^j do {in parallel}
    $[d_{i+l} \; e_{i+l}] = fl\left([d_i \; e_i] Q_{i+1}^{(i+l)}\right)$; $\begin{bmatrix} f_{i-l+1} \\ g_{i-l+1} \end{bmatrix} = fl\left(Q_{i-l+1}^{(i)} \begin{bmatrix} f_{i+1} \\ g_{i+1} \end{bmatrix}\right)$
$d_0 = 1$; $f_{n+1} = 1$
for $i = 1$ to n do {in parallel}
  $t_i^{(i)} = fl(d_{i-1} y_i)$; $u_i^{(i)} = fl(f_{i+1} y_i)$
for $j = 1$ to k do
  $l = 2^{j-1}$
  for $i = 0$ to $n - 2^j$ step 2^j do {in parallel}
    $t_{i+1}^{(i+2l)} = fl\left(t_{i+l+1}^{(i+2l)} + \alpha_{i+l+1}^{(i+2l)} t_{i+1}^{(i+l)}\right)$; $u_{i+1}^{(i+2l)} = fl\left(u_{i+1}^{(i+l)} + \gamma_{i+1}^{(i+l)} u_{i+l+1}^{(i+2l)}\right)$
for $j = k - 1$ downto 1 do

$l = 2^{j-1}$
for $i = 2^j$ to $n - 2^j$ step 2^j do {in parallel}
$t_1^{(i+l)} = fl\left(t_{i+1}^{(i+l)} + \alpha_{i+1}^{(i+l)} t_1^{(i)}\right)$; $u_{i-l+1}^{(n)} = fl\left(u_{i-l+1}^{(i)} + \gamma_{i-l+1}^{(i)} u_{i+1}^{(n)}\right)$
$x_1 = fl\left(u_1^{(n)}/d_n\right)$; $x_n = fl\left(t_1^{(n)}/d_n\right)$
for $i = 2$ to $n - 1$ do {in parallel}
$x_i = fl\left(\left(g_i t_1^{(i-1)} + d_{i-1} u_i^{(n)}\right)/d_n\right)$

Note the required t_i and u_i in Algorithm Sw1 are given as $t_1^{(i)}$ and $u_i^{(n)}$, respectively, in Algorithm Sw2.

The experimental numerical equivalence of Algorithm Sw2 and the usual Gaussian elimination method has been demonstrated in [1]. In this paper we carry out the error analysis of the above two algorithms and show that if the exact x_i of the solution vector x is expressed as $\det p/\det q$ where p and q are matrices formed from the original coefficient matrix and the right-hand side vector, then the computed x_i can be expressed as $\det \hat{p}/\det \hat{q}$, where $\det \hat{p}$ and $\det \hat{q}$ are perturbed $\det p$ and $\det q$, respectively, using either the parallel or the sequential algorithm. The relative error of each product in the determinantal expansion of $\det \hat{p}$ and $\det \hat{q}$ can also be bounded easily. Some preliminary results are given in § 2. The error analysis of Algorithms Sw1 and Sw2 are presented in §§ 3 and 4.

2. Some preliminary results. For convenience, determinants will be used to express the various computed results from Algorithms Sw1 and Sw2. To this end, let

$$D_{\delta_1,\delta_2,\ldots,\delta_k}^{\beta_1,\beta_2,\ldots,\beta_k}$$

be used to denote the determinant of a submatrix formed by the common elements of rows $\beta_1, \beta_2, \ldots, \beta_k$ and columns $\delta_1, \delta_2, \ldots, \delta_k$ of the matrix A. For contiguous rows or columns, they are denoted by

$$\beta_i : \beta_j = \beta_i, \beta_i + 1, \ldots, \beta_j - 1, \beta_j$$

or

$$\delta_i : \delta_j = \delta_i, \delta_i + 1, \ldots, \delta_j - 1, \delta_j, \qquad i \le j.$$

If row indices are the same as the column indices, then only the column indices will be used. Thus $D_{1:2,4}$ is the same as $D_{1:2,4}^{1:2,4}$. We shall also use $D_{i:j}(k)$ for $i \le k \le j$ to denote $D_{i:j}^{i:j}$ with column k, $i \le k \le j$, replaced by a column formed by $y_i, y_{i+1}, \ldots, y_j$. As an example, let $n = 4$. Then we have

$$D_{1,2} = \begin{vmatrix} b_1 & c_1 \\ a_2 & b_2 \end{vmatrix}, \quad D_{2,5}^{3,4} = \begin{vmatrix} a_3 & 0 \\ 0 & c_4 \end{vmatrix}, \quad D_{2:4}(3) = \begin{vmatrix} b_2 & y_2 & \\ a_3 & y_3 & c_3 \\ & y_4 & b_4 \end{vmatrix}.$$

We have the following lemmas.

LEMMA 2.1. *If exact computations were possible, then*

$$D_{i:j} = D_{i:k} D_{k+1:j} - D_{i:k-1,k+1}^{i:k} D_{k,k+2:j}^{k+1:j}, \quad i < j, \quad i \le k, \quad k \le j.$$

Furthermore, for $1 < i < j < n$,

$$D_{i:j-1,j+1}^{i:j} = c_j D_{i:j-1}, \qquad D_{i-1,i+1:j}^{i:j} = a_i D_{i+1:j},$$

$$D^{i:j}_{i-1,i+1:j-1,j+1} = c_j D^{i:j-1}_{i-1,i+1:j-1} = a_i D^{i+1:j}_{i+1:j-1,j+1} = a_i c_j D_{i+1:j-1}.$$

Proof. The second part of the lemma is easily obtained by definition. For the first part we have

$$D_{i:j} = \begin{vmatrix} b_i & c_i & & & & & & \\ a_{i+1} & b_{i+1} & c_{i+1} & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & a_k & b_k & c_k & & & \\ & & & a_{k+1} & b_{k+1} & c_{k+1} & & \\ & & & & a_{k+2} & \ddots & \ddots & \\ & & & & & \ddots & b_{j-1} & c_{j-1} \\ & & & & & & a_j & b_j \end{vmatrix}.$$

Now partition the matrix in the above expression into two parts: one formed by the first $k-i+1$ rows and the other formed by the last $j-k$ rows. Let H_{k-i+1}, and H_{j-k} be used to denote the determinant of any submatrix formed by choosing $k-i+1$ columns of the first part, and the corresponding complementary set of columns (there are $j-k$ of them) of the second part, respectively. Now $D_{i:j}$ can be expressed as a sum of $H_{k-i+1}H_{j-k}$ (with appropriate signs) over all possible ways that $k-i+1$ columns can be chosen from $j-i+1$ columns of the first part as long as $H_{k-i+1}H_{j-k} \neq 0$. In this case there are only two choices: selecting from the first group $k-i+1$ columns so that $H_{k-i+1} = D_{i:k}$, $H_{j-k} = D_{k+1:j}$ and the sign is positive; and selecting from the first group the first $k-i$ plus the $k-i$+2-nd column so that $H_{k-i+1} = D^{i:k}_{i:k-1,k+1}$, $H_{j-k} = D^{k+1:j}_{k,k+2:j}$ and the sign is negative. The sum of these two products is then $D_{i:j}$. Thus the lemma is proved. □

LEMMA 2.2. *If exact computations were possible, then the computed results of Algorithm* Sw2 *are such that*

$$Q^{(i+2l)}_{i+1} = \begin{bmatrix} D_{i+1:i+2l} & D^{i+1:i+2l}_{i+1:i+2l-1,i+2l+1} \\ -D^{i+1:i+2l}_{i,i+2:i+2l} & -D^{i+1:i+2l}_{i,i+2:i+2l-1,i+2l+1} \end{bmatrix},$$

where

$$l = 2^{j-1}, \quad i = 0, 1(2^j), 2(2^j), \ldots, n-2^j, \quad n = 2^k, \quad j = 1, 2, \ldots, k.$$

Proof. We prove by substitution. By definition

$$Q^{(i+2l)}_{i+1} \equiv \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = Q^{(i+l)}_{i+1} Q^{(i+2l)}_{i+l+1}. \tag{2.1}$$

Since

$$Q_{i+1}^{(i+l)} = \begin{bmatrix} D_{i+1:i+l} & D_{i+1:i+l-1,i+l+1}^{i+1:i+l} \\ -D_{i,i+2:i+l}^{i+1:i+l} & -D_{i,i+2:i+l-1,i+l+1}^{i+1:i+l} \end{bmatrix},$$

$$Q_{i+l+1}^{(i+2l)} = \begin{bmatrix} D_{i+l+1:i+2l} & D_{i+l+1:i+2l-1,i+2l+1}^{i+l+1:i+2l} \\ -D_{i+l,i+l+2:i+2l}^{i+l+1:i+2l} & -D_{i+l,i+l+2:i+2l-1,i+2l+1}^{i+l+1:i+2l} \end{bmatrix}.$$

Equating both sides of (2.1) then gives us

$$q_{11} = D_{i+1:i+l}D_{i+l+1:i+2l} - D_{i+1:i+l-1,i+l+1}^{i+1:i+l} D_{i+l,i+l+2:i+2l}^{i+l+1:i+2l} = D_{i+1:i+2l}$$

by Lemma 2.1. Similarly, Lemma 2.1 can be used again to give

$$\begin{aligned} q_{12} &= D_{i+1:i+l}D_{i+l+1:i+2l-1,i+2l+1}^{i+l+1:i+2l} - D_{i+1:i+l-1,i+l+1}^{i+1:i+l} D_{i+l,i+l+2:i+2l}^{i+l+1:i+2l} \\ &= c_{i+2l}\left(D_{i+1:i+l}D_{i+l+1:i+2l-1} - D_{i+1:i+l-1,i+l+1}^{i+1:i+l} D_{i+l,i+l+2:i+2l-1}^{i+l+1:i+2l-1}\right) \\ &= c_{i+2l}D_{i+l:i+2l-1} = D_{i+l:i+2l-1,i+2l+1}^{i+1:i+2l}. \end{aligned}$$

The remaining identities can be verified similarly. This completes our proof. □

3. Error analysis of Algorithm Sw1. Given a normalized floating-point system with a τ-digit base β mantissa, the following equations can be assumed to facilitate the error analysis of general arithmetic expressions using only $+, -, *,$ or $/$ operations [3]

$$fl(x\#y) = (x\#y)\Delta, \qquad \# \in \{+, -, *, /\}, \tag{3.1}$$

where x and y are given machine floating-point numbers and $fl(.)$ is used to denote the computed floating-point result of the given argument, and

$$\Delta = 1 + \epsilon, \qquad |\epsilon| \le u = \begin{cases} \beta^{1-\tau}/2 & \text{for rounded operations,} \\ \beta^{1-\tau} & \text{for chopped operations.} \end{cases}$$

We call Δ the unit Δ-factor. For simplicity, we assume that the given matrix A and right-hand side vector y are exact.

We first look at the computation of d_i and e_i. The basic equations are the following:

$$d_0 = 1, \quad d_1 = b_1, \quad e_1 = c_1, \quad d_i = fl(b_i d_{i-1} - a_i e_{i-1}), \quad e_i = fl(c_i d_{i-1}), \quad i \ge 2.$$

By applying (3.1) to the above equations, we obtain $d_0 = 1$, $d_1 = b_1$, $e_1 = c_1$,

$$d_2 = b_2 d_1 \Delta_2\Delta_3 - a_2 e_1 \Delta_1\Delta_3 = \begin{vmatrix} b_1 & c_1 \\ a_2\Delta_1\Delta_3 & b_2\Delta_2\Delta_3 \end{vmatrix},$$

$$e_2 = c_2 b_1 \Delta_4 = \begin{vmatrix} b_1 & \\ a_2\Delta_1\Delta_3 & c_2\Delta_4 \end{vmatrix}.$$

Note the added $\Delta_1\Delta_3$ factor to a_2 in the expression for e_2. Since usually the various Δ-factors are unknown, the generic form Δ^k is used to denote a product of possible k different Δ-factors. Thus we have

$$d_2 = \begin{vmatrix} b_1 & c_1 \\ a_2\Delta^2 & b_2\Delta^2 \end{vmatrix}, \qquad e_2 = \begin{vmatrix} b_1 & \\ a_2\Delta^2 & c_2\Delta \end{vmatrix}. \tag{3.2a}$$

Similarly,

$$d_3 = b_3d_2\Delta^2 - a_3e_2\Delta^2 = \begin{vmatrix} d_2 & e_2 \\ a_3\Delta^2 & b_3\Delta^2 \end{vmatrix} = \begin{vmatrix} b_1 & c_1 & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta \\ & a_3\Delta^2 & b_3\Delta^2 \end{vmatrix}, \tag{3.2b}$$

$$e_3 = c_3d_2\Delta = \begin{vmatrix} b_1 & c_1 & \\ a_2\Delta^2 & b_2\Delta^2 & \\ & a_3\Delta^2 & c_3\Delta \end{vmatrix}. \tag{3.2c}$$

By induction we then obtain the following theorem.

THEOREM 3.1. *The computed d_i and e_i using Algorithm* Sw1 *satisfy* (3.2) *for $i = 2, 3$. And generally for $3 < i \leq n$ they are such that*

$$d_i = \begin{vmatrix} b_1 & c_1 & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{i-1}\Delta^2 & b_{i-1}\Delta^2 & c_{i-1}\Delta & \\ & & & a_i\Delta^2 & b_i\Delta^2 & \end{vmatrix},$$

$$e_i = \begin{vmatrix} b_1 & c_1 & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{i-2}\Delta^2 & b_{i-2}\Delta^2 & c_{i-2}\Delta & \\ & & & a_{i-1}\Delta^2 & b_{i-1}\Delta^2 & \\ & & & & a_i\Delta^2 & c_i\Delta \end{vmatrix}.$$

By similar reasonings, one can obtain the following theorem.

THEOREM 3.2. *The computed f_i and g_i using Algorithm* Sw1 *are such that*

$$f_i = \begin{vmatrix} b_i\Delta^2 & c_i\Delta^2 & & & \\ a_{i+1}\Delta & b_{i+1}\Delta^2 & c_{i+1}\Delta^2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1}\Delta & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ & & & a_n & b_n \end{vmatrix}, \qquad 1 \le i \le n-1,$$

$$g_{n-1} = \begin{vmatrix} a_{n-1}\Delta & c_{n-1}\Delta^2 \\ & b_n \end{vmatrix}, \qquad g_{n-2} = \begin{vmatrix} a_{n-2}\Delta & c_{n-2}\Delta^2 & \\ & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ & a_n & b_n \end{vmatrix},$$

$$g_i = \begin{vmatrix} a_i\Delta & c_i\Delta^2 & & & & \\ & b_{i+1}\Delta^2 & c_{i+1}\Delta^2 & & & \\ & a_{i+2}\Delta & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-1}\Delta & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ & & & & a_n & b_n \end{vmatrix}, \qquad 1 \le i < n-2.$$

The computation of t_i and u_j are more complicated. For t_2, we have

$$t_2 = fl(-a_2t_1 + d_1y_2) = -a_2t_1\Delta^2 + d_1y_2\Delta^2 = \begin{vmatrix} b_1 & y_1 \\ a_2\Delta^2 & y_2\Delta^2 \end{vmatrix}. \tag{3.3a}$$

Similarly for t_3, we have

$$t_3 = \begin{vmatrix} d_2 & t_2 \\ a_3\Delta^2 & y_3\Delta^2 \end{vmatrix}.$$

Now

$$d_2 = \begin{vmatrix} b_1 & c_1 \\ a_2\Delta^2 & b_2\Delta^2 \end{vmatrix}.$$

Although the expressions for d_2 and t_2 share the same $a_2\Delta^2$ in the above equations, they are most likely different because d_2 and t_2 are computed in separate occasions. Thus strictly speaking, one cannot write the equality

$$t_3 = \begin{vmatrix} b_1 & c_1 & y_1 \\ a_2\Delta^2 & b_2\Delta^2 & y_2\Delta^2 \\ & a_3\Delta^2 & y_3\Delta^2 \end{vmatrix}. \tag{3.3b}$$

However, the above equality is valid if the right-hand side is interpreted as a compact representation of the sum of products contained in the expansion of the determinant. Because then $a_2\Delta^2$ can be part of many different products and one can assign different values for Δ^2 if necessary. Henceforth the above interpretation of equality will be used in our discussions. Note that the new interpretation does not affect the validity of Theorems 3.1 and 3.2.

For u_{n-1} and u_{n-2}, we have

$$(3.3c)\qquad u_{n-1} = \begin{vmatrix} y_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ y_n & b_n \end{vmatrix}, \qquad u_{n-2} = \begin{vmatrix} y_{n-2}\Delta^2 & c_{n-2}\Delta^2 & \\ y_{n-1}\Delta^2 & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ y_n & a_n & b_n \end{vmatrix}.$$

In general, we have the following theorem.

THEOREM 3.3. *The computed t_i and u_j using Algorithm* Sw1 *are such that t_2, t_3, u_{n-1}, and u_{n-2} satisfy* (3.3) *and for $3 < i \le n$ and $1 \le j < n-2$,*

$$(3.4)\qquad t_i = \begin{vmatrix} b_1 & c_1 & & & & y_1 \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta & & & y_2\Delta^2 \\ & \ddots & \ddots & \ddots & & \vdots \\ & & a_{i-2}\Delta^2 & b_{i-2}\Delta^2 & c_{i-2}\Delta & y_{i-2}\Delta^2 \\ & & & a_{i-1}\Delta^2 & b_{i-1}\Delta^2 & y_{i-1}\Delta^2 \\ & & & & a_i\Delta^2 & y_i\Delta^2 \end{vmatrix},$$

$$u_j = \begin{vmatrix} y_j\Delta^2 & c_j\Delta^2 & & & & \\ y_{j+1}\Delta^2 & b_{j+1}\Delta^2 & c_{j+1}\Delta^2 & & & \\ y_{j+2}\Delta^2 & a_{j+2}\Delta & b_{j+2}\Delta^2 & c_{j+2}\Delta^2 & & \\ \vdots & & \ddots & \ddots & \ddots & \\ y_{n-1}\Delta^2 & & & a_{n-1}\Delta & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ y_n & & & & a_n & b_n \end{vmatrix}.$$

Proof. By construction we have

$$t_i = fl(-a_i t_{i-1} + d_{i-1}y_i) = -a_i t_{i-1}\Delta^2 + d_{i-1}y_i\Delta^2.$$

Now the structure of d_{i-1} is given in Theorem 3.1. If we assume that Theorem 3.3 is true for t_{i-1}, then the above expression is exactly equivalent to the expression obtained when expanding the right-hand side determinant of (3.4) by its last row. Hence (3.4) is valid. Similar reasonings can also be used to verify the expression for u_j. This time

the expansion of its right-hand side determinant is by its first row. This completes our proof. □

From Theorems 3.1, 3.2, and 3.3 one can obtain the following theorem.

THEOREM 3.4. *The computed x_i for $1 \le i \le n$ are such that*

$$x_1 = \frac{u_1\Delta}{d_n} = \frac{\hat{u}_1}{d_n}, \qquad \hat{u}_1 = \begin{vmatrix} y_1\Delta^3 & c_1\Delta^2 & & & & \\ y_2\Delta^3 & b_2\Delta^2 & c_2\Delta^2 & & & \\ y_3\Delta^3 & a_3\Delta & b_3\Delta^2 & c_3\Delta^2 & & \\ \vdots & & \ddots & \ddots & \ddots & \\ y_{n-1}\Delta^3 & & & a_{n-1}\Delta & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ y_n\Delta & & & & a_n & b_n \end{vmatrix},$$

$$x_n = \frac{t_n\Delta}{d_n} = \frac{\hat{t}_n}{d_n}, \qquad \hat{t}_n = \begin{vmatrix} b_1 & c_1 & & & & y_1\Delta \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta & & & y_2\Delta^3 \\ & \ddots & \ddots & \ddots & & \vdots \\ & & a_{n-2}\Delta^2 & b_{n-2}\Delta^2 & c_{n-2}\Delta & y_{n-2}\Delta^3 \\ & & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & y_{n-1}\Delta^3 \\ & & & & a_n\Delta^2 & y_n\Delta^3 \end{vmatrix},$$

$$x_i = \frac{\begin{vmatrix} d_{i-1} & t_{i-1}\Delta^3 \\ g_i & u_i\Delta^3 \end{vmatrix}}{d_n} = \frac{\xi_i}{d_n},$$

where

$$\xi_i = \begin{vmatrix} b_1 & c_1 & & & & y_1\Delta^3 & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta & & & y_2\Delta^5 & & & & \\ & \cdots & \cdots & & & \cdots & & & & \\ & a_{i-2}\Delta^2 & b_{i-2}\Delta^2 & c_{i-2}\Delta & & y_{i-2}\Delta^5 & & & & \\ & & a_{i-1}\Delta^2 & b_{i-1}\Delta^2 & & y_{i-1}\Delta^5 & & & & \\ & & & a_i\Delta & & y_i\Delta^5 & c_i\Delta^2 & & & \\ & & & & & y_{i+1}\Delta^5 & b_{i+1}\Delta^2 & c_{i+1}\Delta^2 & & \\ & & & & & y_{i+2}\Delta^5 & a_{i+2}\Delta & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & \\ & & & & & \cdots & & \cdots & \cdots & \\ & & & & & y_{n-1}\Delta^5 & & a_{n-1}\Delta & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ & & & & & y_n\Delta^3 & & & a_n & b_n \end{vmatrix}.$$

Proof. Results for x_1 and x_n are straightforward by using Theorem 3.3. For x_i, one can expand ξ_i using a similar approach as described in the proof of Lemma 2.1 to show that

$$\xi_i = d_{i-1}u_i\Delta^3 - t_{i-1}g_i\Delta^3 = \begin{vmatrix} d_{i-1} & t_{i-1}\Delta^3 \\ g_i & u_i\Delta^3 \end{vmatrix}.$$

This completes our proof. □

4. Error analysis of Algorithm Sw2. To give consistent and compact expressions similar to Theorems 3.1–3.4 for Algorithm Sw2, we introduce the relation $\preceq$ as follows: If for a certain matrix B, we have

$$p^* = \det B = \sum_{j=1}^{\lambda(\det B)} p_j^*,$$

where p_j^*'s are exact products of error-free data and $\lambda(\det B)$ is the total number of such products in $\det B$, then

$$p = fl(\det B) = \sum_{j=1}^{\lambda(\det B)} p_j^*\Delta^{\gamma_j^*} \preceq \sum_{j=1}^{\lambda(\det B)} p_j^*\Delta^{\gamma_j} = \det \tilde{B} \quad \text{if} \quad \gamma_j^* \le \gamma_j \quad \text{for all } j.$$

In other words, γ_j is an upper bound of γ_j^*. Thus, for example,

$$\begin{vmatrix} b_1 & c_1 \\ a_2\Delta^2 & b_2\Delta^2 \end{vmatrix} \preceq \begin{vmatrix} b_1\Delta & c_1 \\ a_2\Delta^2 & b_2\Delta^2 \end{vmatrix} \preceq \begin{vmatrix} b_1\Delta & c_1\Delta \\ a_2\Delta^2 & b_2\Delta^2 \end{vmatrix}.$$

Furthermore, we use the following model in evaluating a general 2×2 determinant:

$$fl\left(\begin{vmatrix} x & y \\ z & w \end{vmatrix}\right) = \begin{vmatrix} x\Delta & y\Delta \\ z\Delta & w\Delta \end{vmatrix}.$$

For simplicity, the same $D_{\delta_1,\delta_2,\ldots,\delta_k}^{\beta_1,\beta_2,\ldots,\beta_k}$ is used to denote the computed determinant of a submatrix formed by the common elements of rows $\beta_1, \beta_2, \ldots, \beta_k$ and columns $\delta_1, \delta_2, \ldots, \delta_k$ of the matrix A.

Consider the computation of $Q_{i+1}^{(i+2l)}$ for $n = 8$ first. It proceeds as follows: stage 1, the computation of $Q_1^{(2)}, Q_3^{(4)}, Q_5^{(6)}, Q_7^{(8)}$; stage 2, the computation of $Q_1^{(4)}, Q_5^{(8)}$; stage 3, the computation of $Q_1^{(8)}$. We give the computed $Q_1^{(2)}, Q_3^{(4)}$, and $Q_1^{(4)}$ first.

$$Q_1^{(2)} = \begin{bmatrix} D_{1:2} & D_{1,3}^{1:2} \\ 0 & 0 \end{bmatrix}, \quad Q_3^{(4)} = \begin{bmatrix} D_{3:4} & D_{3,5}^{3:4} \\ -D_{2,4}^{3:4} & -D_{2,5}^{3:4} \end{bmatrix}, \quad Q_1^{(4)} = \begin{bmatrix} D_{1:4} & D_{1:3,5}^{1:4} \\ 0 & 0 \end{bmatrix},$$

where

$$D_{1:2} = \begin{vmatrix} b_1\Delta & c_1\Delta \\ a_2\Delta & b_2\Delta \end{vmatrix}, \quad D_{3:4} = \begin{vmatrix} b_3\Delta & c_3\Delta \\ a_4\Delta & b_4\Delta \end{vmatrix}, \quad D_{1,3}^{1:2} = \begin{vmatrix} b_1 & 0 \\ a_2\Delta & c_2\Delta \end{vmatrix} \preceq \begin{vmatrix} b_1\Delta & 0 \\ a_2\Delta & c_2\Delta \end{vmatrix},$$

$$D_{3,5}^{3:4} \preceq \begin{vmatrix} b_3\Delta & 0 \\ a_4\Delta & c_4\Delta \end{vmatrix}, \quad D_{2,4}^{3:4} \preceq \begin{vmatrix} a_3\Delta & c_3\Delta \\ 0 & b_4\Delta \end{vmatrix}, \quad D_{2,5}^{3:4} = \begin{vmatrix} a_3 & 0 \\ 0 & c_4\Delta \end{vmatrix} \preceq \begin{vmatrix} a_3\Delta & 0 \\ 0 & c_4\Delta \end{vmatrix},$$

and

$$D_{1:4} = D_{1:2}D_{3:4}\Delta^2 - D_{1,3}^{1:2}D_{2,4}^{3:4}\Delta^2 = (D_{1:2}\Delta)(D_{3:4}\Delta) - (D_{1,3}^{1:2}\Delta)(D_{2,4}^{3:4}\Delta)$$

$$\preceq \begin{vmatrix} b_1\Delta & c_1\Delta \\ a_2\Delta^2 & b_2\Delta^2 \end{vmatrix} \begin{vmatrix} b_3\Delta^2 & c_3\Delta^2 \\ a_4\Delta & b_4\Delta \end{vmatrix} - \begin{vmatrix} b_1\Delta & 0 \\ a_2\Delta^2 & c_2\Delta^2 \end{vmatrix} \begin{vmatrix} a_3\Delta^2 & c_3\Delta^2 \\ 0 & b_4\Delta \end{vmatrix}$$

$$\preceq \begin{vmatrix} b_1\Delta & c_1\Delta & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & \\ & a_3\Delta^2 & b_3\Delta^2 & c_3\Delta^2 \\ & & a_4\Delta & b_4\Delta \end{vmatrix},$$

$$D_{1:3,5}^{1:4} = D_{1:2}D_{3,5}^{3:4}\Delta^2 - D_{1,3}^{1:2}D_{2,5}^{3:4}\Delta^2 \preceq \begin{vmatrix} b_1\Delta & c_1\Delta & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & \\ & a_3\Delta^2 & b_3\Delta^2 & \\ & & a_4\Delta & c_4\Delta \end{vmatrix}.$$

Similar expressions can also be obtained for $Q_5^{(6)}$ and $Q_7^{(6)}$. We show only the computed $Q_5^{(8)}$ and $Q_1^{(8)}$:

$$Q_5^{(8)} = \begin{bmatrix} D_{5:8} & 0 \\ -D_{4,6:8}^{5:8} & 0 \end{bmatrix}, \qquad Q_1^{(8)} = \begin{bmatrix} D_{1:8} & 0 \\ 0 & 0 \end{bmatrix},$$

where

$$D_{5:8} \preceq \begin{vmatrix} b_5\Delta & c_5\Delta & & \\ a_6\Delta^2 & b_6\Delta^2 & c_6\Delta^2 & \\ & a_7\Delta^2 & b_7\Delta^2 & c_7\Delta^2 \\ & & a_8\Delta & b_8\Delta \end{vmatrix}, \qquad D_{4,6:8}^{5:8} \preceq \begin{vmatrix} a_5\Delta & c_5\Delta & & \\ & b_6\Delta^2 & c_6\Delta^2 & \\ & a_7\Delta^2 & b_7\Delta^2 & c_7\Delta^2 \\ & & a_8\Delta & b_8\Delta \end{vmatrix},$$

$$D_{1:8} \preceq \begin{vmatrix} b_1\Delta & c_1\Delta & & & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & & & \\ & a_3\Delta^2 & b_3\Delta^2 & c_3\Delta^2 & & & & \\ & & a_4\Delta^2 & b_4\Delta^2 & c_4\Delta^2 & & & \\ & & & a_5\Delta^2 & b_5\Delta^2 & c_5\Delta^2 & & \\ & & & & a_6\Delta^2 & b_6\Delta^2 & c_6\Delta^2 & \\ & & & & & a_7\Delta^2 & b_7\Delta^2 & c_7\Delta^2 \\ & & & & & & a_8\Delta & b_8\Delta \end{vmatrix}.$$

In general, the following theorem can be obtained easily by induction.

THEOREM 4.1. *The computed* $Q_{i+1}^{(i+2l)}$ *are such that for*

$$l = 2^{j-1}, \quad i = 0, 1(2^j), 2(2^j), \ldots, n - 2^j, \quad n = 2^k, \quad j = 1, 2, \ldots, k,$$

$$Q_{i+1}^{(i+2l)} = \begin{bmatrix} D_{i+1:i+2l}^{i+1:i+2l} & D_{i+1:i+2l-1,i+2l+1}^{i+1:i+2l} \\ -D_{i,i+2:i+2l}^{i+1:i+2l} & -D_{i,i+2:i+2l-1,i+2l+1}^{i+1:i+2l} \end{bmatrix},$$

where

$$D_{i+1,i+2} = \begin{vmatrix} b_{i+1}\Delta & c_{i+1}\Delta \\ a_{i+2}\Delta & b_{i+2}\Delta \end{vmatrix}, \qquad D_{i+1,i+3}^{i+1,i+2} \preceq \begin{vmatrix} b_{i+1}\Delta & 0 \\ a_{i+2}\Delta & c_{i+2}\Delta \end{vmatrix},$$

$$D_{i,i+2}^{i+1,i+2} \preceq \begin{vmatrix} a_{i+1}\Delta & c_{i+1}\Delta \\ 0 & b_{i+2}\Delta \end{vmatrix}, \qquad D_{i,i+3}^{i+1,i+2} \preceq \begin{vmatrix} a_{i+1}\Delta & 0 \\ 0 & c_{i+2}\Delta \end{vmatrix},$$

$$D_{i+1:i+2l} \preceq \hat{D}_{i+1:i+2l} = \begin{vmatrix} b_{i+1}\Delta & c_{i+1}\Delta & & & \\ a_{i+2}\Delta^2 & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & c_{i+2l-1}\Delta^2 \\ & & & a_{i+2l}\Delta & b_{i+2l}\Delta \end{vmatrix},$$

$$D^{i+1:i+2t}_{i+1:i+2l-1,i+2l+1} \preceq \hat{D}^{i+1:i+2l}_{i+1:i+2l-1,i+2l+1}$$

$$= \begin{vmatrix} b_{i+1}\Delta & c_{i+1}\Delta & & & & & \\ a_{i+2}\Delta^2 & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_{i+2l-2}\Delta^2 & b_{i+2l-2}\Delta^2 & c_{i+2l-2}\Delta^2 & & \\ & & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & & \\ & & & & a_{i+2l}\Delta & c_{i+2l}\Delta \end{vmatrix},$$

$$D^{i+1:i+2l}_{i,i+2:i+2l} \preceq \hat{D}^{i+1:i+2l}_{i,i+2:i+2l}$$

$$= \begin{vmatrix} a_{i+1}\Delta & c_{i+1}\Delta & & & & \\ & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & & \\ & a_{i+3}\Delta^2 & b_{i+3}\Delta^2 & c_{i+3}\Delta^2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & c_{i+2l-1}\Delta^2 \\ & & & & a_{i+2l}\Delta & b_{i+2l}\Delta \end{vmatrix},$$

$$D^{i+1:i+2l}_{i,i+2:i+2l-1,i+2l+1} \preceq \hat{D}^{i+1:i+2l}_{i,i+2:i+2l-1,i+2l+1}$$

$$= \begin{vmatrix} a_{i+1}\Delta & c_{i+1}\Delta & & & & & \\ & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & & & \\ & a_{i+3}\Delta^2 & b_{i+3}\Delta^2 & c_{i+3}\Delta^2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & a_{i+2l-2}\Delta^2 & b_{i+2l-2}\Delta^2 & c_{i+2l-2}\Delta^2 & \\ & & & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & \\ & & & & & a_{i+2l}\Delta & c_{i+2l}\Delta \end{vmatrix}.$$

Before we turn our attention to other parts of Algorithm Sw2, we need to define some additional terms. For any positive integer i, let i be expressed in powers of 2 as follows:

$$i = 2^{s_1} + 2^{s_2} + \cdots + 2^{s_j}, \quad 0 \le s_1 < s_2 < \cdots < s_j, \quad 1 \le j.$$

Then we can define the following vectors:

$$e^{(2^{s_l})} = \sum_{p=1}^{2^{s_l}} e_p^{(2^{s_l})}, \qquad q^{(i)} = \begin{bmatrix} q_1^{(i)} \\ q_2^{(i)} \\ \vdots \\ q_i^{(i)} \end{bmatrix} = \begin{bmatrix} q^{(2^{s_1})} + e^{(2^{s_1})} \\ q^{(2^{s_2})} + e_1^{(2^{s_2})} + e^{(2^{s_2})} \\ \vdots \\ q^{(2^{s_{j-1}})} + e_1^{(2^{s_{j-1}})} + e^{(2^{s_{j-1}})} \\ q^{(2^{s_j})} + e_1^{(2^{s_j})} \end{bmatrix},$$

where each $q^{(2^{s_l})}$ is a 2^{s_l}-vector, $e_p^{(2^{s_l})}$ is the pth column of a 2^{s_l} by 2^{s_l} identity matrix, and

$$q^{(2^0)} = [0], \quad q^{(2^1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad q^{(2^{s_j})} = \begin{bmatrix} q^{(2^{s_j-1})} + e^{(2^{s_j-1})} \\ q^{(2^{s_j-1})} \end{bmatrix} \quad \text{for } s_j \geq 2.$$

One can also easily establish the following bounds:

$$1 \leq q_j^{(i)} \leq \lceil \log_2 n \rceil, \qquad 1 \leq j \leq i. \tag{4.1}$$

As an example, let $i = 7 = 2^0 + 2^1 + 2^2$, then

$$q^{(7)} = \begin{bmatrix} q^{(1)} + e^{(1)} \\ q^{(2)} + e_1^{(1)} + e^{(1)} \\ q^{(4)} + e^{(4)} \end{bmatrix} = [1, 3, 2, 3, 3, 2, 2]^T.$$

We have the following theorems.

THEOREM 4.2. *The computed* $\alpha_{i+1}^{(i+2l)}$, $\gamma_{i+1}^{(i+2l)}$, $t_{i+1}^{(i+2l)}$, *and* $u_{i+1}^{(i+2l)}$ *are such that for*

$$l = 2^{j-1}, \quad i = 0, 1(2^j), 2(2^j), \ldots, n - 2^j, \quad n = 2^k, \quad j = 1, 2, \ldots, k,$$

$$\alpha_{i+1}^{(i+2l)} \preceq \hat{\alpha}_{i+1}^{(i+2l)}$$

$$= \begin{vmatrix} a_{i+1}\Delta & b_{i+1}\Delta & c_{i+1}\Delta & & & & \\ & a_{i+2}\Delta^2 & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & a_{i+2l-2}\Delta^2 & b_{i+2l-2}\Delta^2 & c_{i+2l-2}\Delta^2 & \\ & & & & a_{i+2l-1}\Delta & b_{i+2l-1}\Delta & \\ & & & & & a_{i+2l}\Delta & \end{vmatrix},$$

$$\gamma_{i+1}^{(i+2l)} \preceq \hat{\gamma}_{i+1}^{(i+2l)}$$

$$= \begin{vmatrix} c_{i+1}\Delta & & & & & & \\ b_{i+2}\Delta & c_{i+2}\Delta & & & & & \\ a_{i+3}\Delta^2 & b_{i+3}\Delta^2 & c_{i+3}\Delta^2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & c_{i+2l-1}\Delta^2 & & \\ & & & a_{i+2l}\Delta & b_{i+2l}\Delta & c_{i+2l}\Delta \end{vmatrix},$$

where $\alpha_{i+1}^{(i+2l)}$ *and* $\gamma_{i+1}^{(i+2l)}$ *are expressed as the determinant of an upper and lower triangular matrix, respectively. Also,*

$$t_{i+1}^{(i+2l)} \preceq \hat{t}_{i+1}^{(i+2l)}$$

$$= \begin{vmatrix} b_1\Delta & c_1\Delta & & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & & \\ & \ddots & \ddots & \ddots & & & y_{i+1}\Delta^{q_{2l}^{(2l)}} \\ & & \ddots & \ddots & \ddots & & \vdots \\ & & & a_{i+2l-2}\Delta^2 & b_{i+2l-2}\Delta^2 & c_{i+2l-2}\Delta^2 & y_{i+2l-2}\Delta^{q_3^{(2l)}} \\ & & & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & y_{i+2l-1}\Delta^{q_2^{(2l)}} \\ & & & & & a_{i+2l}\Delta & y_{i+2l}\Delta^{q_1^{(2l)}} \end{vmatrix},$$

$$u_{i+1}^{(i+2l)} \preceq \hat{u}_{i+1}^{(i+2l)}$$

$$= \begin{vmatrix} y_{i+1}\Delta^{q_1^{(2l)}} & c_{i+1}\Delta & & & & & \\ y_{i+2}\Delta^{q_2^{(2l)}} & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & & & \\ y_{i+3}\Delta^{q_3^{(2l)}} & a_{i+3}\Delta^2 & b_{i+3}\Delta^2 & c_{i+3}\Delta^2 & & & \\ \vdots & & \ddots & \ddots & \ddots & & \\ y_{i+2l}\Delta^{q_{2l}^{(2l)}} & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ & & & & & a_n\Delta & b_n\Delta \end{vmatrix}.$$

Proof. We prove by substitution. For $\alpha_{i+1}^{(i+2l)}$, we have

$$\alpha_{i+1}^{(i+2l)} = fl\left(\alpha_{i+1}^{(i+l)}\alpha_{i+l+1}^{(i+2l)}\right) = \alpha_{i+1}^{(i+l)}\alpha_{i+l+1}^{(i+2l)}\Delta \preceq \hat{\alpha}_{i+1}^{(i+l)}\hat{\alpha}_{i+l+1}^{(i+2l)}\Delta.$$

Now

$$\hat{\alpha}_{i+1}^{(i+l)} \preceq \begin{vmatrix} a_{i+1}\Delta & b_{i+1}\Delta & c_{i+1}\Delta & & & & \\ & a_{i+2}\Delta^2 & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & a_{i+l-2}\Delta^2 & b_{i+l-2}\Delta^2 & c_{i+l-2}\Delta^2 & \\ & & & & a_{i+l-1}\Delta^2 & b_{i+l-1}\Delta^2 & \\ & & & & & a_{i+l}\Delta^2 & \end{vmatrix}$$

because the right-hand side expression in the above equation is a modified $\hat{\alpha}_{i+1}^{(i+l)}$ with its last two rows multiplied by Δ. Similarly,

$$\hat{\alpha}_{i+l+1}^{(i+2l)}\Delta \preceq \begin{vmatrix} a_{i+l+1}\Delta^2 & b_{i+l+1}\Delta^2 & c_{i+l+1}\Delta^2 & & & & \\ & a_{i+l+2}\Delta^2 & b_{i+l+2}\Delta^2 & c_{i+l+2}\Delta^2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & a_{i+2l-2}\Delta^2 & b_{i+2l-2}\Delta^2 & c_{i+2l-2}\Delta^2 & \\ & & & & a_{i+2l-1}\Delta & b_{i+2l-1}\Delta & \\ & & & & & a_{i+2l}\Delta & \end{vmatrix}.$$

Hence

$$\alpha_{i+1}^{(i+2l)} \preceq \hat{\alpha}_{i+1}^{(i+l)}\hat{\alpha}_{i+l+1}^{(i+2l)}\Delta \preceq \hat{\alpha}_{i+1}^{(i+2l)}.$$

For $t_{i+1}^{(i+2l)}$, we have

$$\begin{aligned} t_{i+1}^{(i+2l)} &= fl\left(t_{i+l+1}^{(i+2l)} + \alpha_{i+l+1}^{(i+2l)}t_{i+1}^{(i+l)}\right) = t_{i+l+1}^{(i+2l)}\Delta + \alpha_{i+l+1}^{(i+2l)}t_{i+1}^{(i+l)}\Delta^2 \\ &\preceq \hat{t}_{i+l+1}^{(i+2l)}\Delta + \hat{\alpha}_{i+l+1}^{(i+2l)}\Delta^2\hat{t}_{i+1}^{(i+l)}. \end{aligned}$$

Now

$$\hat{t}_{i+1}^{(i+l)} = \begin{vmatrix} b_1\Delta & c_1\Delta & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & \\ & \ddots & \ddots & \ddots & & y_{i+1}\Delta^{q_l^{(l)}} \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & a_{i+l-2}\Delta^2 & b_{i+l-2}\Delta^2 & c_{i+l-2}\Delta^2 & y_{i+l-2}\Delta^{q_3^{(l)}} \\ & & & & a_{i+l-1}\Delta^2 & b_{i+l-1}\Delta^2 & y_{i+l-1}\Delta^{q_2^{(l)}} \\ & & & & & a_{i+l}\Delta & y_{i+l}\Delta^{q_1^{(l)}} \end{vmatrix},$$

and in $\hat{\alpha}_{i+l+1}^{(i+2l)}\Delta^2$ one can multiply one of the Δ-factor to the first row of $\hat{\alpha}_{i+l+1}^{(i+2l)}$ and the other one to the next to the last row and obtain

$$\hat{\alpha}_{i+l+1}^{(i+2l)}\Delta^2 = \begin{vmatrix} a_{i+l+1}\Delta^2 & b_{i+l+1}\Delta^2 & c_{i+l+1}\Delta^2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{i+2l-2}\Delta^2 & b_{i+2l-2}\Delta^2 & c_{i+2l-2}\Delta^2 & \\ & & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & \\ & & & & a_{i+2l}\Delta & \end{vmatrix}.$$

Furthermore, the Δ-factor in $\hat{t}_{i+l+1}^{(i+2l)}\Delta$ can be multiplied to the last column of $\hat{t}_{i+l+1}^{(i+2l)}$ to obtain

$\hat{t}_{i+l+1}^{(i+2l)}\Delta =$

$$\begin{vmatrix} b_1\Delta & c_1\Delta & & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_{i+l+1}\Delta^2 & b_{i+l+1}\Delta^2 & c_{i+l+1}\Delta^2 & & y_{i+l+1}\Delta^{q_l^{(l)}+1} \\ & & & \ddots & \ddots & \ddots & \vdots \\ & & & & a_{i+2l-2}\Delta^2 & b_{i+2l-2}\Delta^2 & c_{i+2l-2}\Delta^2 & y_{i+2l-2}\Delta^{q_3^{(l)}+1} \\ & & & & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & y_{i+2l-1}\Delta^{q_2^{(l)}+1} \\ & & & & & & a_{i+2l}\Delta & y_{i+2l}\Delta^{q_1^{(l)}+1} \end{vmatrix}.$$

Now

$$\hat{t}_{i+l+1}^{(i+2l)}\Delta + \hat{\alpha}_{i+l+1}^{(i+2l)}\Delta^2\hat{t}_{i+1}^{(i+l)} =$$

$$\begin{vmatrix} b_1\Delta & c_1\Delta & & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & & \\ & \ddots & \ddots & \ddots & & & y_{i+1}\Delta^{q_l^{(l)}} \\ & & \ddots & \ddots & \ddots & & \vdots \\ & & & a_{i+l}\Delta^2 & b_{i+l}\Delta^2 & c_{i+l}\Delta^2 & y_{i+l}\Delta^{q_1^{(l)}} \\ & & & & \ddots & \ddots & \ddots & y_{i+l+1}\Delta^{q_l^{(l)}+1} \\ & & & & & a_{i+2l-1}\Delta^2 & b_{i+2l-1}\Delta^2 & \vdots \\ & & & & & & a_{i+2l}\Delta & y_{i+2l}\Delta^{q_1^{(l)}+1} \end{vmatrix}$$

since one can expand the above determinant by splitting the last column into two parts with nontrivial parts formed by the vectors

$$\begin{bmatrix} y_{i+1}\Delta^{q_l^{(l)}} \\ \vdots \\ y_{i+l}\Delta^{q_1^{(l)}} \end{bmatrix}, \quad \begin{bmatrix} y_{i+l+1}\Delta^{q_l^{(l)}+1} \\ \vdots \\ y_{i+2l}\Delta^{q_1^{(l)}+1} \end{bmatrix},$$

and their sum is precisely $\hat{t}_{i+l+1}^{(i+2l)}\Delta + \hat{\alpha}_{i+l+1}^{(i+2l)}\Delta^2\hat{t}_{i+1}^{(i+l)}$. Furthermore, by using the recursive definition

$$q^{(2^j)} = \begin{bmatrix} q^{(2^{j-1})} + e^{(2^{j-1})} \\ q^{(2^{j-1})} \end{bmatrix},$$

the nontrivial parts of the last column can be expressed as

$$\begin{bmatrix} y_{i+1}\Delta^{q_{2l}^{(2l)}} & \cdots & y_{i+l}\Delta^{q_{l+1}^{(2l)}} & y_{i+l+1}\Delta^{q_l^{(2l)}} & \cdots & y_{i+2l}\Delta^{q_1^{(2l)}} \end{bmatrix}^T.$$

Thus

$$\hat{t}_{i+l+1}^{(i+2l)}\Delta + \hat{\alpha}_{i+l+1}^{(i+2l)}\Delta^2\hat{t}_{i+1}^{(i+l)} = \hat{t}_{i+1}^{i+2l}.$$

The validity of the expressions for γ_{i+1}^{i+2l} and u_{i+1}^{i+2l} can also be proved using similar reasonings. This completes our proof. □

The following theorems can also be proved easily by induction.

THEOREM 4.3. *The computed d_i, e_i, and $t_1^{(i)}$ for $n \geq i \geq 2$ are such that*

$$d_i \preceq \hat{d}_i = \begin{vmatrix} b_1\Delta & c_1\Delta & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{i-1}\Delta^2 & b_{i-1}\Delta^2 & c_{i-1}\Delta^2 & \\ & & & a_i\Delta & b_i\Delta & \end{vmatrix},$$

$$e_i \preceq \hat{e}_i = \begin{vmatrix} b_1\Delta & c_1\Delta & & & & \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{i-2}\Delta^2 & b_{i-2}\Delta^2 & c_{i-2}\Delta^2 & \\ & & & a_{i-1}\Delta^2 & b_{i-1}\Delta^2 & \\ & & & & a_i\Delta & c_i\Delta \end{vmatrix},$$

$$t_1^{(i)} \preceq \hat{t}_1^{(i)} = \begin{vmatrix} b_1\Delta & c_1\Delta & & & & y_1\Delta^{q_i^{(i)}} \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & y_2\Delta^{q_{i-1}^{(i)}} \\ & \ddots & \ddots & \ddots & & \vdots \\ & & a_{i-2}\Delta^2 & b_{i-2}\Delta^2 & c_{i-2}\Delta^2 & y_{i-2}\Delta^{q_3^{(i)}} \\ & & & a_{i-1}\Delta^2 & b_{i-1}\Delta^2 & y_{i-1}\Delta^{q_2^{(i)}} \\ & & & & a_i\Delta & y_i\Delta^{q_1^{(i)}} \end{vmatrix}.$$

THEOREM 4.4. *The computed f_i, g_i, and $u_i^{(n)}$ for $n-1 \geq i \geq 1$ are such that*

$$f_i \preceq \hat{f}_i = \begin{vmatrix} b_i\Delta & c_i\Delta & & & \\ a_{i+1}\Delta^2 & b_{i+1}\Delta^2 & c_{i+1}\Delta^2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ & & & a_n\Delta & b_n\Delta \end{vmatrix},$$

$$g_i \preceq \hat{g}_i = \begin{vmatrix} a_i\Delta & c_i\Delta & & & \\ & b_{i+1}\Delta^2 & c_{i+1}\Delta^2 & & \\ & a_{i+2}\Delta^2 & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & \\ & & \ddots & \ddots & \ddots \\ & & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ & & & & a_n\Delta & b_n\Delta \end{vmatrix},$$

$$u_i^{(n)} \preceq \hat{u}_i^{(n)} = \begin{vmatrix} y_i\Delta^{q_1^{(n-i+1)}} & c_i\Delta & & & & \\ y_{i+1}\Delta^{q_2^{(n-i+1)}} & b_{i+1}\Delta^2 & c_{i+1}\Delta^2 & & & \\ y_{i+2}\Delta^{q_3^{(n-i+1)}} & a_{i+2}\Delta^2 & b_{i+2}\Delta^2 & c_{i+2}\Delta^2 & & \\ \vdots & & \ddots & \ddots & \ddots & \\ y_{n-1}\Delta^{q_{n-i}^{(n-i+1)}} & & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ y_n\Delta^{q_{n-i+1}^{(n-i+1)}} & & & & a_n\Delta & b_n\Delta \end{vmatrix}.$$

From Theorems 4.1–4.4 one can easily obtain the following theorem.

THEOREM 4.5. *The computed x_i for $1 \le i \le n$ are such that*

$$x_1 = \frac{u_1^{(n)}\Delta}{d_n},$$
$$x_n = \frac{t_1^{(n)}\Delta}{d_n},$$

where

$$u_1^{(n)}\Delta \preceq \hat{u}_1^{(n)} = \begin{vmatrix} y_1\Delta^{1+q_1^{(n)}} & c_1\Delta & & & & \\ y_2\Delta^{1+q_2^{(n)}} & b_2\Delta^2 & c_2\Delta^2 & & & \\ y_3\Delta^{1+q_3^{(n)}} & a_3\Delta^2 & b_3\Delta^2 & c_3\Delta^2 & & \\ \vdots & & \ddots & \ddots & \ddots & \\ y_{n-1}\Delta^{1+q_{n-1}^{(n)}} & & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\ y_n\Delta^{1+q_n^{(n)}} & & & & a_n\Delta & b_n\Delta \end{vmatrix},$$

$$t_1^{(n)}\Delta \preceq \hat{t}_1^{(n)} = \begin{vmatrix} b_1\Delta & c_1\Delta & & & & y_1\Delta^{1+q_n^{(n)}} \\ a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & & y_2\Delta^{1+q_{n-1}^{(n)}} \\ & \ddots & \ddots & \ddots & & \vdots \\ & & a_{n-2}\Delta^2 & b_{n-2}\Delta^2 & c_{n-2}\Delta^2 & y_{n-2}\Delta^{1+q_3^{(n)}} \\ & & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & y_{n-1}\Delta^{1+q_2^{(n)}} \\ & & & & a_n\Delta & y_n\Delta^{1+q_1^{(n)}} \end{vmatrix},$$

and

$$x_i = \frac{\begin{vmatrix} d_{i-1} & t_1^{(i-1)}\Delta^3 \\ g_i & u_i^{(n)}\Delta^3 \end{vmatrix}}{d_n} = \frac{\xi_i}{d_n},$$

where

$\xi_i \preceq \hat{\xi}_i =$

$$\begin{vmatrix}
b_1\Delta & c_1\Delta & & & y_1\Delta^{3+q_{i-1}^{(i-1)}} & & & & \\
a_2\Delta^2 & b_2\Delta^2 & c_2\Delta^2 & & y_2\Delta^{3+q_{i-2}^{(i-1)}} & & & & \\
& \ddots & \ddots & & \vdots & & & & \\
& & a_{i-1}\Delta & b_{i-1}\Delta & y_{i-1}\Delta^{3+q_1^{(i-1)}} & & & & \\
& & & a_i\Delta & y_i\Delta^{3+q_1^{(n-i+1)}} & c_i\Delta & & & \\
& & & & y_{i+1}\Delta^{3+q_2^{(n-i+1)}} & b_{i+1}\Delta^2 & c_{i+1}\Delta^2 & & \\
& & & & \vdots & & \ddots & \ddots & \\
& & & & y_{n-1}\Delta^{3+q_{n-i}^{(n-i+1)}} & & a_{n-1}\Delta^2 & b_{n-1}\Delta^2 & c_{n-1}\Delta^2 \\
& & & & y_n\Delta^{3+q_{n-i+1}^{(n-i+1)}} & & & a_n\Delta & b_n\Delta
\end{vmatrix}.$$

5. Concluding remarks. From Theorems 3.4 and 4.5 we see that although the computed solution component x_i cannot be expressed as the exact answer of a neighboring system of the original matrix A and the right-hand side vector y, nevertheless the computed solution component is of the form $\frac{p}{q}$ where p and q are sums of products of the perturbed input data. Furthermore, if the exact solution component x_i is expressed as

$$x_i = \frac{\sum_{j=1}^{\lambda(D_{1:n}(i))} p_j}{\sum_{k=1}^{\lambda(D_{1:n})} q_k},$$

then

$$x_i|_{Sw1} = \frac{\sum_{j=1}^{\lambda(D_{1:n}(i))} \bar{p}_j}{\sum_{k=1}^{\lambda(D_{1:n})} \bar{q}_k}, \qquad x_i|_{Sw2} = \frac{\sum_{j=1}^{\lambda(D_{1:n}(i))} \hat{p}_j}{\sum_{k=1}^{\lambda(D_{1:n})} \hat{q}_k},$$

where

$$\sum_{k=1}^{\lambda(D_{1:n})} \bar{q}_k = d_n|_{Sw1}, \qquad \sum_{j=1}^{\lambda(D_{1:n}(i))} \bar{p}_j = \begin{cases} \hat{u}_1|_{Sw1} \text{ if } i = 1, \\ \xi_i|_{Sw1} \text{ for } 2 \le i \le n-1, \\ \hat{t}_n|_{Sw1} \text{ if } i = n, \end{cases}$$

$$\sum_{k=1}^{\lambda(D_{1:n})} \hat{q}_k \preceq \hat{d}_n|_{Sw2}, \qquad \sum_{j=1}^{\lambda(D_{1:n}(i))} \hat{p}_j \preceq \begin{cases} \hat{u}_1|_{Sw2}^{(n)} \text{ if } i = 1, \\ \hat{\xi}_i|_{Sw2} \text{ for } 2 \le i \le n-1, \\ \hat{t}_1|_{Sw2}^{(n)} \text{ if } i = n. \end{cases}$$

By using Theorems 3.4, 4.5, and (4.1), one can easily bound the relative error of p_j or q_k as follows:

$$\frac{|\bar{p}_j - p_j|}{|p_j|} \le |\Delta^{2n+1} - 1| \le (2n+1)u + O(u^2),$$

$$\frac{|\hat{p}_j - p_j|}{|p_j|} \leq |\Delta^{2n+q_1^{(n)}} - 1| \leq (2n + \lceil \log_2 n \rceil)u + O(u^2),$$

$$\frac{|\bar{q}_k - q_k|}{|q_k|} \leq |\Delta^{2(n-1)} - 1| \leq 2(n-1)u + O(u^2),$$

$$\frac{|\hat{q}_k - q_k|}{|q_k|} \leq |\Delta^{2(n-1)} - 1| \leq 2(n-1)u + O(u^2).$$

The above first-order bounds can thus be used to assess the relative importance of errors caused by the computation versus those caused by the inherent data error.

Acknowledgment. The author wishes to thank the anonymous referee for giving numerous suggestions that greatly helped to improve the presentation of this paper.

REFERENCES

[1] P. N. SWARZTRAUBER, *A parallel algorithm for solving general tridiagonal equations*, Math. Comp., 33 (1979), pp. 185–199.

[2] H. S. STONE, *Parallel tridiagonal equation solvers*, ACM Trans. Math. Software, 1 (1975), pp. 289–307.

[3] J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1963.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 715-728, July, 1994

SENSITIVITY OF THE STATIONARY DISTRIBUTION OF A MARKOV CHAIN*

CARL D. MEYER†

Abstract. It is well known that if the transition matrix of an irreducible Markov chain of moderate size has a subdominant eigenvalue which is close to 1, then the chain is ill conditioned in the sense that there are stationary probabilities which are sensitive to perturbations in the transition probabilities. However, the converse of this statement has heretofore been unresolved. The purpose of this article is to address this issue by establishing upper and lower bounds on the condition number of the chain such that the bounding terms are functions of the eigenvalues of the transition matrix. Furthermore, it is demonstrated how to obtain estimates for the condition number of an irreducible chain with little or no extra computational effort over that required to compute the stationary probabilities by means of an LU or QR factorization.

Key words. Markov chains, stationary distribution, stochastic matrix, sensitivity analysis, perturbation theory, character of a Markov chain, condition numbers

AMS subject classifications. 65U05, 65F35, 60J10, 60J20, 15A51, 15A12, 15A18

1. Introduction.

The problem under consideration is that of analyzing the effects of small perturbations to the transition probabilities of a finite, irreducible, homogeneous Markov chain. More precisely, if $\mathbf{P}_{n\times n}$ is the transition probability matrix for such a chain, and if $\boldsymbol{\pi}^T = (\pi_1, \pi_2, \ldots, \pi_n)$ is the stationary distribution vector satisfying $\boldsymbol{\pi}^T\mathbf{P} = \boldsymbol{\pi}^T$ and $\sum_{i=1}^n \pi_i = 1$, the goal is to describe the effect on $\boldsymbol{\pi}^T$ when $\mathbf{P}$ is perturbed by a matrix $\mathbf{E}$ such that $\tilde{\mathbf{P}} = \mathbf{P} + \mathbf{E}$ is the transition probability matrix of another irreducible Markov chain.

Schweitzer (1968) provided the first perturbation analysis in terms of Kemeny and Snell's "fundamental matrix" $\mathbf{Z} = (\mathbf{A} + \mathbf{e}\boldsymbol{\pi}^T)^{-1}$ in which $\mathbf{A} = \mathbf{I} - \mathbf{P}$ and $\mathbf{e}$ is a column of 1's. If $\mathbf{A}^{\#}$ denotes the group inverse of $\mathbf{A}$ [Meyer (1975) or Campbell and Meyer (1991)], then

$$\mathbf{Z} = (\mathbf{A} + \mathbf{e}\boldsymbol{\pi}^T)^{-1} = \mathbf{A}^{\#} + \mathbf{e}\boldsymbol{\pi}^T.$$

But in virtually all applications involving $\mathbf{Z}$, the term $\mathbf{e}\boldsymbol{\pi}^T$ is redundant; i.e., all relevant information is contained in $\mathbf{A}^{\#}$. In particular, if $\tilde{\boldsymbol{\pi}}^T = (\tilde{\pi}_1, \tilde{\pi}_2, \ldots, \tilde{\pi}_n)$ is the stationary distribution for $\tilde{\mathbf{P}} = \mathbf{P} + \mathbf{E}$, then

$$\tilde{\boldsymbol{\pi}}^T = \boldsymbol{\pi}^T \left(\mathbf{I} + \mathbf{E}\mathbf{A}^{\#}\right)^{-1} \tag{1.1}$$

and

$$\left\|\boldsymbol{\pi}^T - \tilde{\boldsymbol{\pi}}^T\right\| \leq \|\mathbf{E}\| \left\|\mathbf{A}^{\#}\right\| \tag{1.2}$$

in which $\|\star\|$ can be either the 1-, 2-, or ∞-norm. If the jth column and the (i,j)-entry of $\mathbf{A}^{\#}$ are denoted by $\mathbf{A}^{\#}_{*j}$ and $a^{\#}_{ij}$, respectively, then

$$|\pi_j - \tilde{\pi}_j| \leq \|\mathbf{E}\| \left\|\mathbf{A}^{\#}_{*j}\right\| \tag{1.3}$$

*Received by the editors April 6, 1992; accepted for publication (in revised form) October 30, 1992. This work was supported in part by National Science Foundation grants DMS-9020915 and DDM-8906248.

†North Carolina State University, Mathematics Department, Raleigh, North Carolina 27695-8205 (meyer@math.ncsu.edu).

and

$$\max_j |\pi_j - \tilde{\pi}_j| \leq \|\mathbf{E}\|_\infty \max_{i,j} |a_{ij}^{\#}|. \tag{1.4}$$

This bound is about as good as possible—see Ipsen and Meyer (1994) for a discussion of optimal bounds. Moreover, if the transition probabilities are analytic functions of a parameter t so that $\mathbf{P} = \mathbf{P}(t)$, then

$$\frac{d\boldsymbol{\pi}^T}{dt} = \boldsymbol{\pi}^T \frac{d\mathbf{P}}{dt}\mathbf{A}^{\#} \quad \text{and} \quad \frac{d\pi_j}{dt} = \boldsymbol{\pi}^T \frac{d\mathbf{P}}{dt}\mathbf{A}^{\#}_{*j}. \tag{1.5}$$

The results (1.1) and (1.2) are due to Meyer (1980), and (1.3) appears in Golub and Meyer (1986). The inequality (1.4) was given by Funderlic and Meyer (1986), and the formulas (1.5) are derived in Golub and Meyer (1986) and Meyer and Stewart (1988). Seneta (1991) established an inequality similar to (1.2) using the coefficient of ergodicity $\tau_1(\mathbf{A}^{\#})$ in place of $\left\|\mathbf{A}^{\#}\right\|$.

These facts make it absolutely clear that the entries in $\mathbf{A}^{\#}$ determine the extent to which $\boldsymbol{\pi}^T$ is sensitive to small changes in $\mathbf{P}$, so, on the basis of (1.4), it is natural to adopt the following definition of Funderlic and Meyer (1986).

DEFINITION 1.1. The *condition* of a Markov chain with a transition matrix $\mathbf{P}$ is measured by the size of its *condition number,* which is defined to be

$$\kappa = \max_{i,j} |a_{ij}^{\#}|$$

where $a_{ij}^{\#}$ is the (i,j)-entry in the group inverse $\mathbf{A}^{\#}$ of $\mathbf{A} = \mathbf{I} - \mathbf{P}$. It is an elementary fact that κ is invariant under permutations of the states of the chain.

For chains of moderate size, it is not difficult to show (see the proof of Theorem 2.1 given in §4) that if there exists a subdominant eigenvalue of $\mathbf{P}$ which is close to 1, then κ must be large. However, the converse of this statement has heretofore been unresolved, and our purpose is to focus on this issue. More precisely, we address the following question.

> *If the subdominant eigenvalues of an irreducible Markov chain are well separated from* 1, *can we be sure that the chain is well conditioned? In other words, do the subdominant eigenvalues of* $\mathbf{P}$ (*or equivalently, the nonzero eigenvalues of* $\mathbf{A}$) *somehow provide complete information about the sensitivity of the chain—or do we really need to know something about the singular values of* $\mathbf{A}$?

The conjecture that $\kappa = \max_{i,j} |a_{ij}^{\#}|$ is somehow controlled by the nonzero eigenvalues of $\mathbf{A}$ is contrary to what is generally true—a standard example is the triangular matrix

(1.6)

$$\mathbf{T}_{n\times n} = \begin{pmatrix} 1 & -2 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \mathbf{T}^{-1} = \begin{pmatrix} 1 & 2 & 4 & \cdots & 2^{n-2} & 2^{n-1} \\ 0 & 1 & 2 & \cdots & 2^{n-3} & 2^{n-2} \\ 0 & 0 & 1 & \ddots & 2^{n-4} & 2^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 2 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

for which $\max_{i,j}[\mathbf{T}^{-1}]_{ij}$ is immense for even moderate values of n, but the eigenvalues of $\mathbf{T}$ provide no clue whatsoever that this occurs. The fact that the eigenvalues are repeated or that $\mathbf{T}$ is nonsingular is irrelevant—consider a small perturbation of $\mathbf{T}$ or the matrices

$$\tilde{\mathbf{T}} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{T} \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{T}}^{\#} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{T}^{-1} \end{pmatrix}.$$

We will prove that, unlike the situation illustrated above, irreducible stochastic matrices $\mathbf{P}$ possess enough structure to guarantee that growth of the entries in $\mathbf{A}^{\#}$ is controlled by the nonzero eigenvalues of $\mathbf{A} = \mathbf{I} - \mathbf{P}$. As a consequence, it will follow that the sensitivity of an irreducible Markov chain is governed by the location of its subdominant eigenvalues.

2. The main result. In the sequel, it is convenient to adopt the following terminology and notation.

DEFINITION 2.1. Let $\mathbf{P}$ be the transition probability matrix of an n-state irreducible Markov chain, and let

$$\sigma(\mathbf{P}) = \{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$$

denote the eigenvalues of $\mathbf{P}$. The *character*[1] of the chain is defined to be the (necessarily real) number

$$\chi = (1 - \lambda_2)(1 - \lambda_3)\cdots(1 - \lambda_n).$$

It will follow from later developments that

$$0 < \chi \leq n. \tag{2.1}$$

A chain is said to be of "weak character" when χ is close to 0, and the chain is said to have a "strong character" when χ is significantly larger than 0.

If

$$\mathbf{P} = \mathbf{T}^{-1}\begin{pmatrix} 1 & 0 \\ 0 & \mathbf{C} \end{pmatrix}\mathbf{T}$$

(e.g., this may be the reduction to Jordan form) where the spectral radius of $\mathbf{C}$ is less than 1, then

$$\mathbf{A} = \mathbf{T}^{-1}\begin{pmatrix} 0 & 0 \\ 0 & \mathbf{I} - \mathbf{C} \end{pmatrix}\mathbf{T} \quad \text{and} \quad \mathbf{A}^{\#} = \mathbf{T}^{-1}\begin{pmatrix} 0 & 0 \\ 0 & (\mathbf{I} - \mathbf{C})^{-1} \end{pmatrix}\mathbf{T}$$

[Campbell and Meyer (1991)], so

$$\chi = \det(\mathbf{I} - \mathbf{C}) \quad \text{and} \quad \chi^{-1} = \det(\mathbf{I} - \mathbf{C})^{-1}.$$

In other words, χ and χ^{-1} are the respective determinants of the nonsingular parts of $\mathbf{A}$ and $\mathbf{A}^{\#}$ in the sense that

$$\chi = \det\left(\mathbf{A}_{/\mathcal{R}(\mathbf{A})}\right) \quad \text{and} \quad \chi^{-1} = \det\left(\mathbf{A}^{\#}_{/\mathcal{R}(\mathbf{A})}\right)$$

where $\mathbf{A}_{/\mathcal{R}(\mathbf{A})}$ denotes the linear operator defined by restricting $\mathbf{A}$ to $R(\mathbf{A})$. It is also true that $\chi^{-1} = \det(\mathbf{Z})$ where $\mathbf{Z}$ is Kemeny and Snell's "fundamental matrix."

The main result of this paper is the following theorem which establishes the connection between the condition of an irreducible chain and its character.

---

[1] The character was defined by Meyer (1993) to be $\chi = n^{-1}(1 - \lambda_2)(1 - \lambda_3)\cdots(1 - \lambda_n)$, which is the normalization of the definition given here.

THEOREM 2.1. *For an irreducible stochastic matrix* $\mathbf{P}_{n\times n}$, *let* $\mathbf{A} = \mathbf{I} - \mathbf{P}$, *and for* $i \neq j$, *let* $\delta_{ij}(\mathbf{A})$ *denote the deleted product of diagonal entries*

$$\delta_{ij}(\mathbf{A}) = \prod_{k\neq i,j} a_{kk} = \prod_{k\neq i,j} (1 - p_{kk}).$$

If $\delta = \max_{i,j} \delta_{ij}(\mathbf{A})$ (*the product of all but the two smallest diagonal entries*), *then the condition number* κ *is bounded by*

$$\frac{1}{n \min\limits_{\lambda_i \neq 1} |1 - \lambda_i|} \leq \kappa < \frac{2\delta(n-1)}{\chi} \leq \frac{2(n-1)}{\chi}. \tag{2.2}$$

The proof of this theorem depends on exploiting the rich structure of $\mathbf{A}$, some of which is apparent, and some of which requires illumination. Before giving a formal argument, it is necessary to detail the various components of this structure, so the important facets are first laid out in §3 as a sequence of lemmas. After the necessary framework is in place, it will be a simple matter to connect the lemmas together in order to construct a proof; this is contained in §4.

By combining Theorem 2.1 with (1.4) and the other facts listed in §1, we arrive at the following conclusion.

THEOREM 2.2. *The condition of an irreducible Markov chain is primarily governed by how close the subdominant eigenvalues of the chain are to* 1. *More precisely, if an irreducible chain is well conditioned, then all subdominant eigenvalues must be well separated from* 1, *and if all subdominant eigenvalues are well separated from* 1 *in the sense that the chain has a strong character, then it must be well conditioned.*

It is a corollary of Theorem 2.1 that if $\max_{\lambda_i\neq 1} |\lambda_i| << 1$, then the chain is not overly sensitive, but it is important to underscore the point that the issue of sensitivity is not equivalent to the question of how close $\max_{\lambda_i\neq 1} |\lambda_i|$ is to 1. Knowing that some $|\lambda_i| \approx 1$ is not sufficient to guarantee that the chain is sensitive; e.g., consider the well-conditioned periodic chain (or any small perturbation thereof) for which

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{A}^{\#} = \frac{1}{3}\begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}.$$

3. The underlying structure. The purpose of this section is to organize relevant properties of $\mathbf{A} = \mathbf{I} - \mathbf{P}$ into a sequence of lemmas from which the formal proof of Theorem 2.1 can be constructed. Some of the more transparent or well-known features of $\mathbf{A}$ are stated in the first lemma.

LEMMA 3.1. *If* $\mathbf{A} = \mathbf{I} - \mathbf{P}$ *where* $\mathbf{P}_{n\times n}$ *is an irreducible stochastic matrix, then the following statements are true.*

(3.1) $\mathbf{A}$ *as well as each principal submatrix of* $\mathbf{A}$ *has strictly positive diagonal entries, and the off-diagonal entries are nonpositive.*

(3.2) $\mathbf{A}$ *is a singular M-matrix of rank* $n - 1$.

(3.3) *If* $\mathbf{B}_{k\times k}$ $(k < n)$ *is a principal submatrix of* $\mathbf{A}$, *then each of the following statements is true.*

- (a) $\mathbf{B}$ *is a nonsingular M-matrix.*
- (b) $\mathbf{B}^{-1} \geq 0$.
- (c) $\det(\mathbf{B}) > 0$.
- (d) $\mathbf{B}$ *is diagonally dominant.*
- (e) $\det(\mathbf{B}) \leq b_{11}b_{22}\cdots b_{kk} \leq 1$.

Proof. These facts are either self-evident, or they are direct consequences of well-known results—see Berman and Plemmons (1979) or Horn and Johnson (1991). □

Part of the less transparent structure of $\mathbf{A}$ is illuminated in the following sequence of lemmas.

LEMMA 3.2. *If $\mathbf{P}_{n\times n}$ is an irreducible stochastic matrix, and if $\mathbf{A}_i$ denotes the principal submatrix of $\mathbf{A} = \mathbf{I} - \mathbf{P}$ obtained by deleting the ith row and column from $\mathbf{A}$, then*

$$\chi = \sum_{i=1}^{n} \det(\mathbf{A}_i).$$

Proof. Suppose that the eigenvalues of $\mathbf{A}$ are denoted by $\{\mu_1, \mu_2, \ldots, \mu_n\}$, and write the characteristic equation for $\mathbf{A}$ as

$$x^n + \alpha_{n-1}x^{n-1} + \cdots + \alpha_1 x + \alpha_0 = 0.$$

Each coefficient α_{n-k} is given by $(-1)^k$ times the sum of the product of the eigenvalues of $\mathbf{A}$ taken k at a time. That is,

$$\alpha_{n-k} = (-1)^k \sum_{1\le i_1<\cdots<i_k\le n} \mu_{i_1}\mu_{i_2}\cdots\mu_{i_k}. \tag{3.4}$$

But it is also a standard result from elementary matrix theory that each coefficient α_{n-k} can be described as

$$\alpha_{n-k} = (-1)^k \sum(\text{all } k\times k \text{ principal minors of } \mathbf{A}).$$

Since 0 is a simple eigenvalue for $\mathbf{A}$, there is only one nonzero term in the sum (3.4) when $k = n-1$, and hence

$$\begin{aligned}\alpha_1 &= (-1)^{n-1}\mu_2\mu_3\cdots\mu_n = (-1)^{n-1}(1-\lambda_2)(1-\lambda_3)\cdots(1-\lambda_n)\\ &= (-1)^{n-1}\sum_{i=1}^{n}\det(\mathbf{A}_i).\end{aligned}$$

Therefore,

$$\sum_{i=1}^{n}\det(\mathbf{A}_i) = \prod_{k=2}^{n}(1-\lambda_k) = \chi. \qquad \square$$

LEMMA 3.3. *If $\mathbf{A}_i$ denotes the principal submatrix of $\mathbf{A} = \mathbf{I} - \mathbf{P}$ obtained by deleting the ith row and column from $\mathbf{A}$, and if π_i is the ith stationary probability, then the character of the chain is given by*

$$\chi = \frac{\det(\mathbf{A}_i)}{\pi_i}.$$

Proof. This result follows directly from Lemma 3.2 and the fact that the stationary distribution $\boldsymbol{\pi}^T$ is given by the formula

$$\boldsymbol{\pi}^T = \frac{1}{\sum_{i=1}^{n}\det(\mathbf{A}_i)}\Big(\det(\mathbf{A}_1), \det(\mathbf{A}_2), \ldots, \det(\mathbf{A}_n)\Big)$$

[Golub and Meyer (1986) or Iosifescu (1980), p. 123]. □

The mean return time for the kth state is $R_k = 1/\pi_k$ [Kemeny and Snell (1960)], and, since not all of the π_k's can be less than $1/n$, there must exist a state such that $R_k \le n$. By combining this with (3.3c) and (3.3e), an interesting corollary—which proves (2.1)—is produced.

COROLLARY 3.1. *If R_k denotes the mean return time for the kth state then*

$$0 < \det(\mathbf{A}_i) < \chi \leq \min_k R_k \leq n \quad \textit{for each } i = 1, 2, \ldots, n.$$

LEMMA 3.4. *If $\mathbf{A} = \mathbf{I} - \mathbf{P}$ where $\mathbf{P}_{n\times n}$ is an irreducible stochastic matrix, and if $\mathbf{B}_{k\times k}$ $(k < n)$ is a principal submatrix of $\mathbf{A}$, then the largest entry in each column of $\mathbf{B}^{-1}$ is the diagonal entry. That is, for $j = 1, 2, \ldots, k$, it must be the case that*

$$[\mathbf{B}^{-1}]_{jj} \geq [\mathbf{B}^{-1}]_{ij} \quad \textit{for each } i \neq j.$$

At least two different proofs are possible, and we shall give both because each is instructive in its own right. The first argument is shorter and more probabilistic, but it rests on a result which requires a proof of its own. The second argument involves more algebraic details, but it is entirely self-contained and depends only on elementary concepts.

Probabilistic proof. Without loss of generality, assume that $\mathbf{B}$ is the leading $k \times k$ principal submatrix of $\mathbf{A}$ so that $\mathbf{P}$ has the form

$$\mathbf{P} = \begin{pmatrix} \mathbf{I} - \mathbf{B} & \star \\ \star & \star \end{pmatrix}.$$

Consider any pair of states i and j in the set $\mathcal{S} = \{1, 2, \ldots, k\}$, and let N_j denote the number of times the process is in state j before first hitting a state in the complement $\overline{\mathcal{S}} = \{k+1, k+2, \ldots, n\}$. If X_n denotes the state of the process after n steps, and if

$$h_{ij} = P(\text{hitting state } j \text{ before entering } \overline{\mathcal{S}} \mid X_0 = i),$$

then

$$E[N_j \mid X_0 = i] = d_{ij} + h_{ij} E[N_j \mid X_0 = j] \quad \text{where} \quad d_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \tag{3.5}$$

This statement (which appears without proof on p. 62 in Kemeny and Snell (1960)) is intuitive, but it is not trivial. The theory of absorbing chains says that

$$[\mathbf{B}^{-1}]_{ij} = E[N_j \mid X_0 = i],$$

so for $i \neq j$ we have $[\mathbf{B}^{-1}]_{ij} = h_{ij}[\mathbf{B}^{-1}]_{jj} \leq [\mathbf{B}^{-1}]_{jj}$. □

Algebraic proof. Assume that $\mathbf{B}$ is the leading $k \times k$ principal submatrix of $\mathbf{A}$, and suppose the states have been arranged so that the jth state is listed first and the ith state is listed second. The goal is to prove that $[\mathbf{B}^{-1}]_{11} \geq [\mathbf{B}^{-1}]_{21}$. Because

$$[\mathbf{B}^{-1}]_{11} = \frac{\det(\mathbf{B}_{11})}{\det(\mathbf{B})} \quad \text{and} \quad [\mathbf{B}^{-1}]_{21} = \frac{-\det(\mathbf{B}_{12})}{\det(\mathbf{B})}$$

where $\mathbf{B}_{ij}$ denotes the submatrix of $\mathbf{B}$ obtained by deleting the ith row and jth column from $\mathbf{B}$ and because Lemma 3.1 guarantees that $\det(\mathbf{B}) > 0$, it suffices to prove that

$$\det(\mathbf{B}_{11}) + \det(\mathbf{B}_{12}) \geq 0.$$

Denote the first unit vector by $\mathbf{e}_1^T = (1, 0, \ldots, 0)$, and partition $\mathbf{B}$ as

(3.6)
$$\mathbf{B} = \begin{pmatrix} 1-p_{11} & -p_{12} & \cdots & -p_{1k} \\ -p_{21} & 1-p_{22} & \cdots & -p_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ -p_{k1} & -p_{k2} & \cdots & 1-p_{kk} \end{pmatrix} = \left(\begin{array}{c|c|c|c} 1-p_{11} & -p_{12} & \cdots & -p_{1k} \\ \hline \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_k \end{array}\right).$$

In terms of these quantities, $\det(\mathbf{B}_{11}) + \det(\mathbf{B}_{12})$ is given by

$$\begin{aligned} \det(\mathbf{B}_{11}) + \det(\mathbf{B}_{12}) &= \det(\mathbf{b}_2 \,|\, \mathbf{b}_3 \,|\, \cdots \,|\, \mathbf{b}_k) + \det(\mathbf{b}_1 \,|\, \mathbf{b}_3 \,|\, \cdots \,|\, \mathbf{b}_k) \\ &= \det(\mathbf{b}_2 + \mathbf{b}_1 \,|\, \mathbf{b}_3 \,|\, \cdots \,|\, \mathbf{b}_k) \\ &= \det(\mathbf{B}_{11} + \mathbf{b}_1 \mathbf{e}_1^T) \\ &= \det(\mathbf{B}_{11})\,(1 + \mathbf{e}_1^T \mathbf{B}_{11}^{-1} \mathbf{b}_1). \end{aligned}$$

Lemma 3.1 also insures that $\det(\mathbf{B}_{11}) > 0$, so the proof can be completed by arguing that

$$1 + \mathbf{e}_1^T \mathbf{B}_{11}^{-1} \mathbf{b}_1 \geq 0.$$

To do so, modify the chain by making state 1 as well as states $k+1, k+2, \ldots, n$ absorbing states so that the transition matrix has the form

$$\tilde{\mathbf{P}} = \left(\begin{array}{c|ccc|c|ccc} 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \hline p_{21} & p_{22} & p_{23} & \cdots & p_{2k} & p_{2,k+1} & \cdots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3k} & p_{3,k+1} & \cdots & p_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ p_{k1} & p_{k2} & p_{k3} & \cdots & p_{kk} & p_{k,k+1} & \cdots & p_{kn} \\ \hline 0 & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{array}\right)$$

$$= \left(\begin{array}{c|c|c} 1 & \mathbf{0} & \mathbf{0} \\ \hline -\mathbf{b}_1 & \mathbf{Q} & \mathbf{R} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I}_{n-k} \end{array}\right).$$

It follows from the elementary theory of absorbing chains that the entries in the matrix

$$(\mathbf{I} - \mathbf{Q})^{-1}(-\mathbf{b}_1 \,|\, \mathbf{R}) = \mathbf{B}_{11}^{-1}(-\mathbf{b}_1 \,|\, \mathbf{R})$$

represent the various absorption probabilities, and consequently all entries in $-\mathbf{B}_{11}^{-1}\mathbf{b}_1$ are between 0 and 1 so that

$$0 \leq 1 + \mathbf{e}_1^T \mathbf{B}_{11}^{-1} \mathbf{b}_1 \leq 1. \qquad \square$$

Note. Although it may not be of optimal efficiency, the algebraic argument given above is also a proof of the statement (3.5).

LEMMA 3.5. *If* $\mathbf{A} = \mathbf{I} - \mathbf{P}$ *where* $\mathbf{P}_{n\times n}$ *is an irreducible stochastic matrix, and if* $\mathbf{B}_{k\times k}$ $(k < n)$ *is a principal submatrix of* $\mathbf{A}$, *then*

$$0 < \det(\mathbf{B}) \leq \frac{\max_i \delta_i(\mathbf{B})}{\max_{i,j} [\mathbf{B}^{-1}]_{ij}} \leq \frac{1}{\max_{i,j} [\mathbf{B}^{-1}]_{ij}}$$

where $\delta_r(\mathbf{B})$ *denotes the deleted product* $\delta_r(\mathbf{B}) = b_{11}b_{22}\cdots b_{kk}/b_{rr}$.

Proof. Lemma 3.4 insures that there is some diagonal entry $[\mathbf{B}^{-1}]_{rr}$ of $\mathbf{B}^{-1}$ such that

$$[\mathbf{B}^{-1}]_{rr} = \max_{i,j} [\mathbf{B}^{-1}]_{ij}. \tag{3.7}$$

If $\mathbf{B}_{rr}$ is the principal submatrix of $\mathbf{B}$ obtained by deleting the rth row and column from $\mathbf{B}$, then (3.3e) together with (3.7) produces

$$\begin{aligned}\det(\mathbf{B}) = \frac{\det(\mathbf{B}_{rr})}{[\mathbf{B}^{-1}]_{rr}} &\leq \frac{\delta_r(\mathbf{B})}{[\mathbf{B}^{-1}]_{rr}} = \frac{\delta_r(\mathbf{B})}{\max_{i,j}[\mathbf{B}^{-1}]_{ij}} \\ &\leq \frac{\max_i \delta_i(\mathbf{B})}{\max_{i,j}[\mathbf{B}^{-1}]_{ij}} \leq \frac{1}{\max_{i,j}[\mathbf{B}^{-1}]_{ij}}. \qquad \square\end{aligned}$$

LEMMA 3.6. *For an irreducible stochastic matrix* $\mathbf{P}_{n\times n}$, *let* $\mathbf{A}_j$ *be the principal submatrix of* $\mathbf{A} = \mathbf{I} - \mathbf{P}$ *obtained by deleting the* j*th row and column from* $\mathbf{A}$, *and let* $\mathbf{Q}$ *be the permutation matrix such that*

$$\mathbf{Q}^T\mathbf{A}\mathbf{Q} = \begin{pmatrix} \mathbf{A}_j & \mathbf{c}_j \\ \mathbf{d}_j^T & a_{jj} \end{pmatrix}.$$

If the stationary distribution for $\mathbf{Q}^T\mathbf{P}\mathbf{Q}$ *is written as* $\boldsymbol{\psi}^T = \boldsymbol{\pi}^T\mathbf{Q} = (\overline{\boldsymbol{\pi}}^T, \pi_j)$, *then the group inverse of* $\mathbf{A}$ *is given by*

$$\mathbf{A}^{\#} = \mathbf{Q}\begin{pmatrix} (\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T)\mathbf{A}_j^{-1}(\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T) & -\pi_j(\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T)\mathbf{A}_j^{-1}\mathbf{e} \\ -\overline{\boldsymbol{\pi}}^T\mathbf{A}_j^{-1}(\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T) & \pi_j\,\overline{\boldsymbol{\pi}}^T\mathbf{A}_j^{-1}\mathbf{e} \end{pmatrix}\mathbf{Q}^T$$

where $\mathbf{e}$ *is a column of* 1*'s whose size is determined by the context in which it appears.*

Proof. The group inverse possesses the property that $(\mathbf{T}^{-1}\mathbf{A}\mathbf{T})^{\#} = \mathbf{T}^{-1}\mathbf{A}^{\#}\mathbf{T}$ for all nonsingular matrices $\mathbf{T}$ [Campbell and Meyer (1991)], so

$$\mathbf{A}^{\#} = \mathbf{Q}\begin{pmatrix} \mathbf{A}_j & \mathbf{c}_j \\ \mathbf{d}_j^T & a_{jj} \end{pmatrix}^{\#}\mathbf{Q}^T.$$

Since $\operatorname{rank}(\mathbf{Q}^T\mathbf{A}\mathbf{Q}) = n - 1$, it follows that $a_{jj} - \mathbf{d}_j^T\mathbf{A}_j^{-1}\mathbf{c}_j = 0$, and this is used to verify that

$$\begin{aligned}\begin{pmatrix} \mathbf{A}_j & \mathbf{c}_j \\ \mathbf{d}_j^T & a_{jj} \end{pmatrix}^{\#} &= (\mathbf{I} - \mathbf{e}\boldsymbol{\psi}^T)\begin{pmatrix} \mathbf{A}_j^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{pmatrix}(\mathbf{I} - \mathbf{e}\boldsymbol{\psi}^T) \\ &= \begin{pmatrix} (\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T)\mathbf{A}_j^{-1}(\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T) & -\pi_j(\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T)\mathbf{A}_j^{-1}\mathbf{e} \\ -\overline{\boldsymbol{\pi}}^T\mathbf{A}_j^{-1}(\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T) & \pi_j\,\overline{\boldsymbol{\pi}}^T\mathbf{A}_j^{-1}\mathbf{e} \end{pmatrix}. \qquad \square\end{aligned}$$

4. Proof of the main theorem. The preceding sequence of lemmas are now connected together to prove the primary results stated in Theorem 2.1.

The upper bound. To derive the inequalities

$$\max_{i,j} |a_{ij}^{\#}| < \frac{2\delta(n-1)}{\chi} \leq \frac{2(n-1)}{\chi}, \tag{4.1}$$

begin by letting $\mathbf{Q}$ be the permutation matrix given in Lemma 3.6 so that for $i \neq j$, the (i,j)-entry of $\mathbf{A}^{\#}$ is the (k,n)-entry of $\mathbf{Q}^T\mathbf{A}^{\#}\mathbf{Q}$ where $k \neq n$. In succession, use the formula of Lemma 3.6 and Hölder's inequality followed by the results of Lemmas 3.5 and 3.3 to write

$$\begin{aligned}
|a_{ij}^{\#}| &= \pi_j |\mathbf{e}_k^T(\mathbf{I} - \mathbf{e}\overline{\boldsymbol{\pi}}^T)\mathbf{A}_j^{-1}\mathbf{e}| \leq \pi_j \|\mathbf{e}_k - \overline{\boldsymbol{\pi}}\|_1 \left\|\mathbf{A}_j^{-1}\mathbf{e}\right\|_\infty \\
&< 2\pi_j \left\|\mathbf{A}_j^{-1}\right\|_\infty \leq 2\pi_j(n-1)\max_{r,s}\left[\mathbf{A}_j^{-1}\right]_{rs} \\
&\leq \frac{2\pi_j(n-1)\max_i \delta_i(\mathbf{A}_j)}{\det(\mathbf{A}_j)} \leq \frac{2\pi_j(n-1)\delta}{\det(\mathbf{A}_j)} \\
&= \frac{2\delta(n-1)}{\chi} \leq \frac{2(n-1)}{\chi}.
\end{aligned}$$

Now consider the diagonal elements. The (j,j)-entry of $\mathbf{A}^{\#}$ is the (n,n)-entry of $\mathbf{Q}^T\mathbf{A}^{\#}\mathbf{Q}$, so proceeding in a manner similar to that above produces

$$\begin{aligned}
|a_{jj}^{\#}| &= \pi_j |\overline{\boldsymbol{\pi}}^T\mathbf{A}_j^{-1}\mathbf{e}| \leq \pi_j \|\overline{\boldsymbol{\pi}}\|_1 \left\|\mathbf{A}_j^{-1}\mathbf{e}\right\|_\infty \\
&< \pi_j \left\|\mathbf{A}_j^{-1}\right\|_\infty \leq \pi_j(n-1)\max_{r,s}\left[\mathbf{A}_j^{-1}\right]_{rs} \\
&\leq \frac{\pi_j(n-1)\max_i \delta_i(\mathbf{A}_j)}{\det(\mathbf{A}_j)} \leq \frac{\pi_j(n-1)\delta}{\det(\mathbf{A}_j)} \\
&= \frac{\delta(n-1)}{\chi} \leq \frac{(n-1)}{\chi},
\end{aligned}$$

thus proving (4.1).

The lower bound. To establish that

$$\frac{1}{n\min_{\lambda_i \neq 1}|1-\lambda_i|} \leq \max_{i,j}|a_{ij}^{\#}|, \tag{4.2}$$

make use of the fact that if $\mathbf{Ax} = \mu\mathbf{x}$ for $\mu \neq 0$, then $\mathbf{A}^{\#}\mathbf{x} = \mu^{-1}\mathbf{x}$ [Campbell and Meyer (1991), p. 129]. In particular, if $\lambda \neq 1$ is an eigenvalue of $\mathbf{P}$, and if $\mathbf{x}$ is a corresponding eigenvector, then $\mathbf{Ax} = (1-\lambda)\mathbf{x}$ implies that $\mathbf{A}^{\#}\mathbf{x} = (1-\lambda)^{-1}\mathbf{x}$, so

$$\frac{1}{1-\lambda} \leq \left\|\mathbf{A}^{\#}\right\|_\infty \leq n\max_{i,j}|a_{ij}^{\#}|. \qquad \square$$

5. Using an LU factorization. Except for chains which are too large to fit into a computer's main memory, the stationary distribution $\boldsymbol{\pi}^T$ is generally computed by direct methods; i.e., either an LU or QR factorization of $\mathbf{A} = \mathbf{I} - \mathbf{P}$ (or $\mathbf{A}^T$) is computed [Harrod and Plemmons (1984); Grassmann et al. (1985); Funderlic and Meyer (1986); Golub and Meyer (1986); Barlow (1993)]. Even for very large chains which are nearly uncoupled, direct methods are usually involved—they can be the basis of the main algorithm [Stewart and Zhang (1991)], or they can be used to solve the aggregated and coupling chains in iterative aggregation/disaggregation algorithms [Chatelin and Miranker (1982), Haviv (1987)]. In the conclusion of their paper, Golub and Meyer (1986) make the following observation.

> *Computational experience suggests that when a triangular factorization of* $\mathbf{A}_{n\times n}$ *is used to solve an irreducible chain, the condition of the chain seems to be a function of the size of the nonzero pivots, and this means that it should be possible to estimate* κ *with little or no extra cost beyond that incurred in computing* $\boldsymbol{\pi}^T$. *For large chains, this can be a significant savings over the* $O(n^2)$ *operations demanded by traditional condition estimators.*

Of course, this is contrary to the situation which exists for general nonsingular matrices because the absence of small pivots (or the existence of a large determinant) is not a guarantee of a well-conditioned matrix—consider the matrix in (1.6). A mathematical formulation and proof (or even an intuitive explanation) of Golub and Meyer's observation has heretofore not been given, but the results of §2 and §3 now make it possible to give a more precise statement and a rigorous proof of the Golub–Meyer observation. The arguments hinge on the fact that whenever $\boldsymbol{\pi}^T$ is computed by means of a triangular factorization of $\mathbf{A}$ (or $\mathbf{A}^T$), the character of the chain is always an immediate by-product. The results for an LU factorization are given below, and the analogous theory for a QR factorization is given in the next section.

Suppose that the LU factorization [2] of $\mathbf{A} = \mathbf{I} - \mathbf{P}$ is computed to be

$$\mathbf{A} = \mathbf{LU} = \begin{pmatrix} \mathbf{L}_n & \mathbf{0} \\ \mathbf{r}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{U}_n & \mathbf{c} \\ \mathbf{0} & 0 \end{pmatrix}.$$

If $\mathbf{A}_n$ is the principal submatrix of $\mathbf{A}$ obtained by deleting the last row and column from $\mathbf{A}$, then $\mathbf{A}_n$ is a nonsingular M-matrix, and its LU factorization is $\mathbf{A}_n = \mathbf{L}_n\mathbf{U}_n$. Since the LU factors of a nonsingular M-matrix are also nonsingular M-matrices [Berman and Plemmons (1979), Horn and Johnson (1991)], it follows that $\mathbf{L}_n$ and $\mathbf{U}_n$ are nonsingular M-matrices, and hence $\mathbf{L}_n^{-1} \geq \mathbf{0}$ and $\mathbf{U}_n^{-1} \geq \mathbf{0}$. Consequently, $\mathbf{r}^T \leq \mathbf{0}$, so the solution (obtained by a simple substitution process with no divisions) of the nonsingular triangular system $\mathbf{x}^T\mathbf{L}_n = -\mathbf{r}^T$ is nonnegative. This together with the result of Lemma 3.3 and Theorem 2.1 produces the following conclusion.

THEOREM 5.1. *For an irreducible Markov chain whose transition matrix is* $\mathbf{P}$, *let the* LU *factorization of* $\mathbf{A} = \mathbf{I} - \mathbf{P}$ *be given by*

$$\mathbf{A} = \mathbf{LU} = \begin{pmatrix} \mathbf{L}_n & \mathbf{0} \\ \mathbf{r}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{U}_n & \mathbf{c} \\ \mathbf{0} & 0 \end{pmatrix}.$$

---

[2] Regardless of whether $\mathbf{A}$ or $\mathbf{A}^T$ is used, Gaussian elimination with finite-precision arithmetic can prematurely produce a zero (or even a negative) pivot, and this can happen for well-conditioned chains. Practical implementation demands a strategy to deal with this situation, and Funderlic and Meyer (1986) and Stewart and Zhang (1991) discuss this problem along with possible remedies. Practical algorithms involve reordering schemes which introduce permutation matrices, but these permutations are not important in the context of this section, so they are suppressed.

If $\mathbf{x}^T$ *is the solution of* $\mathbf{x}^T\mathbf{L}_n = -\mathbf{r}^T$, *then each of the following statements is true. The stationary distribution of the chain is*

$$\boldsymbol{\pi}^T = \frac{1}{1+\|\mathbf{x}\|_1}(\mathbf{x}^T, 1). \tag{5.1}$$

The character of the chain is

$$\chi = \frac{\det(\mathbf{U}_n)}{\pi_n} = (1+\|\mathbf{x}\|_1)\det(\mathbf{U}_n). \tag{5.2}$$

The condition number for the chain is bounded above by

$$\kappa < \frac{2\delta(n-1)\pi_n}{\det(\mathbf{U}_n)} = \frac{2\delta(n-1)}{(1+\|\mathbf{x}\|_1)\det(\mathbf{U}_n)} \leq \frac{2(n-1)}{(1+\|\mathbf{x}\|_1)\det(\mathbf{U}_n)}. \tag{5.3}$$

The condition number for the chain is bounded below by

$$\pi_n \sum_{i=1}^{n-1} \frac{\pi_i}{u_{ii}} = \frac{1}{(1+\|\mathbf{x}\|_1)^2} \sum_{i=1}^{n-1} \frac{x_i}{u_{ii}} \leq \kappa \tag{5.4}$$

where u_{ii} *is the ith pivot in* $\mathbf{U}_n$.

Proof. Statements (5.1), (5.2), and (5.3) are straightforward consequences of the previous discussion. To establish (5.4), first recall from Lemma 3.6 that

$$a_{nn}^{\#} = \pi_n \overline{\boldsymbol{\pi}}^T \mathbf{A}_n^{-1}\mathbf{e} = \pi_n \overline{\boldsymbol{\pi}}^T \mathbf{U}_n^{-1}\mathbf{L}_n^{-1}\mathbf{e} > 0.$$

Since $\mathbf{U}_n^{-1} \geq \mathbf{0}$ and $\mathbf{L}_n^{-1} \geq \mathbf{0}$, it follows that $\overline{\boldsymbol{\pi}}^T\mathbf{U}_n^{-1}$ and $\mathbf{L}_n^{-1}\mathbf{e}$ can be written as

$$\overline{\boldsymbol{\pi}}^T\mathbf{U}_n^{-1} = \left(\frac{\pi_1}{u_{11}}, \frac{\pi_2}{u_{22}} + \alpha_2, \ldots, \frac{\pi_{n-1}}{u_{n-1,n-1}} + \alpha_{n-1}\right),$$

$$\mathbf{L}_n^{-1}\mathbf{e} = (1,\ 1+\beta_2, \ldots, 1+\beta_{n-1})^T$$

where each α_i and β_i is nonnegative, and consequently (setting $\alpha_0 = \beta_0 = 0$)

$$\overline{\boldsymbol{\pi}}^T\mathbf{A}_n^{-1}\mathbf{e} = \overline{\boldsymbol{\pi}}^T\mathbf{U}_n^{-1}\mathbf{L}_n^{-1}\mathbf{e} = \sum_{i=1}^{n-1} \frac{(\pi_i+\alpha_i)(1+\beta_i)}{u_{ii}} \geq \sum_{i=1}^{n-1} \frac{\pi_i}{u_{ii}}.$$

Therefore,

$$\kappa \geq a_{nn}^{\#} = \pi_n \overline{\boldsymbol{\pi}}^T\mathbf{U}_n^{-1}\mathbf{L}_n^{-1}\mathbf{e} \geq \pi_n \sum_{i=1}^{n-1} \frac{\pi_i}{u_{ii}} = \frac{1}{(1+\|\mathbf{x}\|_1)^2} \sum_{i=1}^{n-1} \frac{x_i}{u_{ii}}. \qquad \square$$

As mentioned before, the pivots or the determinant need not be indicators of the condition of a general nonsingular matrix. In particular, the absence of small pivots (or the existence of a large determinant) is not a guarantee of a well-conditioned matrix. However, for our special matrices $\mathbf{A} = \mathbf{I} - \mathbf{P}$, the bounds in Theorem 5.1 allow the pivots to be used as condition estimators.

COROLLARY 5.1. *For an irreducible Markov chain whose transition matrix is* $\mathbf{P}$, *suppose that the* LU *factorization of* $\mathbf{A} = \mathbf{I} - \mathbf{P}$ *and the stationary distribution* $\boldsymbol{\pi}^T$ *have been computed as described in Theorem* 5.1. *If the pivots* u_{ii} *are large relative to* π_n *in the sense that* $\pi_n / \det(\mathbf{U}_n)$ *is not too small, then the chain is well conditioned. If there are pivots* u_{ii} *which are small relative to* $\pi_n \pi_i$ *in the sense that*

$$\pi_n \sum_{i=1}^{n-1} \pi_i / u_{ii}$$

is large, then the chain is ill conditioned.

6. Using a QR factorization. The utility of orthogonal triangularization is well documented in the vast literature on matrix computations, and the use of a QR factorization to solve and analyze Markov chains is discussed by Golub and Meyer (1986). The following theorem shows that the character of an irreducible chain can be directly obtained from the diagonal entries of $\mathbf{R}$ and the last column of $\mathbf{Q}$, and this will establish an upper bound using a QR factorization which is analogous to that in Theorem 5.1 for an LU factorization. A lower bound analogous to the one in Theorem 5.1 is not readily available.

THEOREM 6.1. *For an irreducible Markov chain whose transition matrix is* $\mathbf{P}$, *the QR factorization of* $\mathbf{A} = \mathbf{I} - \mathbf{P}$ *is given by*

$$\mathbf{A} = \mathbf{QR} = \begin{pmatrix} \mathbf{Q}_n & \mathbf{c} \\ \mathbf{d}^T & q_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{R}_n & -\mathbf{R}_n \mathbf{e} \\ \mathbf{0} & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_n \mathbf{R}_n & -\mathbf{Q}_n \mathbf{R}_n \mathbf{e} \\ \mathbf{d}^T \mathbf{R}_n & -\mathbf{d}^T \mathbf{R}_n \mathbf{e} \end{pmatrix}.$$

If $\mathbf{q}$ *denotes the last column of* $\mathbf{Q}$, *then each of the following statements are true. The stationary distribution of the chain is*

$$\boldsymbol{\pi}^T = \frac{\mathbf{q}^T}{\sum_{i=1}^{n} q_{in}}. \tag{6.1}$$

The character of the chain is

$$\chi = \|\mathbf{q}\|_1 \det(\mathbf{R}_n). \tag{6.2}$$

The condition number for the chain is bounded above by

$$\kappa < \frac{2\delta(n-1)}{\|\mathbf{q}\|_1 \det(\mathbf{R}_n)} \leq \frac{2(n-1)}{\|\mathbf{q}\|_1 \det(\mathbf{R}_n)}. \tag{6.3}$$

Proof. The formula (6.1) for $\boldsymbol{\pi}^T$ is derived in Golub and Meyer (1986). To prove (6.2), first recall the result of Lemma 3.3, and observe that

$$\chi^2 = \left(\frac{\det \mathbf{A}_n}{\pi_n} \right)^2 = \frac{(\det \mathbf{Q}_n \mathbf{R}_n)^2}{\pi_n^2} = \frac{(\det \mathbf{Q}_n)^2 (\det \mathbf{R}_n)^2}{q_{nn}^2 / \|\mathbf{q}\|_1^2}.$$

Use the fact that $\mathbf{QQ}^T = \mathbf{I}$ implies $\mathbf{Q}_n \mathbf{Q}_n^T + \mathbf{cc}^T = \mathbf{I}$ to obtain

$$(\det \mathbf{Q})^2 = \det\left(\mathbf{Q}_n \mathbf{Q}_n^T\right) = \det\left(\mathbf{I} - \mathbf{cc}^T\right) = 1 - \mathbf{c}^T \mathbf{c} = q_{nn}^2,$$

and substitute this into the previous expression to obtain (6.2). The bound (6.3) is now a consequence of the result of Theorem 2.1. □

7. Concluding remarks. It has been argued that the sensitivity of an irreducible chain is primarily governed by how close the subdominant eigenvalues are to 1 in the sense that the condition number of the chain is bounded by

$$\frac{1}{n \min\limits_{\lambda_i \neq 1} |1 - \lambda_i|} \leq \kappa < \frac{2\delta(n-1)}{\chi}. \tag{7.1}$$

Although the upper bound explicitly involves n, it is generally not the case that $2\delta(n-1)/\chi$ grows in proportion to n. Except in the special case when the diagonal entries of $\mathbf{P}$ are 0, the term δ somewhat mitigates the presence of n because as n becomes larger, δ becomes smaller.

Computational experience suggests that $2\delta(n-1)/\chi$ is usually a rather conservative estimate of κ, and the term δ/χ by itself, although not always an upper bound for κ, is often of the same order of magnitude as κ. However, there exist pathological cases for which even δ/χ severely overestimates κ. This seems to occur for chains which are not too badly conditioned and no single eigenvalue is extremely close to 1, but enough eigenvalues are within range of 1 to force χ^{-1} to be too large. This suggests that for the purposes of bounding κ above, perhaps not all of the subdominant eigenvalues need to be taken into account. In a forthcoming article, Seneta (1993) addresses this issue by an analysis involving coefficients of ergodicity.

When direct methods are used to solve an irreducible chain, standard condition estimators can be used to produce reliable estimates for κ, but the cost of doing so is $O(n^2)$ operations beyond the solution process. The results of Theorems 5.1 and 6.1 make it possible to estimate κ with the same computations which produce $\boldsymbol{\pi}^T$. Although the bounds for κ produced by Theorem 5.1 are sometimes rather loose, they are nevertheless virtually free. One must balance the cost of obtaining condition estimates against the information one desires to obtain from these estimates.

8. Acknowledgments. The exposition of this article was enhanced by suggestions provided by Dianne O'Leary, Guy Latouche, and Paul Schweitzer.

REFERENCES

J. L. BARLOW (1993), *Error bounds for the computation of null vectors with applications to Markov chains*, SIAM J. Matrix Anal. Appl., 14, pp. 598–618.

A. BERMAN AND R. J. PLEMMONS (1979), *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York.

S. L. CAMPBELL AND C. D. MEYER (1991), *Generalized Inverses of Linear Transformations*, Dover Publications (1979 edition by Pitman Pub. Ltd., London), New York.

F. CHATELIN AND W. L. MIRANKER (1982), *Acceleration by aggregation of successive approximation methods*, Linear Algebra Appl., 43, pp. 17–47.

R. E. FUNDERLIC AND C. D. MEYER (1986), *Sensitivity of the stationary distribution vector for an ergodic Markov chain*, Linear Algebra Appl., 76, pp. 1–17.

G. H. GOLUB AND C. D. MEYER (1986), *Using the QR factorization and group inversion to compute, differentiate, and estimate the sensitivity of stationary probabilities for Markov chains*, SIAM J. Algebraic Discrete Meth., 7, pp. 273–281.

W. K. GRASSMANN, M. I. TAKSAR, AND D. P. HEYMAN (1985), *Regenerative analysis and steady state distributions for Markov chains*, Oper. Res., 33, pp. 1107–1116.

W. J. HARROD AND R. J. PLEMMONS (1984), *Comparison of some direct methods for computing stationary distributions of Markov chains*, SIAM J. Sci. Statist. Comput., 5, pp. 453–469.

M. HAVIV (1987), *Aggregation/disaggregation methods for computing the stationary distribution of a Markov chain*, SIAM J. Numer. Anal., 22, pp. 952–966.

R. A. HORN AND C. R. JOHNSON (1991), *Topics In Matrix Analysis*, Cambridge University Press, Cambridge.

M. IOSIFESCU (1980), *Finite Markov Processes and their Applications*, John Wiley and Sons, New York.

I. C. F. IPSEN AND C. D. MEYER (1994), *Uniform stability of Markov chains*, SIAM J. Algebraic Discrete Meth., 15, to appear.

J. G. KEMENY AND J. L. SNELL (1960), *Finite Markov Chains*, D. Van Nostrand, New York.

C. D. MEYER (1975), *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Review, 17, pp. 443–464.

——— (1980), *The condition of a finite Markov chain and perturbation bounds for the limiting probabilities*, SIAM J. Algebraic Discrete Meth., 1, pp. 273–283.

——— (1993), *The character of a finite Markov chain*, in Proceedings of the IMA Workshop on Linear Algebra, Markov Chains, and Queuing Models, C. D. Meyer and R. J. Plemmons, eds., Springer-Verlag, New York, to appear.

C. D. MEYER AND G. W. STEWART (1988), *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25, pp. 679–691.

P. J. SCHWEITZER (1968), *Perturbation theory and finite Markov chains*, J. Appl. Prob., 5, pp. 401–413.

E. SENETA (1991), *Sensitivity analysis, ergodicity coefficients, and rank-one updates for finite Markov chains*, in Numerical Solution Of Markov Chains, W. J. Stewart, ed., Probability: Pure and Applied, No. 8, Marcel Dekker, New York, pp. 121–129.

——— (1993), *Sensitivity of finite Markov chains under perturbation*, Statist. and Probab. Lett., 17, to appear.

G. W. STEWART AND G. ZHANG (1991), *On A direct method for the solution of nearly uncoupled Markov chains*, Numer. Math., 59, pp. 1–11.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 729–732, July 1994

002

STRONGLY INERTIA-PRESERVING MATRICES *

ABRAHAM BERMAN† AND DAFNA SHASHA†

Abstract. It is shown that a matrix can be strongly inertia preserving without being diagonally stable.

Key words. inertia-preserving matrices, strongly inertia-preserving matrices, diagonally stable and semistable matrices

AMS subject classifications. 15A18, 15A99, 93D05

1. Introduction.

All matrices in this note are real. For a square matrix A let $i_+(A)$ be the number of eigenvalues with a positive real part, $i_0(A)$ the number of pure imaginary eigenvalues, and $i_-(A)$ the number of eigenvalues with a negative real part. As usual, the triple

$$\text{In } A = \{i_+(A), i_0(A), i_-(A)\}$$

is the *inertia* of A. We further denote by $B_0(A)$ the class of all positive semidefinite matrices B with

$$(BA)_{ii} = 0, \qquad i = 1, \ldots, n,$$

and let

$$r_A = \min\{\text{rank } B; 0 \neq B \in B_0(A)\} \quad \text{if } B_0(A) \neq \{0\},$$

where

$$r_A = \infty \quad \text{if } B_0(A) = \{0\}.$$

An $n \times n$ matrix A is (*positive*) *stable* if $i_+(A) = n$. Motivated by the classical Lyapunov theorem, which says that A is stable if and only if there exists a positive definite matrix X such that $AX + XA^T$ is positive definite, we say that A is *Lyapunov diagonally stable* if there exists a positive diagonal matrix D such that $AD + DA^T$ is positive definite. Furthermore, A is *Lyapunov diagonally semistable* if there exists a positive diagonal matrix D such that $AD+DA^T$ is positive semidefinite, and A is *Lyapunov diagonally near stable* if it is Lyapunov diagonally semistable but not Lyapunov diagonally stable. For applications of these classes of matrices see [3] and [5].

We say that a matrix A is *inertia preserving* if $\text{In } AG = \text{In } G$ for every invertible diagonal matrix G and *strongly inertia preserving* if $\text{In } AG = \text{In } G$ for every (possibly singular) diagonal matrix G.

Observe that A is strongly inertia preserving if and only if every principal submatrix of A is inertia preserving.

---

*Received by the editors April 23, 1992; accepted for publication (in revised form) January 26, 1993. This research was supported by Technion Vice President for Research–the E. and J. Bishop Research Fund.

†Department of Mathematics, Technion–Israel Institute of Technology, Haifa, Israel 32000 (mar64aa@technion.bitnet)

In [2] it is shown that every Lyapunov diagonally stable matrix is strongly inertia preserving. Using an example given in [5], we show that the converse is not true by proving that a Lyapunov diagonally near stable matrix with $r_A \geq 3$ is strongly inertia preserving. We conjecture that Lyapunov diagonally semistable matrices are strongly inertia preserving if and only if $r_A \geq 3$.

2. Lyapunov diagonally semistable matrices with $r_A \geq 3$. We use the following result.

THEOREM 1 ([2, Corollary 5.4]). *A Lyapunov diagonally semistable matrix A is inertia preserving if and only if for every $B \in B_0(A)$ of rank not exceeding 2 and for every diagonal invertible G, the matrix BAG is skew symmetric only when $B = 0$.*

We start with a lemma in which A need not be Lyapunov diagonally semistable.

LEMMA 1. *Let $A[\alpha]$ be the principal submatrix of a square matrix A based on the indices in α. Then*

$$r_{A[\alpha]} \geq r_A.$$

Proof. If $B_0(A[\alpha]) = \{0\}$, then $r_{A[\alpha]} = \infty$. Otherwise, let $0 \neq C \in B_0(A[\alpha])$ be of minimal rank. Let B be a matrix of the same order as A such that $B[\alpha] = C$ and all other entries of B are zeros. Then, $B \in B_0(A)$ and $r_A \leq \text{rank } B = \text{rank } B[\alpha] = r_{A[\alpha]}$. □

THEOREM 2. *If A is Lyapunov diagonally semistable and $r_A \geq 3$, then A is strongly inertia preserving.*

Proof. If $r_A \geq 3$, then the only matrix of rank ≤ 2 in $B_0(A)$ is the zero matrix, so by Theorem 1, A is inertia preserving. By Lemma 2 and Theorem 1 all principal submatrices of A are inertia preserving, so A is strongly inertia preserving. □

By Theorem 3.4 of [6] or Theorem 3.1 of [4], a Lyapunov diagonally semistable matrix is Lyapunov diagonally stable if and only if $r_A = \infty$ ($B_0(A) = \{0\}$). In [5] it is shown that the matrix

$$A = \begin{bmatrix} 10 & 0 & 1 & 0 & 5 & 8 \\ -4 & 5 & 10 & 0 & -3 & -2 \\ 6 & -6 & 4 & -4 & 5 & -4 \\ 8 & 4 & 4 & 4 & 6 & 4 \\ -2 & 7 & -1 & -2 & 2 & 14 \\ 0 & 5 & 6 & 2 & -3 & 4 \end{bmatrix}$$

is Lyapunov diagonally semistable with $r_A = 3$. Thus, it is an example of a strongly inertia-preserving matrix that is not Lyapunov diagonally stable.

3. Lyapunov diagonally semistable matrices with $r_A \leq 2$. We start with a proposition where it is not assumed that A is Lyapunov diagonally semistable.

PROPOSITION 1. *A matrix A has a singular principal submatrix if and only if $r_A = 1$.*

Proof. Suppose $r_A = 1$. Then there exists a nonzero vector $\mathbf{x}$ such that $(\mathbf{x}\mathbf{x}^T A)_{ii} = 0$, for all i. Let α be the support of $\mathbf{x}$, that is, $\alpha = \{i; \mathbf{x}_i \neq 0\}$. Then $A[\alpha]$, the principal submatrix of A based on the indices in α, is singular.

Conversely, suppose $A[\alpha]$ is a singular principal submatrix of A. Let $\mathbf{x} \in \mathbb{R}^n$ be a nonzero vector such that

$$x[\{1, 2, \ldots, n\} - \alpha] = 0 \quad \text{and} \quad x[\alpha]^T A[\alpha] = 0.$$

Then $\mathbf{x}\mathbf{x}^T A = 0$, so $\mathbf{x}\mathbf{x}^T \in B_0(A)$ and $r_A = 1$. □

For Lyapunov diagonally semistable matrices, we suggest the following.

CONJECTURE 1. *Let A be Lyapunov diagonally semistable. Then A is strongly inertia preserving if and only if $r_A \geq 3$.*

To prove the conjecture, it suffices by the previous section and the above proposition to show that if $r_A = 2$, then A cannot be strongly inertia preserving.

A partial result in this direction is as follows.

PROPOSITION 2. *If $A \in \mathbb{R}^{3\times 3}$ is Lyapunov diagonally semistable and if $r_A = 2$, then A is not inertia preserving.*

Proof. We shall prove that there exists a nonzero matrix $B \in B_0(A)$ and an invertible diagonal matrix G such that BAG is skew symmetric; so, by Theorem 1, A is not inertia preserving.

Let B be a rank 2 matrix in $B_0(A)$. Then

$$BA = \begin{bmatrix} 0 & a & b \\ c & 0 & d \\ e & h & 0 \end{bmatrix}.$$

We show that there exists an invertible diagonal matrix $G = \operatorname{diag}\{g_1, g_2, g_3\}$ such that

$$BAG + GA^TB = 0.$$

Indeed, the system

$$\begin{aligned} cg_1 + ag_2 &= 0, \\ eg_1 + bg_3 &= 0, \\ fg_2 + dg_3 &= 0 \end{aligned}$$

has a nontrivial solution g_1, g_2, g_3, since

$$\det \begin{bmatrix} c & a & 0 \\ e & 0 & b \\ 0 & f & d \end{bmatrix} = -\det \begin{bmatrix} 0 & a & b \\ c & 0 & d \\ e & f & 0 \end{bmatrix} = \det BA,$$

and $\det BA = 0$ as rank $B = 2$.

To show that g_1, g_2, g_3, are nonzero, it is enough, without loss of generality, to prove the impossibility of the following two cases:

(a) $g_1 = 0, \quad g_2 \neq 0, \quad g_3 \neq 0.$

(b) $g_1 = g_2 = 0, \quad g_3 \neq 0.$

In the first case

$$\begin{aligned} ag_2 &= 0, \\ bg_3 &= 0, \\ fg_2 + dg_3 &= 0, \end{aligned}$$

where g_2 and g_3 are nonzero. Consequently, $a = b = 0$, so

$$BA = \begin{bmatrix} 0 & 0 & 0 \\ c & 0 & d \\ e & f & 0 \end{bmatrix}.$$

Thus, $B_1A = 0$, where B_1 denotes the first row of B.

By Proposition 1, A is nonsingular; hence $B_1 = 0$, so

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_{22} & b_{23} \\ 0 & b_{23} & b_{33} \end{bmatrix}.$$

Since the 3×3 matrix A has no singular principal submatrices and since a Lyapunov diagonally semistable matrix is Lyapunov diagonally stable if and only if it has no singular principal submatrices (see [1]), the matrix $A[2,3]$ is Lyapunov diagonally stable. This implies that $B[2,3] = 0$ in contradiction to rank $B = 2$.

In case (b),

$$\begin{aligned} bg_3 &= 0, \\ dg_3 &= 0, \end{aligned}$$

where g_3 is nonzero. Thus,

$$BA = \begin{bmatrix} 0 & a & 0 \\ c & 0 & 0 \\ e & f & 0 \end{bmatrix}.$$

We now use the fact that A is Lyapunov diagonally semistable, so $AD + DA^T$ is positive semidefinite for a positive diagonal matrix D, which implies (see [4], [6]) that

$$B(AD + DA^T) = 0. \tag{1}$$

Since the third columns of BA and BAD equal zero, it follows by (1) that the third column of BDA^T also vanishes, so the third column and row of AD lie in the one dimensional null space of B. Observe that by Proposition 1, a_{33} is nonzero, so the third column and the third row of AD are equal. Let

$$AD = \begin{bmatrix} k & l & m \\ n & p & q \\ m & q & t \end{bmatrix}.$$

Then

$$AD + DA^T = \begin{bmatrix} 2k & 1+n & 2m \\ 1+n & 2p & 2q \\ 2m & 2q & 2t \end{bmatrix}.$$

Thus, $AD[2,3] = \frac{1}{2}(AD + DA^T)[2,3]$. But, by (1), rank $(AD + DA^T) = 1$, so $(AD + DA^T)[2,3]$ is singular. This implies that $AD[2,3]$ and $A[2,3]$ are singular, in contradiction to the assumption that $r_A = 2$. □

REFERENCES

[1] G. P. BARKER, A. BERMAN, AND R. J. PLEMMONS, *Positive diagonal solutions to the Lyapunov equations*, Linear and Multilinear Algebra, 5 (1978), pp. 249–266.

[2] A. BERMAN AND D. SHASHA, *Inertia preserving matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 209–219.

[3] D. HERSHKOWITZ, *Recent directions in matrix stability*, Linear Algebra Appl., 171 (1992), pp. 161–186.

[4] D. HERSHKOWITZ AND D. SHASHA, *Cones of real positive semidefinite matrices associated with matrix stability*, Linear and Multilinear Algebra, 23 (1988), pp. 165–181.

[5] J. F. B. M. KRAAIJEVANGER AND J. SCHNEID, *On the unique solvability of the Runge–Kutta equation*, Numer. Math., 59 (1991), pp. 129–157.

[6] D. SHASHA AND A. BERMAN, *On the uniqueness of the Lyapunov scaling factors*, Linear Algebra Appl., 91 (1987), pp. 51–63.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 733–754, July 1994

DYNAMICAL SYSTEMS THAT COMPUTE BALANCED REALIZATIONS AND THE SINGULAR VALUE DECOMPOSITION*

U. HELMKE†, J. B. MOORE‡, AND J. E. PERKINS‡

Abstract. The tasks of finding balanced realizations in systems theory and the singular value decomposition (SVD) of matrix theory are accomplished by finding the limiting solutions of differential equations. Several alternative sets of equations and their convergence properties are investigated. The dynamical systems for these tasks generate flows on the space of realizations that leave the transfer functions invariant. They are termed isodynamical flows. Isodynamical flows are generalizations of isospectral flows on matrices. These flows evolve on the actual system matrices and thus remove the need for considering coordinate transformation matrices. The methods are motivated by the power of parallel processing and the ability of a differential equations approach to tackle time-varying or adaptive tasks.

Key words. balanced realization, singular value decomposition, gradient flow, differential equation, dynamical systems

AMS subject classifications. 93B20, 93B40, 15A18, 65F15

1. Introduction.

In current practice, the problems of finding a balanced realization for a linear control system, as well as achieving an SVD of a matrix are solved using algebraic matrix manipulations, implemented in standard computer programs. Balanced realizations are a useful tool in systems theory to increase numerical robustness, and they allow a sensible model order reduction to be performed. This operation has been widely studied [10], [11] and computation methods have been described. Certainly, these methods are widely used, reliable, and well understood. On the other hand, recent advances in neural network theory and associative memories have shown that gradient-type algorithms can lead to effective and fast methods for algebraic tasks such as principle component analysis. This latter task is equivalent to the SVD. It follows that gradient flows can be an effective tool for SVD, although the full possibilities and limitations of this approach are not yet fully clear.

Brockett [1], again motivated by the renewed interest in neural networks, parallel processing, and analog computing, has also shown that other linear algebra and combinatorial problems can be solved in terms of the limiting solutions of ordinary differential equations (ODEs) that are gradient flows on orthogonal matrices. In [2] a systematic approach to balanced realizations of linear systems was developed, which treats balanced realizations as the global minima of objective functions, defined on the set of all realizations of a given transfer function. Aspects of this work are generalized in [3] for the task of finding an SVD using gradient flows on unitary matrices. In an earlier paper [4], it is shown how certain types of balancing problems can be solved using gradient flows on positive definite matrices with an exponential rate of convergence. Such algorithms are possibly suitable for application to time-varying systems [5].

In this paper, a systematic attempt is made to construct and analyze dynamical systems that are capable of achieving balancing or the SVD. Based on the cost

* Received by the editors November 25, 1991; accepted for publication (in revised form) September 23, 1992. This work was partially supported by Boeing (BCAC).

† Department of Mathematics, University of Regensburg, 93040 Regensburg, Germany (helmke@vax1.rz.uni-regensburg.d400.de).

‡ Department of Systems Engineering, Research School of Physical Sciences, Australian National University, GPO Box 4, Canberra ACT 2601 (jane@oberon.dsto.gov.au, jbm101@rsphy1.anu.edu.au).

function approach developed in [2], we propose several different gradient flows that solve the problem of finding a balanced realization, given an initial system realization. Each of these equations has an exponential rate of convergence and we compare their respective rates. It is envisaged that for particular applications there will be one gradient flow that will give a better convergence rate than other algorithms. First we review the linear and quadratic gradient flows of [4] that evolve on $P = T'T > 0$, where T is the state space transformation matrix that gives the balanced realization. The next solution method we consider are differential equations that evolve on the actual transformation matrix T. This solution method is of interest because it circumvents the need to find T given $P = T'T$.

Next we propose alternative ODEs that solve the balanced realization problem. These differential equations, termed isodynamical flows, evolve on the actual system matrices (A, B, C) rather than having the intermediate step of transformation matrices. They have the obvious advantage of immediacy as well as giving a clearer indication as to how the system is evolving. This is the first time a *direct* method to compute balanced realizations, without computing any balancing transformations, has been given. The class of all isodynamical flows can be viewed as a generalization of the isospectral flows, studied in matrix theory, as in [1], [3], [6], [7], [9], and their references.

In §2 gradient flows that give the transformation matrices for balanced realizations are studied, and in §3 related ODEs are developed for a direct evolution of the system matrices. In §4 flows achieving the SVD of a matrix are studied, and in §5 conclusions are drawn. The Appendix summarizes important results about gradient flows on manifolds.

2. Gradient flows for balancing transformations. In this section we consider the problem of computing balancing coordinate transformations via differential equations. While a part of this problem has been already considered in [4], we review some of the material developed in [4] and emphasize some new points as well.

We consider linear dynamical systems in continuous or discrete time

$$\left\{ \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{array} \right\} \quad \text{or} \quad \left\{ \begin{array}{c} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{array} \right\}$$

defined by the system matrices $(A, B, C) \in \Re^{n\times n} \times \Re^{n\times m} \times \Re^{p\times n}$. Such a system is called asymptotically stable if the eigenvalues of A are in the open complex left half-plane or in the open unit disc, respectively. For any asymptotically stable system (A,B,C), the controllability and observability gramians W_c and W_o are, respectively, defined in discrete time and continuous time by the symmetric matrices

$$W_c = \sum_{k=0}^{\infty} A^k BB'A'^k, \qquad W_o = \sum_{k=0}^{\infty} A'^k C'CA^k, \tag{2.1a}$$

$$W_c = \int_0^{\infty} e^{At} BB' e^{tA'} \mathrm{dt}, \qquad W_o = \int_0^{\infty} e^{tA'} C'Ce^{At} \mathrm{dt}. \tag{2.1b}$$

For unstable systems the controllability and observability gramians are likewise defined by finite sums or integrals rather than by the above infinite sums or integrals. In the following we will assume asymptotic stability of A; however, all results hold mutatis mutandis in the unstable case using finite gramians. To emphasize the dependence of the gramians on (A, B, C), we also write $W_c(A, B)$ and $W_o(A, C)$ for the controllability and observability gramians of (A, B, C).

In the sequel we fix an initial asymptotically stable controllable and observable realization $(A, B, C) \in \Re^{n\times n} \times \Re^{n\times m} \times \Re^{p\times n}$ of a given transfer function $G(s) = C(sI - A)^{-1}B \in \Re(s)^{p\times m}$. Thus, by Kalman's realization theorem, see, e.g., Kailath [8], all other minimal realizations of $G(s)$ are of the form (TAT^{-1}, TB, CT^{-1}) for a uniquely determined invertible coordinate transformation T.

Any linear change of coordinates in the state space $\Re^n$ by an invertible transformation $T \in GL(n, \Re)$ changes the realization according to $(A, B, C) \mapsto (TAT^{-1}, TB, CT^{-1})$ and thus transforms the gramians via

$$W_c \mapsto TW_cT', \qquad W_o \mapsto (T')^{-1}W_oT^{-1}. \tag{2.2}$$

We call a state space representation (A, B, C) of the transfer function *balanced* if $W_c = W_o$. This is more general than the usual definition of balanced realizations, (see Moore [10]), which requires that $W_c = W_o = diagonal$. In this case we refer to (A, B, C) as a *diagonal balanced realization*, which is thus one particular realization of our class of balanced realizations.

To obtain a quantitative measure of how the gramians change for the various realizations of a transfer function, we consider the function

$$\begin{aligned} \Phi(T) &= \operatorname{tr}(W_c(TAT^{-1}, TB) + W_o(TAT^{-1}, CT^{-1})) \\ &= \operatorname{tr}(TW_cT' + (T')^{-1}W_oT^{-1}) \\ &= \operatorname{tr}(W_cT'T + W_o(T'T)^{-1}) \\ &= \operatorname{tr}(W_cP + W_oP^{-1}) \end{aligned} \tag{2.3}$$

with

$$P = T'T. \tag{2.4}$$

Note that $\Phi(T)$ is the sum of the eigenvalues of the controllability and observability gramians of (TAT^{-1}, TB, CT^{-1}) and is thus a crude numerical measure for the controllability and observability of (TAT^{-1}, TB, CT^{-1}).

2.1. Balancing flows of positive definite matrices. Let $\mathcal{P}(n)$ denote the set of positive definite real symmetric $n\times n$ matrices $P = P' > 0$. $\mathcal{P}(n)$ is an open, convex subset of the set of all symmetric $n \times n$ matrices and is diffeomorphic to Euclidean space $\Re^{(1/2)n(n+1)}$. By (2.3) we are led to study the function

$$\phi : \mathcal{P}(n) \to \Re, \qquad \phi(P) = \operatorname{tr}(W_cP + W_oP^{-1}). \tag{2.5}$$

For a proof of the following results we refer to [4] and [9].

LEMMA 2.1 ([4], [9]). *Let W_c, W_o be the controllability and observability gramians* (2.1) *of an asymptotically stable minimal realization (A, B, C). Then the function $\phi : \mathcal{P}(n) \to \Re, \phi(P) = \operatorname{tr}(W_cP + W_oP^{-1})$, defined on the set $\mathcal{P}(n)$ of positive definite symmetric matrices, has compact sublevel sets, i.e., for all $a \in \Re$ then $\{P \in \mathcal{P}(n) \mid \operatorname{tr}(W_cP + W_oP^{-1}) \le a\}$ is a compact subset of $\mathcal{P}(n)$. In particular, there exists a minimizing $P = P' > 0$ for the function $\phi : \mathcal{P}(n) \to \Re$ defined by* (2.5).

While Lemma 2.1 establishes the existence of a minimizing $P_\infty = P'_\infty > 0$ for the function (2.5), Theorem 2.2 provides a more constructive approach towards finding the minimum by showing that it is the global attractor for the gradient flow on $\mathcal{P}(n)$.

THEOREM 2.2. *Linear index gradient flow* ([4]). *Let W_c, W_o denote the controllability and observability gramians* (2.1) *of an asymptotically stable, controllable, and observable realization (A, B, C).*

(a) *There exists a unique* $P_\infty = P'_\infty > 0$, *which minimizes* $\phi : \mathcal{P}(n) \to \Re, \phi(P) = \operatorname{tr}(W_c P + W_o P^{-1})$, *and* $T_\infty = P_\infty^{1/2}$ *is a balancing transformation for* (A, B, C). *This minimum is given by*

$$P_\infty = W_c^{-1/2}(W_c^{1/2} W_o W_c^{1/2})^{1/2} W_c^{-1/2}.$$

(b) *The gradient flow* $\dot{P}(t) = -\nabla \phi(P(t))$ *of* $\phi : \mathcal{P}(n) \to \Re$ *is given by*

$$\dot{P} = P^{-1} W_o P^{-1} - W_c. \tag{2.6}$$

For every initial condition $P_o = P'_o > 0$, $P(t)$ *exists for all* $t \geq 0$ *and converges exponentially fast to* P_∞ *as* $t \to \infty$ *with a lower bound for the rate of exponential convergence given by*

$$\rho \geq 2 \frac{\lambda_{\min}(W_c)^{3/2}}{\lambda_{\max}(W_o)^{1/2}}, \tag{2.7}$$

where $\lambda_{\min}(A)$ *respectively* $\lambda_{\max}(A)$ *denote the smallest, respectively largest, eigenvalue of* A.

In the sequel we refer to (2.6) as the *linear index gradient flow.* Instead of minimizing $\phi(P)$, we might as well consider the minimization problem for the quadratic index function

$$\Psi(P) = \operatorname{tr}((W_c P)^2 + (W_o P^{-1})^2) \tag{2.8}$$

over all positive definite symmetric matrices $P = P' > 0$.

Since, for $P = T'T$, $\Psi(P)$ is equal to $\operatorname{tr}[(TW_cT')^2 + ((T')^{-1}W_oT^{-1})^2]$, the minimization problem for (2.8) is equivalent to minimizing $\operatorname{tr}[(W_c(TAT^{-1}, TB))^2 + (W_o(TAT^{-1}, CT^{-1}))^2]$ over the set of all realizations (TAT^{-1}, TB, CT^{-1}) of a given transfer function $G(s) = C(sI - A)^{-1}B$. Thus $\Psi(P)$ is the sum of the squared eigenvalues of the controllability and observability gramians of (TAT^{-1}, TB, CT^{-1}). Note also that

$$tr[(TW_cT')^2 + ((T')^{-1}W_oT^{-1})^2] = \|TW_cT' - (T')^{-1}W_oT^{-1}\|^2 + 2tr[W_cW_o].$$

Thus minimizing this quadratic index function is equivalent to minimizing the least square distance $\|TW_cT' - (T')^{-1}W_oT^{-1}\|^2$.

THEOREM 2.3. *Quadratic index gradient flow* ([4]). *Under the same hypotheses as for Theorem* 2.2, *we have:*

(a)

$$P_\infty = W_c^{-1/2}(W_c^{1/2} W_o W_c^{1/2})^{1/2} W_c^{-1/2}$$

is the uniquely determined $P \in \mathcal{P}(n)$ *which minimizes* $\Psi : \mathcal{P}(n) \to \Re$ *and* $T_\infty = P_\infty^{1/2}$ *is a balancing transformation for* (A, B, C).

(b) *The gradient flow* $\dot{P}(t) = -\nabla \Psi(P(t))$ *on* $\mathcal{P}(n)$ *is*

$$\dot{P} = 2P^{-1}W_oP^{-1}W_oP^{-1} - 2W_cPW_c. \tag{2.9}$$

For all initial conditions $P_o = P'_o > 0$, *the solution* $P(t)$ *of* (2.9) *exists for all* $t \geq 0$ *and converges exponentially to* P_∞. *A lower bound on the rate of exponential convergence is*

$$\rho > 4\lambda_{\min}(W_c)^2. \tag{2.10}$$

We refer to (2.9) as *quadratic index gradient flow.* The above results show that both algorithms converge exponentially fast to P_∞. Both algorithms are rather slow if the smallest singular value of W_c is near to zero, i.e., if the system is nearly uncontrollable. In contrast to this behaviour, (2.7) shows that the convergence of the linear index flow becomes relatively fast if $\lambda_{\max}(W_o)$; that is, the 2-norm $\|W_o\|_2$ of the observability gramian is small. Similarly, the bound (2.10) for the rate of convergence of the quadratic index flow is independent of W_o and therefore we expect a certain amount of robustness of our algorithms in the case where the observability properties of the system are poor.

In general, the quadratic index flow seems to behave better than the linear index flow, at least if the smallest singular value of the associated Hankel operator of (A, B, C) is greater than $\frac{1}{2}$, i.e., if $\lambda_{\min}(W_o W_c) > \frac{1}{4}$. This is supported by the following simulations.

Simulations. The following simulations show the exponential convergence of the diagonal elements of P towards the solution matrix P_∞ and illustrate what might affect the convergence rate. In Figs. 1(a)–(c) we have

$$W_o = W_3 = \begin{bmatrix} 7 & 4 & 4 & 3 \\ 4 & 4 & 2 & 2 \\ 4 & 2 & 4 & 1 \\ 3 & 2 & 1 & 5 \end{bmatrix} \quad \text{and} \quad W_c = W_4 = \begin{bmatrix} 5 & 2 & 0 & 3 \\ 2 & 7 & -1 & -1 \\ 0 & -1 & 5 & 2 \\ 3 & -1 & 2 & 6 \end{bmatrix},$$

so that $\lambda_{\min}(W_o W_c) \approx 1.7142 > \frac{1}{4}$. Figure 1(a) concerns the linear index flow, while Fig. 1(b) shows the evolution of the quadratic index flow, both using $P(0) = P_1$, where

$$P(0) = P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad P(0) = P_2 = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}.$$

Figure 1(c) shows the evolution of both algorithms with a starting value of $P(0) = P_2$. These three simulations demonstrate that the quadratic algorithm converges more rapidly than the linear algorithm when $\lambda_{\min}(W_o W_c) > \frac{1}{4}$. This rapid convergence rate is achieved at the expense of twice the number of matrix multiplications in calculating the gradient.

In Fig. 1(d),

$$W_o = W_1 = \begin{bmatrix} 7 & 4 & 4 & 3 \\ 4 & 4 & 2 & 2 \\ 4 & 2 & 4 & 1 \\ 3 & 2 & 1 & 3 \end{bmatrix} \quad \text{and} \quad W_c = W_2 = \begin{bmatrix} 5 & 4 & 0 & 3 \\ 4 & 7 & -1 & -1 \\ 0 & -1 & 5 & 2 \\ 3 & -1 & 2 & 6 \end{bmatrix},$$

so that $\lambda_{\min}(W_o W_c) \approx 0.207 < \frac{1}{4}$. Figure 1(d) compares the linear index flow behaviour with that of the quadratic index flow for $P(0) = P_1$. This simulation demonstrates that the linear algorithm does not necessarily converge more rapidly than the quadratic algorithm when $\lambda_{\min}(W_o W_c) < \frac{1}{4}$, because the bounds on convergence rates are conservative.

2.2. Gradient flows for balancing transformations. In the previous section we studied gradient flows that converged to $P_\infty = T_\infty^2$, where T_∞ is the unique symmetric positive definite balancing transformation for a given asymptotically stable

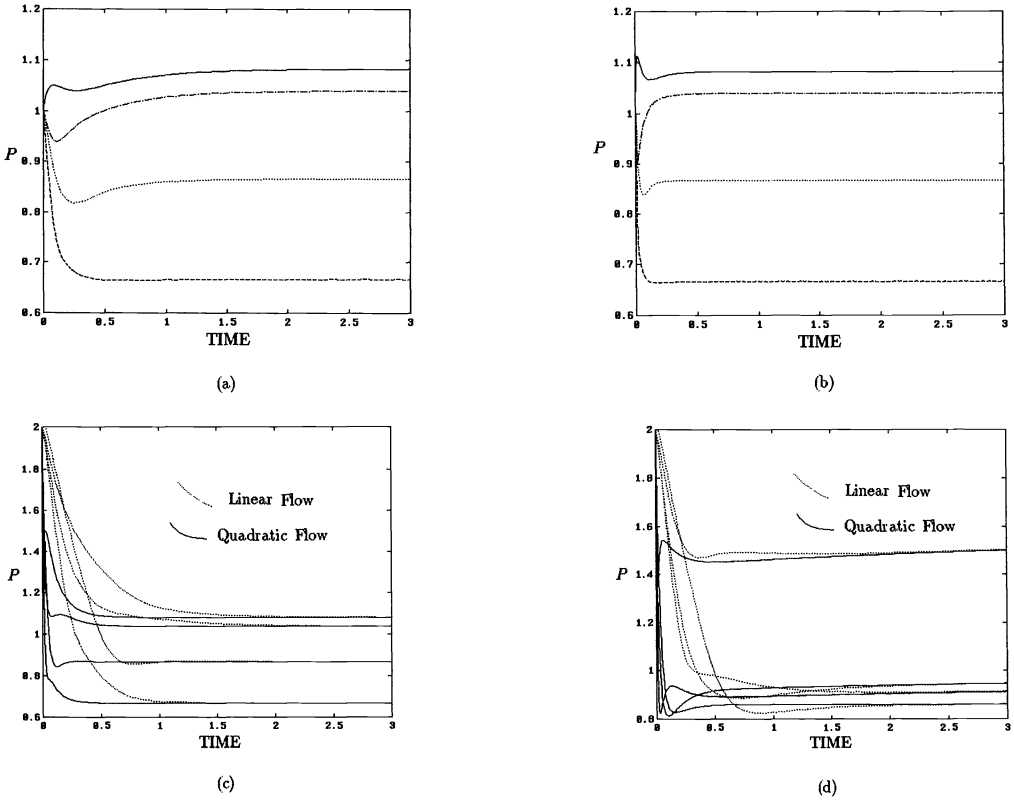


FIG. 1. *Comparison of linear and quadratic flows on $P(t)$.* (a) *Linear index flow when* $\lambda_{\min}(W_oW_c) > \frac{1}{4}$. (b) *Quadratic index flows when* $\lambda_{\min}(W_oW_c) > \frac{1}{4}$. (c) *Linear and quadratic flows when* $\lambda_{\min}(W_oW_c) > \frac{1}{4}$. (d) *Linear and quadratic flows when* $\lambda_{\min}(W_oW_c) < \frac{1}{4}$.

system (A, B, C). T_∞ is then obtained as the unique symmetric positive definite square root of P_∞. In this section we address the general problem of determining *all* balancing transformations $T \in GL(n, \Re)$ for a given asymptotically stable system (A, B, C), using a suitable gradient flow on the set $GL(n, \Re)$ of all invertible $n \times n$-matrices. This allows us to compute balancing transformations without squaring down an operator; cf. [11].

Thus for $T \in GL(n, \Re)$, we consider the cost function $\Phi : GL(n, \Re) \to \Re$ defined by

$$\Phi(T) = \operatorname{tr}(TW_cT' + (T')^{-1}W_oT^{-1}) \tag{2.11}$$

and the associated gradient flow $\dot{T} = -\nabla \Phi(T)$ on $GL(n, \Re)$. Of course, to define the gradient of a function, we must specify a Riemannian metric with respect to which the gradient is defined; see the Appendix. Here, as in the previous section, we endow $GL(n, \Re)$ with its standard Riemannian metric

$$\langle A, B \rangle = 2\operatorname{tr}(A'B), \tag{2.12}$$

i.e., with the constant Frobenius inner product (2.12) defined on the tangent spaces of $GL(n, \Re)$.

THEOREM 2.4. *Let W_c and W_o denote the controllability and observability gramians of the asymptotically stable, controllable and observable realization (A, B, C).*

(a) *The gradient flow* $\dot{T} = -\nabla \Phi(T)$ *of* $\Phi : GL(n, \Re) \to \Re$ *is*

$$\dot{T} = (T')^{-1} W_o (T'T)^{-1} - TW_c, \tag{2.13}$$

and for any initial condition $T_0 \in GL(n, \Re)$, *the solution* $T(t)$ *of* (2.13), $T(0) = T_0$ *exists in* $GL(n, \Re)$ *for all* $t \geq 0$.

(b) *For any initial condition* $T_0 \in GL(n, \Re)$, *the solution* $T(t)$ *of* (2.13) *converges to a balancing transformation* $T_\infty \in GL(n, \Re)$, *and all balancing transformations can be obtained in this way for suitable initial conditions* $T_0 \in GL(n, \Re)$.

(c) *Let* T_∞ *be a balancing transformation and let* $\mathrm{In}(T_\infty)$ *denote the set of all* $T_0 \in GL(n, \Re)$, *such that the solution* $T(t)$ *of* (2.13) *with* $T(0) = T_0$ *converges to* T_∞ *as* $t \to \infty$. *Then* $\mathrm{In}(T_\infty)$ *is an immersed invariant submanifold of* $GL(n, \Re)$ *of dimension* $n(n+1)/2$ *and every solution* $T(t) \in \mathrm{In}(T_\infty)$ *converges exponentially fast in* $\mathrm{In}(T_\infty)$ *to* T_∞.

Proof. $GL(n, \Re)$ is an open subset of $\Re^{n \times n}$ and therefore the tangent space of $GL(n, \Re)$ at T can be identified with the $\Re$-vectorspace of all real $n \times n$ matrices $\xi \in \Re^{n \times n}$. The Fréchet derivative of $\Phi : GL(n, \Re) \to \Re$, at T is the linear operator on the tangent space of $GL(n, \Re)$ at T defined by

$$D\Phi|_T(\xi) = 2\mathrm{tr}[(W_c T' - T^{-1}(T')^{-1} W_o T^{-1})\xi] = 2\mathrm{tr}[(TW_c - (T')^{-1} W_o (T'T)^{-1})'\xi]$$

for all $\xi \in \Re^{n \times n}$. Thus the gradient of Φ with respect to the Riemannian metric (2.12) is

$$\nabla \Phi(T) = TW_c - (T')^{-1} W_o (T'T)^{-1}.$$

To prove that the gradient flow (2.13) is complete, i.e., that the solutions $T(t)$ exist for all $t \geq 0$, it suffices to show that $\Phi : GL(n, \Re) \to \Re_+$ is proper, i.e., that the pre-image $\Phi^{-1}(K)$ of any compact subset $K \subset \Re_+$ is compact in $GL(n, \Re)$. More generally, a continuous map $f : X \to Y$ between topological Hausdorff spaces is called proper if the inverse image of $f^{-1}(K)$ of any compact subset $K \subset Y$ is compact. Let $\mathcal{P}(n) = \{P \in GL(n, \Re) | P = P' > 0\}$. By Lemma 2.1, $P \mapsto \mathrm{tr}(W_c P + W_o P^{-1})$ is a proper function on $\mathcal{P}(n)$. By the polar decomposition, the set of invertible matrices T corresponding to a fixed matrix $T'T$ is compact. More generally, we conclude that the map $GL(n, \Re) \to \mathcal{P}(n), T \mapsto T'T$ is proper. Thus Φ is the composition of proper maps and therefore it is also proper. This shows (a). To prove (b), we note that by (a) and a well-known property of gradient flows, any solution $T(t)$ converges to an equilibrium point T_∞ of (2.13).

$$(T'_\infty)^{-1} W_o (T'_\infty T_\infty)^{-1} = T_\infty W_c \iff (T'_\infty)^{-1} W_o T_\infty^{-1} = T_\infty W_c T'_\infty$$

and hence T_∞ is balancing. This shows (b).

To prove (c), we use the following lemma, where

$$E := \{T_\infty \in GL(n, \Re) | (T'_\infty T_\infty) W_c (T'_\infty T_\infty) = W_o\} \tag{2.14}$$

denotes the set of equilibria points of (2.13).

LEMMA 2.5. *The tangent space of* E *at* $T_\infty \in E$ *is*

$$T_{T_\infty} E = \{S \in \Re^{n \times n} | S' T_\infty + T'_\infty S = 0\}. \tag{2.15}$$

Proof. Let P_∞ denote the unique symmetric positive definite solution of $PW_cP = W_o$. Thus $E = \{T | T'T = P_\infty\}$ and therefore $T_{T_\infty}E$ is the kernel of the derivative of $T \mapsto T'T - P_\infty$ at T_∞. Thus $S \in T_{T_\infty}E$ if and only if $S'T_\infty + T'_\infty S = 0$. □

Let

$$\phi(P) = \operatorname{tr}(W_cP + W_oP^{-1})$$

and

$$\lambda(T) = T'T.$$

Thus $\Phi(T) = \phi(\lambda(T))$. By Theorem 2.2, see also [4] and [2],

(i) $D\phi|_{P_\infty} = 0$.

(ii) $D^2\phi|_{P_\infty} > 0$.

Let X denote the matrix representing the linear operator $D\lambda|_{T_\infty}(S) = T'_\infty S + S'T_\infty$. Using the chain rule, we obtain

$$D^2\Phi|_{T_\infty} = X' \cdot D^2\phi|_{P_\infty} \cdot X \tag{2.16}$$

for all $T_\infty \in E$. By (ii) and (2.16), $D^2\Phi|_{T_\infty} \geq 0$ and $D^2\Phi|T_\infty$ degenerates exactly on the kernel of X, i.e., on the tangent space $T_{T_\infty}E$. Thus Φ is a Morse–Bott function; see the Appendix. Thus Proposition A.3 implies that every solution $T(t)$ of (2.13) converges to an equilibrium point. Moreover, the equilibrium set E is normally hyperbolic.

It follows from the theory of stable manifolds (see, e.g., Irwin [12]) that $\operatorname{In}(T_\infty)$ is the stable manifold of (2.13) at T_∞ and thus is an immersed invariant submanifold of $GL(n, \Re)$ of dimension $\dim GL(n, \Re) - \dim E = n^2 - n(n-1)/2 = n(n+1)/2$. Since the convergence is always exponential on stable manifolds, this completes the proof of (c). □

Now consider the following quadratic version of our objective function Φ. For $T \in GL(n, \Re)$, let $\Psi : GL(n, \Re) \to \Re$ be defined by

$$\Psi(T) := \operatorname{tr}((TW_cT')^2 + ((T')^{-1}W_oT^{-1})^2). \tag{2.17}$$

The gradient flow $\dot{T} = -\nabla \Psi(T)$ on $GL(n, \Re)$ is easily computed to be

$$\dot{T} = (T')^{-1}W_o(T'T)^{-1}W_o(T'T)^{-1} - TW_cT'TW_c. \tag{2.18}$$

The same arguments as for Theorem 2.4 show that for all initial conditions $T_o \in GL(n, \Re)$, the solution $T(t) \in GL(n, \Re)$ of (2.18) exists for all $t \geq 0$ and converges to a balancing transformation for (A, B, C). Thus we can also use (2.18) or suitable discretized versions to compute balancing transformations for a given asymptotically stable minimal realization (A, B, C). We illustrate the behaviour of the gradient flows (2.13) and (2.18) by means of the following simulation experiments.

In Fig. 2 the diagonal entries of $T(t)$ are plotted. Figure 2(a) uses $W_o = W_1$, $W_c = W_2$ and a starting value of $T_0 = P_1$, in (2.13). Figure 2(b) has the same value for the gramians, but has a starting value of

$$T_0 = \begin{bmatrix} 1 & 3 & 4 & 2 \\ 4 & 3 & 2 & 5 \\ 3 & 2 & 4 & 1 \\ 2 & 4 & 3 & 4 \end{bmatrix}.$$

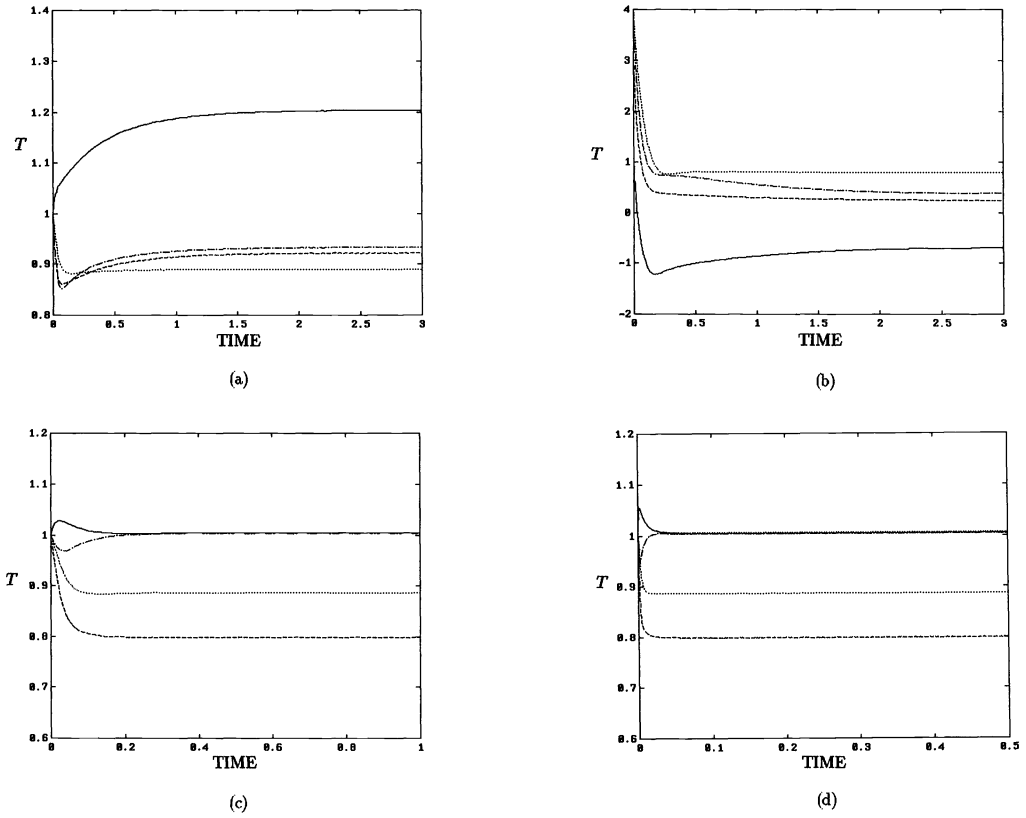


FIG. 2. *Comparison of linear and quadratic index flows on* $GL(n, \Re)$. (a) *Linear flow on* T *when* $\lambda_{\min}(W_o W_c) < \frac{1}{4}$. (b) *Quadratic flow on* T *when* $\lambda_{\min}(W_o W_c) < \frac{1}{4}$. (c) *Linear flow on* T *when* $\lambda_{\min}(W_o W_c) > \frac{1}{4}$. (d) *Quadratic flow on* T *when* $\lambda_{\min}(W_o W_c) > \frac{1}{4}$.

It can be observed that these T_0 values give different final solutions, both of which are generalized balancing transformations. Figures 2(c)–(d) use $W_o = W_3$, $W_c = W_4$ and a starting value of $T_0 = P_1$. Figure 2(c) uses (2.13) while Fig. 2(d) uses (2.18). Note that in this case, (2.18) converges more rapidly than (2.13).

2.3. Diagonal balancing transformations. Here we address the related issue of computing diagonal balancing transformations T for a given asymptotically stable minimal realization, i.e., T satisfies

$$TW_cT' = (T')^{-1}W_oT^{-1} = \text{diagonal}.$$

Any such diagonal balancing transformation T is of the form $T = \Theta \cdot T_\infty$, where $T_\infty = P_\infty^{1/2}$ is the uniquely determined positive definite symmetric balancing transformation whose existence is guaranteed by Theorem 2.1 and where Θ is an orthogonal matrix that diagonalizes $T_\infty W_c T_\infty' = (T_\infty')^{-1} W_o T_\infty^{-1}$.

Let us consider a fixed diagonal positive definite matrix $N = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with distinct eigenvalues $\lambda_1 > \cdots > \lambda_n > 0$. Using N, a weighted cost function for balancing is defined by

$$\Phi_N(T) = \operatorname{tr}(NTW_cT' + N(T')^{-1}W_oT^{-1}). \tag{2.19}$$

The following lemma characterizes the diagonal balancing transformations as the critical points of the weighted cost function Φ_N on $GL(n,\Re)$.

LEMMA 2.6. *Let* $N = \operatorname{diag}(\lambda_1,\dots,\lambda_n)$ *with* $\lambda_1 > \cdots > \lambda_n > 0$ *and let* W_c, W_o *denote the controllability and observability gramians of an asymptotically stable minimal realization* (A,B,C). *Then*

(a) $T \in GL(n,\Re)$ *is a critical point of* $\Phi_N : GL(n,\Re) \to \Re$, $\Phi_N(T) = \operatorname{tr}(NTW_cT' + N(T')^{-1}W_oT^{-1})$, *if and only if* T *is a diagonal balancing transformation, i.e.,*

$$TW_cT' = (T')^{-1}W_oT^{-1} = \text{diagonal}.$$

(b) $\Phi_N : GL(n,\Re) \to \Re$ *has compact sublevel sets. In particular, a global minimum* $T_{\min} \in GL(n,\Re)$ *of* $\Phi_N : GL(n,\Re) \to \Re$ *exists.*

Proof. The Fréchet derivative of $\Phi_N : GL(n,\Re) \to \Re$ at T is the linear map defined by

$$\begin{aligned} D\Phi_N|_T(\xi) &= 2\operatorname{tr}(N\xi W_cT' - N(T')^{-1}W_oT^{-1}\xi T^{-1}) \\ &= 2\operatorname{tr}[(NTW_c - (T')^{-1}W_oT^{-1}N(T')^{-1})'\xi], \end{aligned} \tag{2.20}$$

and therefore the gradient of $\Phi_N(T)$ with respect to the Riemannian metric (2.12) on $GL(n,\Re)$ is

$$\nabla\,\Phi_N(T) = NTW_c - (T')^{-1}W_oT^{-1}N(T')^{-1}. \tag{2.21}$$

It follows that $T \in GL(n,\Re)$ is a critical point of Φ_N if and only if $\nabla\Phi_N(T) = 0$, i.e., if and only if

$$NTW_cT' = (T')^{-1}W_oT^{-1}\cdot N. \tag{2.22}$$

By symmetry of TW_cT' and $(T')^{-1}W_oT^{-1}$, we obtain from (2.22) that

$$N^2TW_cT' = TW_cT'N^2, \tag{2.23a}$$

$$N^2(T')^{-1}W_oT^{-1} = (T')^{-1}W_oT^{-1}N^2. \tag{2.23b}$$

Any symmetric matrix that commutes with N^2 must be diagonal, since N^2 has distinct eigenvalues. Thus we see that (2.22) is equivalent to $TW_cT' = (T')^{-1}W_oT^{-1} =$ diagonal. This proves (a). For (b) note that $\Phi_N(T) \le a$ implies

$$||N^{1/2}TW_c^{1/2}||^2 \le a, \qquad ||W_o^{1/2}T^{-1}N^{1/2}||^2 \le a$$

for the Frobenius norm $||X||^2 = \operatorname{tr}(XX')$. Hence $||T|| \le c_1, ||T^{-1}|| \le c_2$ for positive constants c_1, c_2 that depend only on N, W_c, W_o and a. Thus $\{T \in GL(n,\Re) \mid \Phi_N(T) \le a\}$ is a closed subset of the compact set $\{T \in GL(n,\Re) \mid ||T|| \le c_1,\ ||T^{-1}|| \le c_2\}$ and therefore also compact. This shows that $\Phi_N : GL(n,\Re) \to \Re$ has compact sublevel sets. But any continuous function $f : GL(n,\Re) \to \Re_+$ with compact sublevel sets has a minimizing $T \in GL(n,\Re)$. This completes the proof of Lemma 2.6. □

From Lemma 2.6 and by (2.20), similar arguments as for Theorem 2.4(a) and (b) show the following theorem.

THEOREM 2.7. *Let* W_c, W_o *be the controllability and observability gramians of the asymptotically stable, controllable, and observable realization* (A,B,C) *and let* $N = \operatorname{diag}(\lambda_1,\dots,\lambda_n)$ *with* $\lambda_1 > \cdots > \lambda_n > 0$. *Then*

(a) *gradient flow* $\dot{T} = -\nabla \Phi_N(T)$ *of the weighted cost function* Φ_N: $GL(n, \Re) \to \Re$ *is*

$$\dot{T} = (T')^{-1} W_o T^{-1} N (T')^{-1} - NTW_c. \tag{2.24}$$

For all initial conditions $T(0) \in GL(n, \Re)$, *the solution* $T(t) \in GL(n, \Re)$ *of* (2.24) *exists for all* $t \geq 0$.

(b) *For any initial condition* $T(0) \in GL(n, \Re)$, *the solution* $T(t)$ *of* (2.24) *converges to a diagonal balancing transformation* T_∞ *of* (A, B, C).

(c) *Suppose that the singular values* $0 < d_1 < \cdots < d_n$ *of the Hankel operator of* (A, B, C) *are distinct. Then the stable equilibrium of* (2.24) *are characterized by* $(T'_\infty)^{-1} W_o T_\infty^{-1} = T_\infty W_c T'_\infty = D$, *where* $D = diag(d_1, \ldots, d_n)$ *is diagonal and the diagonal entries are in reverse ordering to those of* N. *Moreover, the gradient flow* (2.24) *converges exponentially fast to the* 2^n *stable equilibria with a convergence rate lower bounded by*

$$\lambda_{\min}((T_\infty T'_\infty)^{-1}) \min_{i<j} [(d_i - d_j)(\lambda_j - \lambda_i), 4 d_i \lambda_i].$$

All other equilibria are unstable.

Proof. Parts (a) and (b) follow easily from Lemma 2.6, using similar arguments as for Theorem 2.4. To prove (c), consider the linearization of (2.24) at an equilibrium point T_∞; that is, where $(T'_\infty)^{-1} W_o T_\infty^{-1} = T_\infty W_c T'_\infty = D$ and

$$\begin{aligned} \dot{\eta} = &-N\eta T_\infty^{-1} D (T'_\infty)^{-1} - D\eta T_\infty^{-1} N (T'_\infty)^{-1} \\ &- (T'_\infty)^{-1} \eta' D N (T'_\infty)^{-1} - D N (T'_\infty)^{-1} \eta' (T'_\infty)^{-1}. \end{aligned}$$

Let $\zeta = \eta T_\infty^{-1}$, then

$$\dot{\zeta}(T_\infty T'_\infty) = -N\zeta D - D\zeta N - \zeta' D N - D N \zeta'$$

and thus, using Kronecker products and the vec notation, and recalling that $\mathrm{vec}(ABC) = (C' \otimes A)\mathrm{vec}(B)$, then

$$[(T_\infty T'_\infty) \otimes I]\mathrm{vec}(\dot{\zeta}) = -[D \otimes N + N \otimes D]\mathrm{vec}(\zeta) - [DN \otimes I + I \otimes DN]\mathrm{vec}(\zeta').$$

Consider first the special case when $T_\infty T'_\infty = I$, and ζ is denoted ζ^*:

$$\mathrm{vec}(\dot{\zeta}^*) = -[D \otimes N + N \otimes D]\mathrm{vec}(\zeta^*) - [DN \otimes I + I \otimes DN]\mathrm{vec}(\zeta^{*\prime}). \tag{2.25}$$

Then for $i < j$,

$$\begin{bmatrix} \dot{\zeta}^*_{ij} \\ \dot{\zeta}^*_{ji} \end{bmatrix} = - \begin{bmatrix} d_i\lambda_j + \lambda_i d_j & d_j\lambda_j + d_i\lambda_i \\ d_i\lambda_i + d_j\lambda_j & d_i\lambda_j + \lambda_i d_j \end{bmatrix} \begin{bmatrix} \zeta^*_{ij} \\ \zeta^*_{ji} \end{bmatrix},$$

and for all i,

$$\dot{\zeta}^*_{ii} = -4 d_i \lambda_i \zeta^*_{ii}.$$

By assumption, $\lambda_i > 0$, and $d_i > 0$ for all i. Thus (2.25) is exponentially stable if and only if $(d_i - d_j)(\lambda_j - \lambda_i) > 0$ for all $i, j, i < j$, or equivalently, if and only if the diagonal entries of D are distinct and in reverse ordering to those of N. In this case, (2.25) is equivalent to (2.26)

$$\mathrm{vec}(\dot{\zeta}^*) = -\mathcal{F}\mathrm{vec}(\zeta^*) \tag{2.26}$$

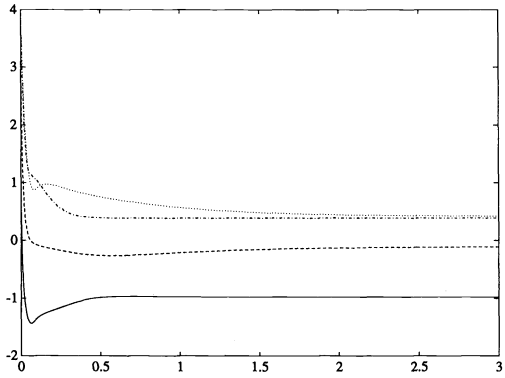


FIG. 3. *Evolution of the diagonalizing transformation T.*

with a symmetric positive definite matrix $\mathcal{F} = \mathcal{F}' > 0$.

Consequently, there is exponential convergence with a rate given by $\lambda_{\min}(\mathcal{F})$ as follows:

$$\begin{aligned}\lambda_{\min}(\mathcal{F}) &= \min\left(\min_{i<j}\left[\lambda_{\min}\left(\begin{bmatrix} d_i\lambda_j + \lambda_i d_j & d_j\lambda_j + d_i\lambda_i \\ d_i\lambda_i + d_j\lambda_j & d_i\lambda_j + \lambda_i d_j \end{bmatrix}\right)\right], \min_i[4d_i\lambda_i]\right) \\ &= \min(\min_{i<j}[d_i\lambda_j + \lambda_i d_j - d_j\lambda_j - d_i\lambda_i], \min_i[4d_i\lambda_i]) \\ &= \min(\min_{i<j}[(d_i - d_j)(\lambda_j - \lambda_i)], \min_i[4d_i\lambda_i]).\end{aligned}$$

Relaxing the assumption $T_\infty T_\infty' = I$ is possible since $T_\infty T_\infty'$ is positive definite so that $(T_\infty T_\infty' \otimes I)$ is positive definite. Thus exponential stability of (2.26) assures exponential stability of

$$[(T_\infty T_\infty') \otimes I]\mathrm{vec}(\dot{\zeta}) = -\mathcal{F}\mathrm{vec}(\zeta).$$

The rate of exponential convergence is given by $\lambda_{\min}[((T_\infty T_\infty')^{-1} \otimes I)\mathcal{F}]$. Now since $A = A' > 0, B = B' > 0$ implies $\lambda_{\min}(AB) \geq \lambda_{\min}(A)\lambda_{\min}(B)$, a lower bound on the convergence rate is given from

$$\begin{aligned}&\lambda_{\min}[((T_\infty T_\infty')^{-1} \otimes I)\mathcal{F}] \geq \lambda_{\min}[(T_\infty T_\infty')^{-1} \otimes I]\lambda_{\min}(\mathcal{F}) \\ &= \lambda_{\min}[(T_\infty T_\infty')^{-1}] \min(\min_{i<j}[(d_i - d_j)(\lambda_j - \lambda_i)], \min_i[4d_i\lambda_i])\end{aligned}$$

as claimed. □

Simulation. In Fig. 3 the diagonal elements of $T(t)$ are plotted. The flow (2.24) is allowed to evolve with $W_o = W_1, W_c = W_2$, $N = \mathrm{diag}(5, 4, 3, 2)$, and initial matrix T_0 as before. At $t = 3$,

$$T(t) = \begin{bmatrix} -0.9788 & 0.6595 & -0.1033 & 0.6623 \\ -0.0807 & -0.1124 & -0.7002 & 0.5847 \\ 0.1079 & -0.4691 & 0.4256 & 0.1943 \\ 0.7586 & 0.5177 & 0.4245 & 0.3909 \end{bmatrix}$$

and this transformation gives

$$W_c = \begin{bmatrix} 0.4554 & 0.0017 & 0.0006 & -0.0005 \\ 0.0017 & 2.7493 & 0.0034 & -0.0000 \\ 0.0006 & 0.0034 & 3.3641 & -0.0034 \\ -0.0005 & -0.0000 & -0.0034 & 11.3114 \end{bmatrix},$$

$$W_o = \begin{bmatrix} 0.4553 & 0.0021 & 0.0009 & 0.0014 \\ 0.0021 & 2.7496 & 0.0042 & 0.0007 \\ 0.0009 & 0.0042 & 3.3625 & 0.0081 \\ 0.0014 & 0.0007 & 0.0081 & 11.3090 \end{bmatrix}.$$

Notice that although convergence has not been completed, the gramians are diagonally dominant with increasing elements.

3. Differential equations for balanced realizations. In this section we construct certain ordinary differential equations

$$\begin{aligned} \dot{A} &= f(A,B,C) \\ \dot{B} &= g(A,B,C) \\ \dot{C} &= h(A,B,C) \end{aligned}$$

evolving on the space of all realizations (A,B,C) of a given transfer function $G(s)$, with the property that their solutions $(A(t),B(t),C(t))$ all converge for $t \to \infty$ to balanced realizations $(\bar{A},\bar{B},\bar{C})$ of $G(s)$.

Let $G(s) \in \Re(s)^{p\times m}$ denote an asymptotically stable strictly proper real rational transfer function of McMillan degree n. Thus $G(s)$ has its poles either in the open left half-plane or in the open unit disc, respectively. We denote by $(A,B,C) \in \Re^{n\times(n+m+p)}$ an asymptotically stable, controllable, and observable realization of $G(s)$, i.e., $G(s) = C(sI-A)^{-1}B$.

Let

$$\mathcal{R}_G = \{(A,B,C) \in \Re^{n\times(n+m+p)} \mid G(s) = C(sI-A)^{-1}B\} \tag{3.1}$$

denote the set of all minimal state space realizations of the transfer function $G(s)$. By Kalman's realization theorem, [8]

$$\mathcal{R}_G = \{(TAT^{-1},TB,CT^{-1}) \in \Re^{n\times(n+m+p)} \mid T \in GL(n,\Re)\} \tag{3.2}$$

for any fixed initial realization $(A,B,C) \in \mathcal{R}_G$. Thus $\mathcal{R}_G$ is an orbit of the $GL(n,\Re)$-similarity action $(A,B,C) \mapsto (TAT^{-1},TB,CT^{-1})$ on $\Re^{n\times(n+m+p)}$.

We consider the function

$$\Phi: \ \mathcal{R}_G \to \Re$$

defined by

$$\Phi(A,B,C) = \operatorname{tr}(W_c(A,B) + W_o(A,C)), \tag{3.3}$$

i.e., by the sum of the eigenvalues of the controllability and observability gramians of (A,B,C). The following proposition summarizes some important properties of $\mathcal{R}_G$ and $\Phi: \mathcal{R}_G \to \Re$.

PROPOSITION 3.1. *It holds that*

(a) *$\mathcal{R}_G$ is a smooth, closed submanifold of $\Re^{n\times(n+m+p)}$. The tangent space of $\mathcal{R}_G$ at $(A,B,C) \in \mathcal{R}_G$ is*

$$T_{(A,B,C)}\mathcal{R}_G = \{(XA - AX, XB, -CX) \mid X \in \Re^{n\times n}\}. \tag{3.4}$$

(b) *The function $\Phi : \mathcal{R}_G \to \Re$ defined by* (3.3) *is smooth and has compact sublevel sets.*

Proof. $\mathcal{R}_G$ is an orbit of the $GL(n,\Re)$-similarity action

$$\begin{aligned}\sigma : GL(n,\Re) \times \Re^{n\times(n+m+p)} &\to \Re^{n\times(n+m+p)}\\ (T,(A,B,C)) &\mapsto (TAT^{-1}, TB, CT^{-1})\end{aligned}$$

and thus, by a general result about algebraic Lie group actions (see, e.g., Appendix C in [9]) is a smooth submanifold of the Euclidean space $\Re^{n\times(n+m+p)}$. By Lemma 3.3 [2], $\mathcal{R}_G$ is a closed subset of $\Re^{n(n+m+p)}$ if (A,B,C) is controllable and observable. Explicitly, by realization theory, $\mathcal{R}_G$ is a fiber of the continuous map

$$\begin{aligned}f : \Re^{n(n+m+p)} &\to \prod_{i=0}^{\infty} \Re^{p\times m}\\ (F,G,H) &\mapsto (HF^iG \mid i \in \mathbb{N}_o)\end{aligned} \tag{3.5}$$

and therefore closed.

To prove (b) and (3.4), we consider the diffeomorphism

$$\begin{aligned}\sigma : GL(n,\Re) &\to \mathcal{R}_G\\ T &\mapsto (TAT^{-1}, TB, CT^{-1})\end{aligned} \tag{3.6}$$

(this requires that (A,B,C) is minimal). The derivative of σ at the identity matrix is the linear map $X \mapsto (XA - AX, XB, -CX)$, which maps $\Re^{n\times n}$ onto $T_{(A,B,C)}\mathcal{R}_G$. This proves (3.4). Furthermore, with $P = T'T$,

$$\begin{aligned}\Phi(\sigma(T)) &= \operatorname{tr}(TW_c(A,B)T' + (T')^{-1}W_o(A,C)T^{-1})\\ &= \operatorname{tr}(W_cP + W_oP^{-1}),\end{aligned}$$

and the result now follows from Lemma 2.1, i.e., that the function $P \mapsto \operatorname{tr}(W_cP + W_oP^{-1})$ on the set of positive definite symmetric matrices has compact sublevel sets. □

We now address the issue of finding gradient flows for the objective function $\Phi : \mathcal{R}_G \to \Re$ relative to some Riemannian metric on $\mathcal{R}_G$. While there are several possible choices for a Riemannian metric on the realization space $\mathcal{R}_G$, the following one leads to a particularly simple expression for the gradient.

In the sequel, we use the Lie bracket notation

$$[A,B] = AB - BA \tag{3.7}$$

for $n \times n$ matrices A, B.

Given two tangent vectors $([X_1,A], X_1B, -CX_1)$ and $([X_2,A], X_2B, -CX_2) \in T_{(A,B,C)}\mathcal{R}_G$ we define

$$\langle\langle([X_1,A], X_1B, -CX_1), \quad ([X_2,A], X_2B, -CX_2)\rangle\rangle := \operatorname{tr}(X_1'X_2). \tag{3.8}$$

To prove that (3.8) defines an inner product on $T_{(A,B,C)}\mathcal{R}_G$, we need the following lemma.

LEMMA 3.2. *Let (A,B,C) be controllable or observable. Then $([X,A],XB,-CX) = (0,0,0)$ implies $X=0$.*

Proof. If $XB=0$ and $AX=XA$, then $X(B,AB,\dots,A^{n-1}B)=0$. Thus controllability implies $X=0$. This is also true for observability. □

It is now easily seen, using Lemma 3.2, that (3.8) defines a nondegenerate symmetric bilinear form on each tangent space $T_{(A,B,C)}\mathcal{R}_G$ and in fact a Riemannian metric on $\mathcal{R}_G$. We refer to this as the *normal Riemannian metric* on $\mathcal{R}_G$.

To determine the gradient flow of $\Phi : \mathcal{R}_G \to \Re$ with respect to the normal Riemannian metric, we need a lemma.

LEMMA 3.3. *Let $N \in \Re^{n\times n}$ be a real symmetric $n\times n$ matrix and let $\Phi_N : \mathcal{R}_G \to \Re$ be defined by $\Phi_N(A,B,C) = \operatorname{tr}(NW_c(A,B) + NW_o(A,C))$ for all $(A,B,C) \in \mathcal{R}_G$. Then the Fréchet derivative of Φ_N at $(A,B,C) \in \mathcal{R}_G$ is the linear map $D\Phi_N(A,B,C) : T_{(A,B,C)}\mathcal{R}_G \to \Re$ defined by*

$$D\Phi_N(A,B,C)([X,A],XB,-CX) = 2\operatorname{tr}[(W_c(A,B)N - NW_o(A,C))X] \tag{3.9}$$

for $X \in \Re^{n\times n}$.

Proof. Let $\sigma : GL(n,\Re) \to \mathcal{R}_G$ be the diffeomorphism defined by $\sigma(T) = (TAT^{-1}, TB, CT^{-1})$. The derivative of σ at the identity matrix I_n is the linear map $X \mapsto ([X,A],XB,-CX)$ on $\Re^{n\times n}$. By the chain rule for the composed map $\Phi_N \circ \sigma$ defined by

$$\Phi_N(\sigma(T)) = \operatorname{tr}(NTW_c(A,B)T' + N(T')^{-1}W_o(A,C)T^{-1}),$$

we have

$$\begin{aligned} D\Phi_N(\sigma(I_n))([X,A],-XB,CX) &= D(\Phi_N \circ \sigma)(I_n)X \\ &= 2\operatorname{tr}(NXW_c(A,B) - NW_o(A,C)X) \\ &= 2\operatorname{tr}[(W_c(A,B)N - NW_o(A,C))X] \end{aligned}$$

for all $X \in \Re^{n\times n}$. The result follows. □

THEOREM 3.4. *Let $\Phi : \mathcal{R}_G \to \Re$ be the objective function defined by $\Phi(A,B,C) = \frac{1}{2}\operatorname{tr}(W_c(A,B) + W_o(A,C))$.*

(a) *The gradient flow $(\dot{A} = -\operatorname{grad}_A\Phi(A,B,C), \dot{B} = -\operatorname{grad}_B\Phi(A,B,C), \dot{C} = -\operatorname{grad}_A\Phi(A,B,C))$ of Φ for the normal Riemannian metric on $\mathcal{R}_G$ is*

$$\begin{aligned} \dot{A} &= [A, W_o(A,C) - W_c(A,B)] \\ \dot{B} &= (W_o(A,C) - W_c(A,B))B \\ \dot{C} &= C(W_c(A,B) - W_o(A,C)). \end{aligned} \tag{3.10}$$

For every initial condition $(A(0),B(0),C(0)) \in \mathcal{R}_G$, the solution $(A(t),B(t),C(t)) \in \mathcal{R}_G$ of (3.10) exists for all $t \geq 0$ and converges for $t \to +\infty$ to a balanced realization $(\bar{A},\bar{B},\bar{C})$ of $G(s)$:

$$W_c(\bar{A},\bar{B}) = W_o(\bar{A},\bar{C}).$$

(b) *Convergence to the class of balanced realizations is exponentially fast.*

(c) *The transfer function of any solution $(A(t),B(t),C(t))$ of (3.10) is independent of t.*

Proof. By definition of a gradient,

$$\text{grad}\Phi(A,B,C) = (\text{grad}_A\Phi(A,B,C), \text{grad}_B\Phi(A,B,C), \text{grad}_C\Phi(A,B,C))$$

is characterized by the properties (see the Appendix)

$$\text{(3.11a)} \qquad \text{grad}\Phi(A,B,C) \in T_{(A,B,C)}\mathcal{R}_G,$$

and

$$\text{(3.11b)} \quad D\Phi(A,B,C)([X,A],XB,-CX) = \langle\langle \text{grad}\Phi(A,B,C), ([X,A],XB,-CX)\rangle\rangle$$

for all $X \in \Re^{n\times n}$. By Proposition 3.1 and Lemma 3.2,

$$\text{(3.12)} \qquad \text{grad}\Phi(A,B,C) = ([A,\Lambda],\Lambda B,-C\Lambda)$$

for a uniquely determined $\Lambda \in \Re^{n\times n}$. Applying Lemma 3.3 for $N = \frac{1}{2}I_n$, we see that (3.11b) is equivalent to

$$\begin{aligned}\text{tr}[(W_c(A,B) - W_o(A,C))X] &= \langle\langle([A,\Lambda],\Lambda B,-C\Lambda),\ ([A,X],XB,-CX)\rangle\rangle \\ &= \text{tr}(\Lambda' X)\end{aligned}$$

for all $X \in \Re^{n\times n}$. Thus

$$\Lambda = W_c(A,B) - W_o(A,C)$$

and $\text{grad}\Phi(A,B,C) = ([A,\Lambda],\Lambda B,-C\Lambda)$. This proves (3.10). Since (3.10) is minus the gradient flow of Φ, $\Phi(A(t),B(t),C(t))$ decreases on any solution of (3.10). By Proposition 3.1(b), $\{(A,B,C) \in \mathcal{R}_G \mid \Phi(A,B,C) \le \Phi(A(0),B(0),C(0))\}$ is a compact subset of $\mathcal{R}_G$, which is invariant under the flow of (3.10). Therefore $(A(t),B(t),C(t))$ stays in that compact subset and thus exists for all $t \ge 0$. The equilibria of (3.10) are characterized by $W_c(A,B) = W_o(A,C)$, i.e., by the balanced realizations. This proves (a), except that we have not yet established convergence to an equilibrium point.

To prove (b), we consider the diffeomorphism $\sigma : GL(n,\Re) \to \mathcal{R}_G$ defined by $\sigma(T) = (TAT^{-1}, TB, CT^{-1})$ for any $(A,B,C) \in \mathcal{R}_G$. At each critical point, (A,B,C) of $\Phi : \mathcal{R}_G \to \Re$, σ induces an invertible congruence transformation between the Hessian of $\Phi : \mathcal{R}_G \to \Re$ at (A,B,C) and the Hessian of $\Phi \circ \sigma$ at I_n. By (2.16) and the proof of Theorem 2.4(c), the Hessian of $\Phi \circ \sigma$ at I_n is positive semidefinite and degenerates exactly on the tangent space (at I_n) of the set of balancing transformations. Therefore the Hessian of Φ at a balanced realization (A,B,C) is positive semidefinite and degenerates exactly on the tangent space of the set of balanced realizations at (A,B,C). (N.B. By Lemma 2.5, the set of balanced realizations of $G(s)$ can be shown to be a smooth submanifold of $\mathcal{R}_G$.) This proves (b). As $\Phi : \mathcal{R}_G \to \Re$ is now seen as a Morse–Bott function, we can apply Proposition A.3 to conclude that $(A(t),B(t),C(t))$ converges to an equilibrium point.

Part (c) is obvious, as the flow evolves on $\mathcal{R}_G$. □

We emphasize that Theorem 3.4 gives, for the first time, a *direct* method to compute balanced realizations, without computing any balancing transformations. We regard this as one of the really new insights that can be obtained by our ODE methods.

Remark. As is shown in the above proof, any flow on symmetric matrices

$$\begin{aligned}\dot{A} &= -[A,\Lambda(A,B,C)] \\ \dot{B} &= -\Lambda(A,B,C)B \\ \dot{C} &= +C\Lambda(A,B,C),\end{aligned}$$

where $\Lambda(A,B,C) \in \Re^{n\times n}$ is an arbitrary matrix valued function of (A,B,C), leaves the transfer function

$$G(t,s) = C(t)(sI_n - A(t))^{-1}B(t) = C(0)(sI_n - A(0))^{-1}B(0)$$

of the system invariant. We therefore term these flows *isodynamical* and a more systematic analysis of such flows is given in [9]. Obviously, these flows leave the eigenvalues of $A(t)$ invariant and in fact generalize the class of isospectral flows on matrices, obtained by letting $B = 0, C = 0$; see, e.g., [1], [3], and the references therein.

Simulations. Figures 4(a)–(c) show the evolution of the system matrices (A,B,C) using this algorithm. In this example, the starting matrices are chosen to be

$$A = \begin{bmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}, \quad C' = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}, \tag{3.13}$$

and after ten "time intervals" the gramians are equal to three significant figures.

A similar "isodynamical flow approach" works also for obtaining diagonal balanced realizations. Here we consider the weighted cost function

$$\Phi_N : \mathcal{R}_G \to \Re,$$

$$\Phi_N(A,B,C) = \frac{1}{2}\mathrm{tr}[N(W_c(A,B) + W_o(A,C))] \tag{3.14}$$

for a real diagonal matrix N.

THEOREM 3.5. *Let $\Phi_N : \mathcal{R}_G \to \Re$ be the objective function defined by* (3.14) *for $N = \frac{1}{2}\mathrm{diag}(\lambda_1,\dots,\lambda_n), \lambda_1 > \cdots > \lambda_n > 0$.*

(a) *The gradient flow*

$$(\dot{A} = -\mathrm{grad}_A\Phi_N(A,B,C), \dot{B} = -\mathrm{grad}_B\Phi_N(A,B,C), \dot{C} = -\mathrm{grad}_C\Phi_N(A,B,C))$$

of Φ_N with respect to the normal Riemannian metric on $\mathcal{R}_G$ is

$$\begin{aligned} \dot{A} &= [A, NW_o(A,C) - W_c(A,B)N] \\ \dot{B} &= (NW_o(A,C) - W_c(A,B)N)B \\ \dot{C} &= C(W_c(A,B)N - NW_o(A,C)). \end{aligned} \tag{3.15}$$

For every initial condition $(A(0),B(0),C(0)) \in \mathcal{R}_G$, the solution $(A(t),B(t),C(t)) \in \mathcal{R}_G$ of (3.15) *exists for all $t \geq 0$ and converges for $t \to +\infty$ to a diagonal balanced realization $(\bar{A},\bar{B},\bar{C})$ of $G(s)$, i.e., $W_c(\bar{A},\bar{B}) = W_o(\bar{A},\bar{C}) =$* diagonal.

(b) *Suppose that the singular values of the Hankel of (A,B,C) are distinct. Then* (3.15) *has exactly 2^n locally asymptotically stable equilibrium points $(\bar{A},\bar{B},\bar{C})$, characterized by $W_c(\bar{A},\bar{B}) = W_o(\bar{A},\bar{C})$=diagonal, with the diagonal elements in the reverse order to that of N. All other equilibria are unstable.*

(c) *The transfer function of every solution $(A(t),B(t),C(t))$ of* (3.15) *is independent of t.*

Proof. The proof runs similarly to that of Theorem 3.4, now applying Lemma 3.3 for $N = \frac{1}{2}\mathrm{diag}(\lambda_1,\dots,\lambda_n)$ and using Proposition A.3. The only points we must check are that the equilibria of (3.15) are just the diagonal balanced realizations and

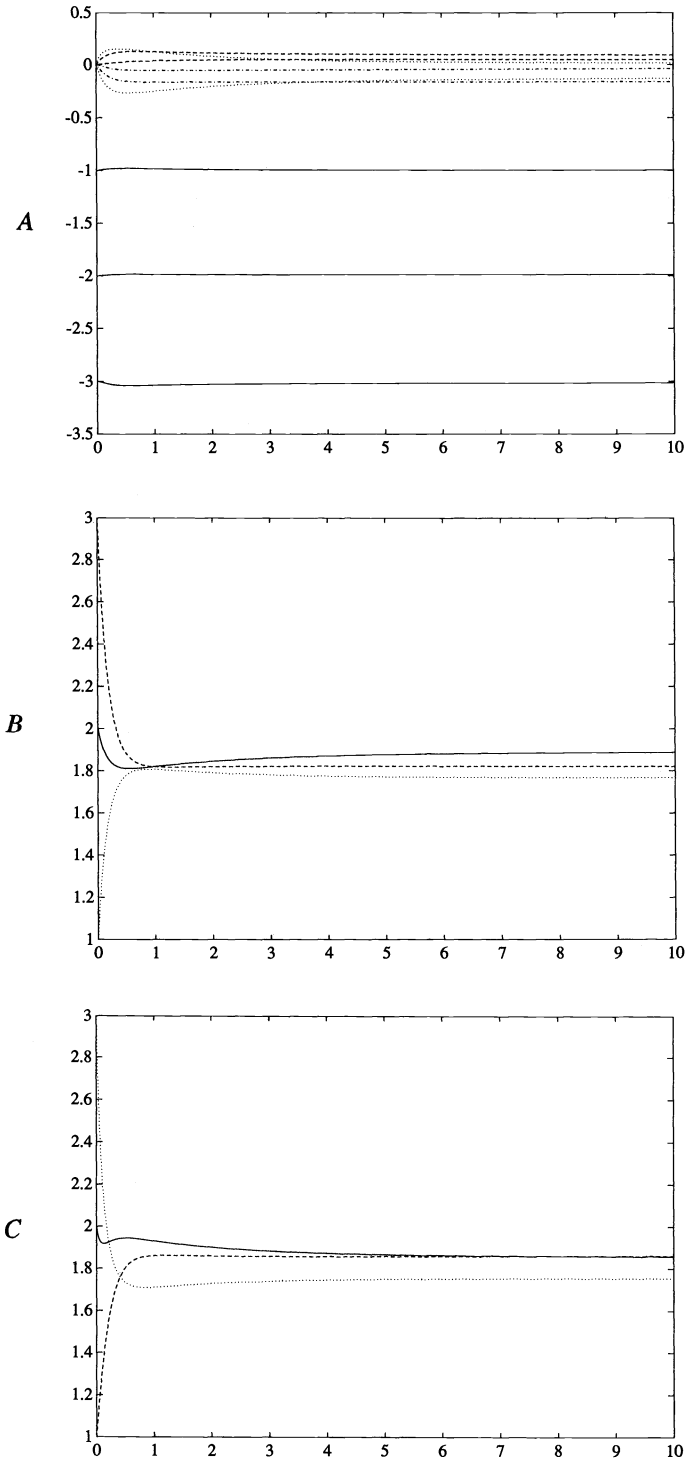


FIG. 4. *Evolution of the system matrices* (A, B, C).

their stability. But the equilibria of (3.15) are the critical points of $\Phi_N : \mathcal{R}_G \to \Re$ and hence correspond to those of $\Phi_N \circ \sigma : GL(n, \Re) \to \Re$. The result now follows from Lemma 2.6, and Theorem 2.7. □

Simulations. Figure 5 shows the evolution of the matrices (A, B, C) for (3.15), with starting condition given in (3.13), and $N = \text{diag}(3, 2, 1)$. After 30 "time intervals" the solution gives

$$W_o = \begin{bmatrix} 2.7720 & 0 & 0 \\ 0 & 0.1367 & 0.0214 \\ 0 & 0.0214 & 0.0048 \end{bmatrix}, \qquad W_c = \begin{bmatrix} 2.7750 & 0 & 0 \\ 0 & 0.1367 & 0.0212 \\ 0 & 0.0212 & 0.0067 \end{bmatrix}$$

as opposed to the true balanced solution $W_o = W_c = \text{diag}(0.0021, 0.1401, 2.7744)$. The convergence in this case can be expected to be slow because the smallest Hankel singular value is near zero.

4. Application to SVD. The common linear algebra problem of SVD can be solved using differential equations. Gradient flow solutions for SVD have been studied in [3], [4], [6], and [7]. Here we consider SVD to be a special case of the balanced realization task.

THEOREM 4.1. *Given an $m \times n$ matrix H of rank r with distinct singular values $\sigma_1 > \cdots > \sigma_r$. Let N be an $r \times r$ diagonal matrix with distinct positive diagonal entries. Let $X_0 \in \Re^{m \times r}$ and $Y_0 \in \Re^{r \times n}$ be matrices of full rank r such that $H = X_0 Y_0$. Then the solution $(X(t), Y(t))$ of*

$$\dot{X} = X(NYY' - X'XN), \qquad X(0) = X_0, \tag{4.1a}$$

$$\dot{Y} = (X'XN - NYY')Y, \qquad Y(0) = Y_0 \tag{4.1b}$$

exists for all $t \geq 0$ and satisfies $H = X(t)Y(t)$ for all $t \geq 0$. The solution $(X(t), Y(t))$ converges to (X_∞, Y_∞) such that $H = X_\infty Y_\infty$ and $X'_\infty X_\infty = Y_\infty Y'_\infty = D = $ diagonal. Moreover, there are 2^r stable equilibria that have the diagonal elements of D in reverse order to those of N. All other equilibrium points are unstable.

Furthermore, this factorization yields $H = USV$, where $U = X_\infty D^{-1/2}$, $S = D$, $V = D^{-1/2} Y_\infty$, $U'U = I$, $VV' = I$.

Proof. In Theorem 3.5, set $A = 0$ and let B, C be full rank matrices. Clearly, (A, B, C) is controllable and observable. Then $W_c = B'B, W_o = CC'$ and (3.15) is equivalent to

$$\dot{B} = -(CC'N - NB'B)B, \qquad \dot{C} = C(CC'N - NB'B). \tag{4.2}$$

The equilibria of (4.2) are characterized by $B'B = CC' =$ diagonal, and the stable equilibria are such that $B'B$ is in reverse order to N.

As (4.2) preserves the transfer function, CB is constant. Hence (4.1) with $X = C$ and $Y = B$ converges to a diagonal balanced matrix factorization $H = X_\infty Y_\infty$. By choosing $U = X_\infty D^{-1/2}, S = D, V = D^{-1/2} Y_\infty$, then

$$USV = X_\infty D^{-1/2} D D^{-1/2} Y_\infty = X_\infty Y_\infty = H, \qquad U'U = D^{-1/2} X'_\infty X_\infty D^{-1/2} = I,$$

and $VV' = D^{-1/2} Y_\infty Y'_\infty D^{-1/2} = I$. The full singular value decomposition can be obtained by extending U and V to make them orthogonal. □

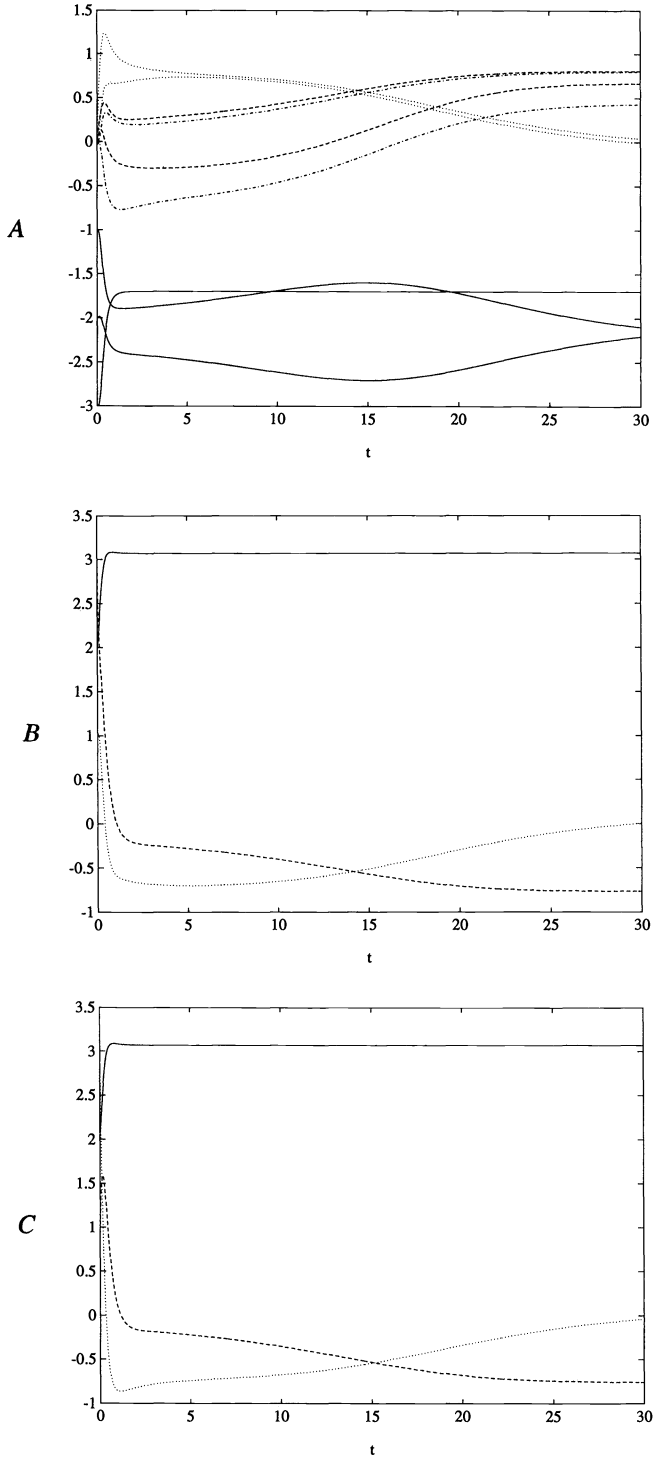


FIG. 5. *Evolution of the diagonalizing system matrices* (A, B, C).

5. Conclusions. There are a number of distinct ODEs that evolve to give the solution to the task of finding a balanced realization of a system or to the task of finding the SVD of a matrix. Each differential equation has distinct transient behaviour but all have exponential convergence rates of the factors. The dynamical systems for balancing gramains that were investigated evolve on either spaces of state coordinate transformation matrices T, its square $P = T'T$, or on manifolds of the actual system matrices (A, B, C). Similar equations for SVD are studied and are a special case of the balancing equations. Different convergence properties make some algorithms more attractive in certain problem settings. These solution methods may be useful when using analog or parallel computers.

Appendix. Riemannian metrics and gradient flows. Let M be a smooth manifold and let TM and T^*M denote its tangent and cotangent bundle, respectively. A Riemannian metric on M is a family of nondegenerate inner products $<,>_x$, defined on each tangent space T_xM, such that $<,>_x$ depends smoothly on $x \in M$. Any (nondegenerate) inner product on $\Re^n$ also defines a Riemannian metric on $\Re^n$ (but not conversely) and thus induces a Riemannian metric on every submanifold M of $\Re^n$.

Let $\Phi : M \to \Re$ be a smooth function defined on the manifold M and let $D\Phi : M \to T^*M$ denote the differential, i.e., a section of the cotangent bundle T^*M. To define the gradient vector field of Φ, we fix a Riemannian metric $<,>$ on M. The gradient $\bigtriangledown\Phi$ of Φ is then characterized by the following properties:

Compatibility condition (a). $D\Phi(x)\xi = \langle \bigtriangledown\Phi(x), \xi \rangle$ for all $\xi \in T_xM$.

Tangency condition (b). $\bigtriangledown\Phi(x) \in T_xM$ for all $x \in M$.

The following result is well known.

PROPOSITION A.1. *There exists a uniquely determined vector field* $\bigtriangledown\Phi$ *on* M *such that* (a) *and* (b) *hold.* $\bigtriangledown\Phi$ *is called the gradient vector field of* Φ.

Note that the gradient vector field depends on the choice of the Riemannian metric; changing the metric will also change the gradient.

It follows immediately from the definition of $\bigtriangledown\Phi$ that the equilibria of the differential equation

$$\dot{x}(t) = - \bigtriangledown \Phi(x(t)) \tag{A.1}$$

are precisely the critical points of Φ. Moreover, the linearization of the gradient flow (A.1) around each equilibrium point is given by the Hessian of Φ and thus has only real eigenvalues.

For any solution of (A.1),

$$\frac{d}{dt}\Phi(x(t)) = -|| \bigtriangledown \Phi(x(t))||^2$$

and therefore $\Phi(x(t))$ is monotonically decreasing. The following standard result is often used in this paper.

PROPOSITION A.2. *Let* $\Phi : M \to \Re$ *be a smooth function on a Riemannian manifold with compact sublevel sets, i.e., for all* $c \in \Re$ *the sublevel set* $\{x \in M \mid \Phi(x) \leq c\}$ *is a compact subset of* M. *Then every solution* $x(t) \in M$ *of the gradient flow* (A.1) *on* M *exists for all* $t \geq 0$. *Furthermore,* $x(t)$ *converges to a connected component of the set of critical points of* Φ *as* $t \to +\infty$.

Note that the condition of the proposition is automatically satisfied if M is compact. Moreover, in suitable local coordinates of M, the linearization of the gradient flow (A.1) around each equilibrium point has only real eigenvalues.

Let M be a smooth manifold and let $\Phi : M \to \Re$ be a smooth function. Let $C(\Phi) \subset M$ denote the set of all critical points of Φ. We say Φ is a *Morse–Bott* function provided the following three conditions (i), (ii), (iii) are satisfied.

(i) $\Phi : M \to \Re$ has compact sublevel sets.

(ii) $C(\Phi) = \bigcup_{j=1}^{k} N_j$ with N_j disjoint, closed, and connected submanifolds of M, such that Φ is constant on $N_j, j = 1, \ldots, k$.

(iii) $\mathrm{Ker}(\mathrm{Hess}\Phi)_x = T_x N_j$ for all $x \in N_j, j = 1, \ldots, k$.

Actually, the original definition of a Morse–Bott function also includes a global topological condition on the negative eigenspace bundle defined by the Hessian, but this condition is not relevant to us.

Recall that the *ω-limit set* $L_\omega(x)$ of a point $x \in M$ for a vector field X on M is the set of points of the form $\lim_{n\to\infty} \phi_{t_n}(x)$, where (ϕ_t) is the flow of X and $t_n \to +\infty$. Similarly, the *α-limit set* $L_\alpha(x)$ is defined by letting $t_n \to -\infty$ instead of $+\infty$.

PROPOSITION A.3.

(a) *Suppose $\Phi : M \to \Re$ has isolated critical points. Then $L_\omega(x), x \in M$, consists of a single critical point. Therefore every solution of the gradient flow* (A.1) *converges for $t \to +\infty$ to a critical point of Φ.*

(b) *Let $\Phi : M \to \Re$ be a Morse–Bott function on a Riemannian manifold M. Then the ω-limit set $L_\omega(x)$, $x \in M$, for the gradient flow* (A.1) *is a single critical point of Φ. Every solution of the gradient flow* (A.1) *converges as $t \to +\infty$ to an equilibrium point.*

REFERENCES

[1] R. W. BROCKETT, *Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems*, Linear Algebra Appl., 146 (1991), pp. 79–91.

[2] U. HELMKE, *Balanced realizations for linear systems: a variational approach*, SIAM J. Control Optim., 31 (1993), pp. 1–15.

[3] U. HELMKE AND J. B. MOORE, *Singular value decomposition via gradient and self-equivalent flows*, Linear Algebra Appl., 169 (1992), pp. 223–248.

[4] J. E. PERKINS, U. HELMKE, AND J. B. MOORE, *Balanced realizations via gradient flows*, Systems and Control Letters, 14 (1990), pp. 369–377.

[5] J. IMAE, J. E. PERKINS, AND J. B. MOORE, *Towards time varying balanced realization via Riccati equations*, Math. Control, Signals, Systems, 1992, pp. 313–326.

[6] S. T. SMITH, *Dynamical systems that perform the singular value decomposition*, Systems Control Letters, 16 (1991), pp. 319–328.

[7] M. T. CHU AND K. R. DRIESSEL, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM J. Numer. Anal., 27 (1990), pp. 1050–1060.

[8] T. KAILATH, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[9] U. HELMKE AND J. B. MOORE, *Optimization and Dynamical Systems*, Springer-Verlag, New York, 1993.

[10] B. C. MOORE, *Principal component analysis in linear systems: controllability, observability, and model reduction*, IEEE Trans. Automat. Control, AC-26 (1981), pp. 17–32.

[11] A. J. LAUB, M. T. HEATH, C. C. PAIGE, AND R. C. WARD, *Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms*, IEEE Trans. Automat. Control, AC-32 (1987), pp. 115–121.

[12] M. C. IRWIN, *Smooth Dynamical Systems*, Academic Press, London, 1980.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 755–778, July 1994

004

TRUST REGION PROBLEMS AND NONSYMMETRIC EIGENVALUE PERTURBATIONS*

RONALD J. STERN† AND HENRY WOLKOWICZ‡

Abstract. A characterization is given for the spectrum of a symmetric matrix to remain real after a nonsymmetric sign-restricted border perturbation, including the case where the perturbation is skew-symmetric. The characterization is in terms of the stationary points of a quadratic function on the unit sphere. This yields interlacing relationships between the eigenvalues of the original matrix and those of the perturbed matrix. As a result of the linkage between the perturbation and stationarity problems, new theoretical insights are gained for each. Applications of the main results include a characterization of those matrices that are exponentially nonnegative with respect to the n-dimensional ice-cream cone, which in turn leads to a decomposition theorem for such matrices. In addition, results are obtained for nonsymmetric matrices regarding interlacing and majorization.

Key words. trust region problems, nonsymmetric perturbation, secular function, secular antiderivative, eigenvalues, interlacing, exponential nonnegativity, majorization, inverse eigenvalue problems

AMS subject classification. 15A18

1. Introduction.

Suppose that B is a real symmetric $(n-1)\times(n-1)$ matrix. Then the classical Rayleigh Principle and Courant–Fischer Minimax Theorem relate the eigenvalues of B to the stationary points of the quadratic function

$$\nu(x) = x^t B x$$

with respect to the constraint set

$$S_{n-1} = \{x \in R^{n-1} : x^t x = 1\}.$$

In particular, if we introduce the *Lagrangian function*

$$L(x,\lambda) = \nu(x) - \lambda(x^t x - 1), \tag{1.1}$$

then the *Lagrange equation*

$$\partial_x L(x,\lambda) = 0 \tag{1.2}$$

becomes

$$Bx - \lambda x = 0. \tag{1.3}$$

If x and λ satisfy the Lagrange equation and $x \in S_{n-1}$, then we shall say that λ and x are a *Lagrange multiplier* and an associated *stationary point* of $\nu(\cdot)$ with respect to S_{n-1}, respectively. Thus there is a one-to-one correspondence between the eigenvalues of B and the Lagrange multipliers. Furthermore, the stationary points, including the maximum and minimum points, can be found by determining the unit eigenvectors of

* Received by the editors May 28, 1991; accepted for publication (in revised form) November 16, 1992. This research was supported by Natural Sciences Engineering Research Council Canada grants A4641 and A9161, respectively.

† Department of Mathematics and Statistics, Concordia University, Montreal, Quebec H4B 1R6, Canada (stern@vax2.concordia.ca).

‡ Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (hwolkowi@orion.uwaterloo.ca).

B. The lack of convexity in the constraint does not cause difficulties in locating the constrained maxima and minima, since such points correspond to the maximum and minimum eigenvalues, respectively.

The eigenvalues of symmetric border perturbations of B have well-known properties. In particular, the eigenvalues $\delta_1 \geq \delta_2 \cdots \geq \delta_n$ of the $n \times n$ matrix

$$A = \begin{pmatrix} B & \eta \\ \eta^t & t \end{pmatrix} \tag{1.4}$$

interlace the eigenvalues of B, which we denote $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_{n-1}$. That is,

$$\delta_1 \geq \gamma_1 \geq \delta_2 \geq \gamma_2 \geq \cdots \geq \gamma_{n-1} \geq \delta_n. \tag{1.5}$$

(See, e.g., pp. 94–97 in Wilkinson [24].)

Nonsymmetric border perturbations of B are not as well understood. For example, the $n \times n$ matrix

$$A = \begin{pmatrix} B & -\alpha \\ \alpha^t & t \end{pmatrix}, \tag{1.6}$$

which is a skew-symmetric perturbation for $t = 0$, may possess either a complex or real spectrum, and may be either diagonalizable or derogatory.

On the other hand, the important problem of finding the Lagrange multipliers and stationary points of the general quadratic function

$$\mu(x) = x^t B x - 2\eta^t x \tag{1.7}$$

on S_{n-1} has been extensively studied in the literature. In particular, we shall consider the "trust region" problems

$$P^{\mu}_{\min}: \quad \min\{\mu(x) \ : \ x \in S_{n-1}\}$$

and

$$P^{\mu}_{\max}: \quad \max\{\mu(x) \ : \ x \in S_{n-1}\}.$$

Such problems arise during the calculation of the step between iterates in an important class of minimization algorithms called "trust region methods." (The step in trust region algorithms is actually calculated with a constraint of the form $\| Gy \| \leq \psi$, for some nonsingular matrix G and $\psi > 0$. However, complementary slackness and the change of variables $x = (1/\psi)Gy$ lead to the form of our trust region problems.) The theory has been discussed in Forsythe and Golub [5], Golub [9], Gander [6], Sorensen [21], Fletcher [4] and Gander, Golub, and von Matt [7]. Furthermore, numerical techniques for solving trust region problems are given in [21], Moré and Sorensen [18], [4], Coleman and Hempel [3], [7], and Golub and von Matt [10].

In the present work, we establish new connections between spectral properties of a nonsymmetrically perturbed symmetric matrix and the stationarity properties of a specific trust region problem. We provide explicit criteria for the spectrum of the perturbed matrix to remain real, as well as eigenvalue interlacing properties. We shall consider certain sign-restricted nonsymmetric border perturbations, including the case (1.6). Our approach, in essence, is to regard the perturbation of a matrix as a linear perturbation of a purely quadratic form. As a result of the interplay between

the trust region and perturbation problems, new theoretical insights are gained for each.

In the next section we summarize required known facts concerning trust region problems, some of which involve the so-called *secular function* associated with $\mu(\cdot)$. In addition, we shall make use of the *secular antiderivative function* associated with $\mu(\cdot)$. This is a key tool which we employ to relate results on trust region problems to perturbation theory. The main results are then given in §3, including interlacing relationships for a nonsymmetrically perturbed matrix.

Section 4 contains applications of our main results. These include a characterization of matrices which are exponentially nonnegative with respect to the n-dimensional ice-cream cone, which leads to a decomposition theorem for such matrices. In addition, results are given for nonsymmetric matrices regarding interlacing and majorization.

2. Trust region problems.

2.1. Some known results. For the real symmetric $(n-1)\times(n-1)$ matrix B and the real $(n-1)$-vector η, consider the quadratic function $\mu(\cdot)$ given by (1.7) on S_{n-1}. Then the Lagrangian function is

$$L(x,\lambda) = x^tBx - 2\eta^tx - \lambda(x^tx-1). \tag{2.1}$$

In all that follows, our terminology regarding Lagrange multipliers and stationary points is as in §1, with the appropriate Lagrange equation replacing (1.3). Presently, the Lagrange equation is

$$(B-\lambda I)x - \eta = 0, \tag{2.2}$$

The set of Lagrange multipliers of $\mu(\cdot)$ with respect to S_{n-1} will be denoted by Λ, and for $\lambda\in\Lambda$, the associated set of stationary points will be denoted by $S_\mu(\lambda)$.

Useful properties concerning trust region problems are summarized in the following theorem.

THEOREM 2.1. *Part* 1. *The vector* $x\in R^{n-1}$, *with* $x^tx=1$, *is a minimum (maximum) point of* $\mu(\cdot)$ *over* S_{n-1} *if and only if there exists a scalar* λ *such that* x *and* λ *together satisfy the Lagrange equation* (2.2), *with the matrix* $B-\lambda I$ *being positive (negative) semidefinite.*

Part 2. *The set* Λ *of Lagrange multipliers of* $\mu(\cdot)$ *with respect to* S_{n-1} *is finite. Let* Λ *be given by*

$$\lambda_1 > \lambda_2 > \cdots > \lambda_k,$$

and let $x^{\lambda_i}\in S_\mu(\lambda_i)$, $i=1,2,\ldots,k$. *Then*

$$\mu(x^{\lambda_1}) > \mu(x^{\lambda_2}) > \cdots > \mu(x^{\lambda_k}).$$

In particular, the minimum (maximum) of $\mu(\cdot)$ *over* S_{n-1} *is attained at any stationary point associated with* λ_k (λ_1).

Part 1 of the above theorem is due to Sorensen [21]; see also pages 101–102 in Fletcher [4]. Part 2 is due to Forsythe and Golub [5]; also see the discussion of Case b below.

Again denoting the spectrum of B by $\gamma_1\geq\gamma_2\geq\cdots\geq\gamma_{n-1}$, let P be an orthogonal matrix such that

$$P^tBP = D = \operatorname{diag}(\gamma_1,\gamma_2,\ldots,\gamma_{n-1}), \tag{2.3}$$

the diagonal matrix with diagonal elements $\gamma_1, \gamma_2, \ldots, \gamma_{n-1}$. Then the Lagrange equation (2.2) becomes

$$(D - \lambda I)\hat{x} = \hat{\eta}, \tag{2.4}$$

where

$$\hat{x} = P^t x, \quad \hat{\eta} = P^t \eta.$$

The set of Lagrange multipliers Λ is not changed by this transformation of the Lagrange equation, and for every $x \in R^{n-1}$, we have

$$\hat{\mu}(\hat{x}) := \hat{x}^t D\hat{x} - 2\hat{\eta}^t\hat{x} = \mu(x). \tag{2.5}$$

We now introduce the condition

$$\hat{\eta}_i \neq 0 \quad \forall i = 1, 2, \ldots, n-1, \tag{2.6}$$

or equivalently,

$$\textit{no column of } P \textit{ is orthogonal to } \eta. \tag{2.7}$$

There are two cases to consider.

Case a. Condition (2.6) holds. Then (2.4) implies that if $\lambda \in \Lambda$, it must be the case that $D - \lambda I$ and $B - \lambda I$ are invertible. Also, the sets $S_\mu(\lambda)$ and $S_{\hat{\mu}}(\lambda)$ are then the singletons

$$x^\lambda = (B - \lambda I)^{-1}\eta \tag{2.8}$$

and

$$\hat{x}^\lambda = (D - \lambda I)^{-1}\hat{\eta} = P^t x^\lambda, \tag{2.9}$$

respectively. Furthermore, in the present case, Λ is the set of solutions to the *implicit secular equation*

$$1 - \eta^t (B - \lambda I)^{-2}\eta = 0, \tag{2.10}$$

which has the same solution set as the *explicit secular equation*

$$f_\mu(\lambda) := 1 - \sum_{i=1}^{n-1} \left(\frac{\hat{\eta}_i}{\gamma_i - \lambda}\right)^2 = 0. \tag{2.11}$$

Continuing to utilize the terminology of [7], we shall call $f_\mu(\cdot)$ the *secular function* associated with $\mu(\cdot)$.

In Case a, unique solutions to $P^\mu_{\min}$ and $P^\mu_{\max}$ are given by x^{λ_k} and x^{λ_1}, respectively, from formula (2.8). Furthermore, in view of Part 1 of Theorem 2.1, the invertibility of $B - \lambda_k I$ implies

$$\lambda_k < \gamma_{n-1}, \tag{2.12}$$

while the invertibility of $B - \lambda_1 I$ implies

$$\lambda_1 > \gamma_1. \tag{2.13}$$

Case b. Condition (2.6) does not hold. In [5] it was proven that in this case, $\Lambda = \tilde{\Lambda} \cup \Gamma$, with $\tilde{\Lambda}$ being the solution set of the explicit secular equation, and where

$$\Gamma = \{\gamma_i \ : \ f_\mu(\gamma_i) > 0\},$$

where we adopt the convention $0/0 = 0$ in defining $f_\mu(\cdot)$. For each $\lambda \in \tilde{\Lambda}$, $B - \lambda I$ is invertible and $S_\mu(\lambda)$ is the singleton x^λ given by formula (2.8). For each $\gamma_i \in \Gamma$, the set $S_\mu(\gamma_i)$ is an $(m_{\gamma_i} - 1)$-dimensional manifold, where m_{γ_i} is the multiplicity of the eigenvalue γ_i.

In Case b, it is possible that $\gamma_1 = \lambda_1$, implying that λ_1 occurs strictly to the right of the maximal root of $f_\mu(\cdot)$. Note that this can happen only if

$$\gamma_i = \lambda_1 \Longrightarrow \hat{\eta}_i = 0.$$

Furthermore, then $f_\mu(\lambda_1) > 0$, implying that $S_\mu(\lambda_{n-1})$, the set of solutions to the trust region problem $P^\mu_{\max}$, is not a singleton. Likewise, it is possible that $\gamma_{n-1} = \lambda_k$, implying that λ_k occurs strictly to the left of the minimal root of $f_\mu(\cdot)$. This is possible only if

$$\gamma_i = \lambda_k \Longrightarrow \hat{\eta}_i = 0.$$

It may then happen that $f_\mu(\lambda_k) > 0$, implying that $S_\mu(\lambda_k)$, the set of solutions to the trust region problem $P^\mu_{\min}$, is not a singleton.

2.2. The secular antiderivative. For the general quadratic function $\mu(\cdot)$ given by (1.7), consider the function

$$g_\mu(\lambda) = \lambda - \sum_{i=1}^{n-1} \left(\frac{\hat{\eta}_i^2}{\gamma_i - \lambda} \right), \tag{2.14}$$

with the convention $0/0 = 0$. Then the singularities of $g_\mu(\cdot)$ are the same as those of the secular function $f_\mu(\cdot)$, and what is more,

$$g'_\mu(\lambda) = f_\mu(\lambda) \tag{2.15}$$

at every nonsingularity λ. We shall call $g_\mu(\cdot)$ the *secular antiderivative function* associated with $\mu(\cdot)$.

The following lemma will be used in the next section to establish connections between trust region problems and perturbation theory. The lemma asserts that in Case a, the secular antiderivative's values on the Lagrange multiplier set Λ are precisely the values of $\mu(\cdot)$ on the corresponding set of stationary points, as given by (2.8). A variant of this result may be found in §2 of Forsythe and Golub [5], where it is used in proving Part 2 of Theorem 2.1 above.

LEMMA 2.1. *Assume that condition* (2.7) *holds* (*i.e., Case* a), *and let* $\lambda \in \Lambda$. *Then*

$$g_\mu(\lambda) = \mu(x^\lambda), \tag{2.16}$$

where $x^\lambda = (B - \lambda I)^{-1}\eta$.

Proof. Using the fact that $(\hat{x}^\lambda)^t \hat{x}^\lambda = 1$, we obtain

$$g_\mu(\lambda) = \sum_{i=1}^{n-1} \left[\lambda \left(\frac{\hat{\eta}_i}{\gamma_i - \lambda} \right)^2 - \frac{\hat{\eta}_i^2}{(\gamma_i - \lambda)} \right]$$

$$= \sum_{i=1}^{n-1} (2\lambda - \gamma_i) \left(\frac{\hat{\eta}_i}{\gamma_i - \lambda} \right)^2$$
$$= \sum_{i=1}^{n-1} \gamma_i \left(\frac{\hat{\eta}_i}{\gamma_i - \lambda} \right)^2 - 2 \sum_{i=1}^{n-1} \left(\frac{\hat{\eta}_i^2}{\gamma_i - \lambda} \right)$$
$$= \hat{\mu}(\hat{x}^\lambda) = \mu(x^\lambda). \qquad \square$$

At this point it will be useful to discuss the graph of the function $g_\mu(\cdot)$ in Case a. Clearly $g_\mu(\cdot)$ possesses a singularity at each eigenvalue γ_i, $i = 1, 2, \ldots, n-1$. Also, $g_\mu(\lambda) \to \infty$ as $\lambda \downarrow \gamma_i$, while $g_\mu(\lambda) \to -\infty$ as $\lambda \uparrow \gamma_i$, for each $i = 1, 2, \ldots, n-1$. Let i be such that $\gamma_{i+1} < \gamma_i$. Then there is at least one root of $g_\mu(\cdot)$ in (γ_{i+1}, γ_i). It is readily checked that $g_\mu'''(\lambda) < 0$, and consequently $g_\mu''(\lambda)$ is monotone decreasing in this interval. It follows that $g_\mu(\cdot)$ has at most one point of inflection on (γ_{i+1}, γ_i), which is possibly also a critical point. Should there be a point of inflection in (γ_{i+1}, γ_i), then on that interval $g_\mu(\cdot)$ is strictly convex to the left of this point, and strictly concave to the right of it. Hence $g_\mu(\cdot)$ has either zero, one, or two critical points on (γ_{i+1}, γ_i), with the possibility of only one critical point being accounted for by the existence of an inflection which is also critical. By again considering $g_\mu''(\cdot)$, we find that $g_\mu(\cdot)$ is strictly convex on the semi-infinite interval (γ_1, ∞), while we have strict concavity on the other semi-infinite interval, namely $(-\infty, \gamma_{n-1})$. Now, since $g_\mu(\lambda) \to \infty$ as $\lambda \downarrow \gamma_1$ and as $\lambda \to \infty$, we conclude that $g_\mu(\cdot)$ has a unique critical point, namely, λ_1 on (γ_1, ∞). Similarly, since $g_\mu(\lambda) \to -\infty$ as $\lambda \uparrow \gamma_{n-1}$ and as $\lambda \to -\infty$, we see that $g_\mu(\cdot)$ has a unique critical point, namely, λ_k on $(-\infty, \gamma_{n-1})$. (Note that this agrees with (2.12) and (2.13).)

In Case a, it is clear that the set of critical points of the secular antiderivative function $g_\mu(\cdot)$ is Λ. Furthermore, in view of our previous discussion, we then have

$$\mu_1 = g_\mu(\lambda_1) > \mu_2 = g_\mu(\lambda_2) > \cdots > \mu_k = g_\mu(\lambda_k), \tag{2.17}$$

where we have adopted the notation

$$\mu_i = \mu(x^{\lambda_i}), \qquad i = 1, 2, \ldots, k$$

for the *stationary values* of $\mu(\cdot)$ on S_{n-1}.

The preceeding discussion is summarized in Fig. 1, which illustrates the graph of a typical secular antiderivative function when (2.7) holds and the γ_i are distinct.

3. Main results. In what follows, we will be considering the border perturbation of the real symmetric $(n-1) \times (n-1)$ matrix B given by

$$A = \begin{pmatrix} B & -\alpha \\ \beta^t & t \end{pmatrix}, \tag{3.1}$$

where α and β are real $(n-1)$-vectors and $t \in R$. Letting P be an orthogonal matrix which diagonalizes B as in (2.3), we define

$$\hat{P} = \begin{pmatrix} P & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.2}$$

Then

$$\hat{A} := \hat{P}^t A \hat{P} = \begin{pmatrix} D & -\hat{\alpha} \\ \hat{\beta}^t & t \end{pmatrix}, \tag{3.3}$$

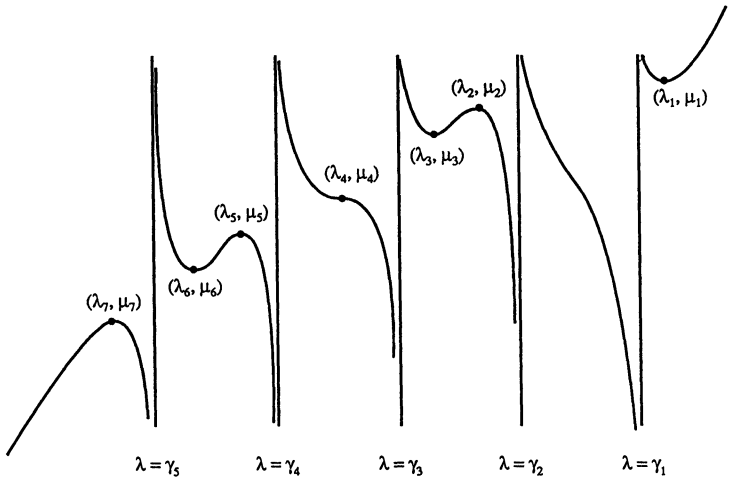


FIG. 1.

where

$$\hat{\alpha} = P^t \alpha, \qquad \hat{\beta} = P^t \beta.$$

Let us assume that

$$\hat{\alpha}_i \hat{\beta}_i \geq 0 \quad \forall i = 1, 2, \ldots, n-1. \tag{3.4}$$

Since a permutation can be built into P, we can without loss of generality assume that

$$\hat{A} = \begin{pmatrix} \tilde{D} & 0 & -\tilde{\alpha} \\ 0 & \bar{D} & -\bar{\alpha} \\ \tilde{\beta}^t & \bar{\beta}^t & t \end{pmatrix}, \tag{3.5}$$

where

$$\begin{aligned} \tilde{D} &= \operatorname{diag}(\tilde{\gamma}_1, \tilde{\gamma}_2, \ldots, \tilde{\gamma}_{\tilde{n}}), \\ \bar{D} &= \operatorname{diag}(\bar{\gamma}_1, \bar{\gamma}_2, \ldots, \bar{\gamma}_{\bar{n}}), \\ \tilde{\alpha}_i \tilde{\beta}_i &= 0 \quad \forall i = 1, 2, \ldots, \tilde{n}, \\ \bar{\alpha}_i \bar{\beta}_i &> 0 \quad \forall i = 1, 2, \ldots, \bar{n}, \end{aligned}$$

and

$$\tilde{n} + \bar{n} = n - 1.$$

Furthermore, we can assume the ordering

$$\bar{\gamma}_1 \geq \bar{\gamma}_2 \geq \cdots \geq \bar{\gamma}_{\bar{n}}.$$

Remark 3.1. (i) Note that condition (3.4) holds if $\alpha = \beta$; that is, when A is given by (1.6).

(ii) It is possible, of course, that either $\tilde{n}$ or $\bar{n}$ may be zero. It is readily shown that

$$\tilde{n} = 0 \iff \textit{no eigenvector of } B \textit{ is orthogonal to } \alpha \textit{ or } \beta. \tag{3.6}$$

Consider the submatrix of $\hat{A}$ given by

$$\bar{A} = \begin{pmatrix} \bar{D} & -\bar{\alpha} \\ \bar{\beta}^t & t \end{pmatrix}. \tag{3.7}$$

We associate with $\bar{A}$ the quadratic function

$$\bar{\mu}(\bar{x}) = \bar{x}^t \bar{D} \bar{x} - 2 \sum_{i=1}^{\bar{n}} (\bar{\alpha}_i \bar{\beta}_i)^{1/2} \bar{x}_i. \tag{3.8}$$

The secular antiderivative function associated with $\bar{\mu}(\cdot)$ is then

$$g_{\bar{\mu}}(\lambda) = \lambda - \sum_{i=1}^{\bar{n}} \left(\frac{\bar{\alpha}_i \bar{\beta}_i}{\bar{\gamma}_i - \lambda} \right). \tag{3.9}$$

From the structure of $\hat{A}$, we see that the characteristic polynomial of A is

$$p(\lambda) = \bar{p}(\lambda) \prod_{i=1}^{\tilde{n}} (\tilde{\gamma}_i - \lambda), \tag{3.10}$$

where

$$\bar{p}(\lambda) = \det(\bar{A} - \lambda \bar{I}). \tag{3.11}$$

In view of (3.10), it is clear that each of the $\tilde{n}$ diagonal entries of $\tilde{D}$ is an eigenvalue of A. Therefore to completely determine the spectrum of A, it is necessary only to determine the spectrum of $\bar{A}$. The following key lemma describes this spectrum in terms of the secular antiderivative function associated with $\bar{\mu}(\cdot)$, and is the basis of our linkage between trust region problems and perturbation theory.

LEMMA 3.1. *The real eigenvalues of $\bar{A}$ that differ from the $\bar{n}$ values $\bar{\gamma}_i$ are the solutions of*

$$g_{\bar{\mu}}(\lambda) = t. \tag{3.12}$$

Proof. Let $\lambda \in R$ where $\lambda \neq \bar{\gamma}_i$ for all $i = 1, 2, \ldots, \bar{n}$. From the Schur complement formula (see [12], p. 22), we then obtain

$$\det(\bar{A} - \lambda I) = \det(\bar{D} - \lambda I)[t - \lambda + \bar{\beta}^t (\bar{D} - \lambda I)^{-1} \bar{\alpha}], \tag{3.13}$$

from which it follows that the real eigenvalues of $\bar{A}$ differing from the $\bar{n}$ numbers $\bar{\gamma}_i$ are the solutions of

$$t - \lambda + \sum_{i=1}^{\bar{n}} \left(\frac{\bar{\alpha}_i \bar{\beta}_i}{\bar{\gamma}_i - \lambda} \right) = 0. \tag{3.14}$$

In view of (3.9), this is equivalent to (3.12). □

Let us denote the set of Lagrange multipliers of $\bar{\mu}(\cdot)$ with respect to $S_{\bar{n}}$ by $\bar{\Lambda}$, and let this set be given by

$$\bar{\lambda}_1 > \bar{\lambda}_2 > \cdots > \bar{\lambda}_m.$$

Since condition (2.6) holds for $\bar{\mu}(\cdot)$, we are presently in Case a. Therefore the set of critical points of $g_{\bar{\mu}}(\cdot)$ is $\bar{\Lambda}$, and moreover, in view of (2.17), we have

$$\bar{\mu}_1 = g_{\bar{\mu}}(\bar{\lambda}_1) > \bar{\mu}_2 = g_{\bar{\mu}}(\bar{\lambda}_2) > \cdots > \bar{\mu}_m = g_{\bar{\mu}}(\bar{\lambda}_m), \tag{3.15}$$

where the stationary values of $\bar{\mu}(\cdot)$ on $S_{\bar{n}}$ are denoted

$$\bar{\mu}_i = \bar{\mu}(\bar{x}^{\bar{\lambda}_i}), \qquad i = 1, 2, \ldots, m.$$

Here

$$\bar{x}^{\bar{\lambda}_i} = (\bar{D} - \bar{\lambda}_i \bar{I})^{-1} \bar{\eta}, \qquad i = 1, 2, \ldots, m,$$

with $\bar{\eta}$ being the $\bar{n}$-vector whose ith component is $(\bar{\alpha}_i \bar{\beta}_i)^{1/2}$.

The next theorem provides a qualitative description of the eigenstructure of the matrix A given by (3.1), when condition (3.4) holds. Realness of the spectrum of A is characterized in terms of the graph of $g_{\bar{\mu}}(\cdot)$, and in particular, in terms of the stationary values of the quadratic function $\bar{\mu}(\cdot)$. Should the spectrum be real, the interlacing relationships between the eigenvalues of A and B are described. Prior to stating the result, we require some further terminology and notation.

Let $\bar{\lambda} \in \bar{\Lambda}$. We shall say that $\bar{\lambda}$ is a *type*-1 *critical point* of $g_{\bar{\mu}}(\cdot)$ if it is a critical point that is also an inflection. Otherwise, we call $\bar{\lambda}$ a *type*-2 *critical point* of $g_{\bar{\mu}}(\cdot)$. From the discussion of the secular antiderivative function given in §2.2, it is clear that the number of type-2 critical points is even, since these points occur pairwise upon the particular bounded intervals $(\bar{\gamma}_{i+1}, \bar{\gamma}_i)$ where they exist, and in addition, there is a single type-2 critical point in each of the semi-infinite intervals $(-\infty, \bar{\gamma}_{\bar{n}})$ and $(\bar{\gamma}_1, \infty)$; these are $\bar{\lambda}_{\bar{n}}$ and $\bar{\lambda}_1$, respectively. Let us denote the sets of type-1 and type-2 critical points of $g_{\bar{\mu}}(\cdot)$ by $\bar{\Lambda}'$ and $\bar{\Lambda}''$, respectively. Then

$$\bar{\Lambda} = \bar{\Lambda}' \cup \bar{\Lambda}''.$$

We shall write the set $\bar{\Lambda}'$ as

$$\bar{\lambda}'_1 > \bar{\lambda}'_2 > \cdots > \bar{\lambda}'_w,$$

while the set $\bar{\Lambda}''$ will be written as

$$\bar{\lambda}_1 > \bar{\lambda}''_1 > \bar{\lambda}''_2 > \cdots > \bar{\lambda}''_{2v} > \bar{\lambda}_{\bar{n}}.$$

Here

$$w + 2v + 2 = m,$$

with w or v possibly being zero. We denote the set of stationary values corresponding to $\bar{\Lambda}'$ as $\{\bar{\mu}'_i\}_{i=1}^{w}$, while the set of stationary values corresponding to $\bar{\Lambda}''$ is written as

$$\{\bar{\mu}_1\} \cup \{\bar{\mu}_m\} \cup \left\{ \bigcup_{i=1}^{2v} \bar{\mu}''_i \right\}.$$

It will be convenient to define the following closed intervals:

$$I_1 = [\bar{\mu}_1, \infty).$$
$$I_m = (-\infty, \bar{\mu}_m].$$
$$I_i'' = [\bar{\mu}_{2i}'', \bar{\mu}_{2i-1}''],\ i = 1, 2, \ldots, v.$$

It is important to note that, in view of (3.15), the intervals defined above are mutually disjoint.

THEOREM 3.1. *Let B be an $(n-1) \times (n-1)$ real symmetric matrix, and let A be the perturbation of B given by* (3.1). *Assume that condition* (3.4) *holds, and that $\hat{A}$ is of the form* (3.5). *Let $\bar{\mu}(\cdot)$ be given by* (3.8). *Then the following hold:*

1. *There exist $n-2$ real eigenvalues $\{\delta_i\}_{i=1}^{n-2}$ of A, including all the eigenvalues of $\tilde{D}$ and $\bar{n}-1$ eigenvalues of $\bar{A}$, which interlace the $n-1$ ordered eigenvalues $\{\gamma_i\}_{i=1}^{n-1}$ of B; that is,*

$$\gamma_1 \geq \delta_1 \geq \gamma_2 \geq \cdots \geq \gamma_{n-2} \geq \delta_{n-2} \geq \gamma_{n-1}. \tag{3.16}$$

2. *The remaining two eigenvalues of A (which are eigenvalues of $\bar{A}$), say $\bar{\delta}_a$ and $\bar{\delta}_b$, are real if and only if*

$$t \in \{\bar{\Lambda}'\} \cup \{I_1\} \cup \{I_m\} \cup \left\{ \bigcup_{i=1}^{v} I_i'' \right\}. \tag{3.17}$$

3. *Furthermore, $\bar{\delta}_a$ and $\bar{\delta}_b$ are real and distinct if and only if t is in the interior of one of the $v+2$ intervals in* (3.17). *In this case, the $\bar{n}+1$ eigenvalues of $\bar{A}$ are real and distinct.*

4. *If* (3.17) *holds, we have the following relations involving $\bar{\delta}_a$ and $\bar{\delta}_b$, where we assume $\bar{\delta}_a \leq \bar{\delta}_b$:*

(a) $t > \bar{\mu}_1 \Longrightarrow \bar{\gamma}_1 < \bar{\delta}_a < \bar{\lambda}_1 < \bar{\delta}_b \leq t.$
(b) $t = \bar{\mu}_1 \Longrightarrow \bar{\gamma}_1 < \bar{\delta}_a = \bar{\lambda}_1 = \bar{\delta}_b \leq t.$
(c) $t = \bar{\mu}_{2i-1}'' \Longrightarrow \bar{\lambda}_{2i}'' < \bar{\delta}_a = \bar{\lambda}_{2i-1}'' = \bar{\delta}_b.$
(d) $t \in (\bar{\mu}_{2i}'', \bar{\mu}_{2i-1}'') \Longrightarrow \bar{\lambda}_{2i}'' < \bar{\delta}_a < \bar{\lambda}_{2i-1}'' < \bar{\delta}_b$ *or* $\bar{\delta}_a < \bar{\lambda}_{2i}'' < \bar{\delta}_b < \bar{\lambda}_{2i-1}''.$
(e) $t = \bar{\mu}_{2i}'' \Longrightarrow \bar{\lambda}_{2i}'' = \bar{\delta}_a = \bar{\lambda}_{2i}'' = \bar{\delta}_b < \bar{\lambda}_{2i-1}''.$
(f) $t = \bar{\mu}_i' \Longrightarrow \bar{\delta}_a = \bar{\mu}_i' = \bar{\delta}_b.$
(g) $t = \bar{\mu}_m \Longrightarrow t \leq \bar{\delta}_a = \bar{\lambda}_m = \bar{\delta}_b < \bar{\gamma}_{\bar{n}}.$
(h) $t < \bar{\mu}_m \Longrightarrow t \leq \bar{\delta}_a < \bar{\lambda}_m < \bar{\delta}_b < \bar{\gamma}_{\bar{n}}.$

Furthermore, in each of the statements (c)–(f), *all values on the right-hand side of $\Longrightarrow$ are contained in a single interval of the form $(\bar{\gamma}_{j+1}, \bar{\gamma}_j)$.*

Proof. Consider the graph of $g_{\bar{\mu}}(\cdot)$, a typical example of which is given in Fig. 2. We see that if $\bar{\gamma}_{i+1} < \bar{\gamma}_i$, then (3.12) has at least one solution $\bar{\delta}_i$ in $(\bar{\gamma}_{i+1}, \bar{\gamma}_i)$, which, in view of Lemma 3.1, is an eigenvalue of $\bar{A}$. Furthermore, since the characteristic equation of $\bar{A}$ is given by

$$\det(\bar{A} - \lambda I) = \prod_{i=1}^{\bar{n}} (\bar{\gamma}_i - \lambda) \left[t - \lambda + \sum_{i=1}^{\bar{n}} \left(\frac{\bar{\alpha}_i \bar{\beta}_i}{\bar{\gamma}_i - \lambda} \right) \right] = 0,$$

it follows that if $\bar{\gamma}_i$ has multiplicity k_i as an eigenvalue of $\bar{D}$, then $\bar{\gamma}_i$ is an eigenvalue of $\bar{A}$ with multiplicity $k_i - 1$. Hence

$$\bar{\gamma}_1 \geq \bar{\delta}_1 \geq \bar{\gamma}_2 \geq \cdots \geq \bar{\gamma}_{\bar{n}-1} \geq \bar{\delta}_{\bar{n}-1} \geq \bar{\gamma}_{\bar{n}}, \tag{3.18}$$

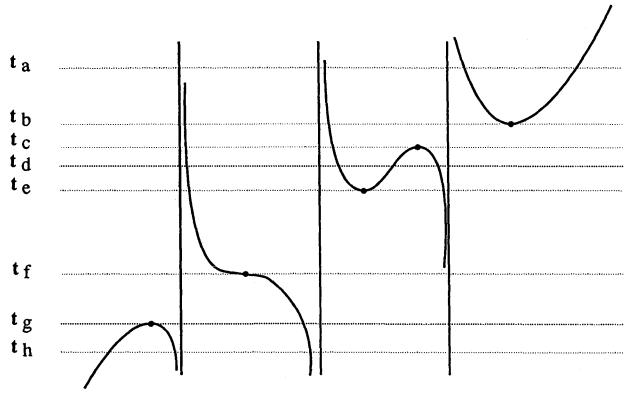


FIG. 2.

where the $\bar{n}-1$ numbers $\bar{\delta}_i$, are eigenvalues of $\bar{A}$. Part 1 of the theorem now follows readily. Parts 2 and 3 are consequences of part 4, which follows directly from consideration of the graph of $g_{\bar{\mu}}(\cdot)$, as in Fig. 2. There the relevant values of t are indicated, with the subscripts on t corresponding to (a)–(h) above. (Note that there are two possibilities for (d).) That $\bar{\delta}_b \leq t$ in (a) and (b) follows from the graph and the fact that the trace of $\bar{A}$ is the sum of the eigenvalues of $\bar{A}$, as does the inequality $t \leq \bar{\delta}_a$ occurring in (g) and (h). □

Remark 3.2. In Theorem 3.1, we can replace $\bar{\mu}(\cdot)$ with any quadratic function of the form

$$\bar{x}^t \bar{D} \bar{x} - 2 \sum_{i=1}^{\bar{n}} \psi_i (\bar{\alpha}_i \bar{\beta}_i)^{1/2} \bar{x}_i,$$

where $\psi_i = \pm 1$, since this change does not alter the Lagrange multipliers or critical values of $\bar{\mu}(\cdot)$ with respect to $S_{\bar{n}}$.

Theorem 3.1 gives a detailed description of the eigenstructure of the perturbation A under assumption (3.4), and in particular, a complete characterization of when the spectrum of A is real. However, to apply the result, one requires an orthogonal diagonalization of B, and this may not be readily available. In the following corollary, sufficient conditions for realness of the spectrum of A are given, without reliance on an orthogonal diagonalization, in case A is given by (1.6); that is, when $\alpha = \beta$.

COROLLARY 3.1. *Let B be an $(n-1) \times (n-1)$ real symmetric matrix, and consider the perturbation of B given by*

$$A = \begin{pmatrix} B & -\alpha \\ \alpha^t & t \end{pmatrix}, \tag{3.19}$$

where α is a real $(n-1)$-vector. Define

$$\mu(x) = x^t B x - 2\alpha^t x. \tag{3.20}$$

Let

$$\mu_1 = \max\{\mu(x) \ : \ x^t x = 1\} \tag{3.21}$$

and

$$\mu_k = \min\{\mu(x) \ : \ x^t x = 1\}. \tag{3.22}$$

Then either of the conditions

$$t \geq \mu_1 \tag{3.23}$$

or

$$t \leq \mu_k \tag{3.24}$$

are sufficient for the spectrum of A to be real.

Proof. As was noted in Remark 3.1, condition (3.4) holds for the present perturbation. Now observe that $\mu(\cdot)$ and $\bar{\mu}(\cdot)$ have the same secular function and secular antiderivative, where $\bar{\mu}(\cdot)$ is given by (3.8) with $\bar{\alpha}_i = \bar{\beta}_i$ for $i = 1, 2, \ldots, \bar{n}$. Since the roots of the secular function $f_\mu(\cdot)$ are the critical points of the secular antiderivative $g_\mu(\cdot)$, the discussion of Cases a and b in §2.1 tells us that

$$\lambda_1 \geq \bar{\lambda}_1$$

and

$$\lambda_k \leq \bar{\lambda}_m.$$

From Part 2 of Theorem 2.1, we then have

$$\mu_1 \geq \bar{\mu}_1$$

and

$$\mu_k \leq \bar{\mu}_m.$$

(The last two inequalities can also be deduced directly from the definitions of the functions $\mu(\cdot)$ and $\bar{\mu}(\cdot)$.) The result now follows from Theorem 3.1, and in particular, from part 4, (a), (b), (g), and (h). □

The values μ_1 and μ_k in Corollary 3.1, which are the optimal objective function values of the trust region problems $P^{\mu}_{\max}$ and $P^{\mu}_{\min}$, respectively, may be efficiently determined numerically by the method of Moré and Sorensen [18].

In Corollary 3.2, we give a Gersgorin-like sufficient condition for realness of the spectrum of A given by (3.19). We use the notation $\| \cdot \|$ for both the euclidean norm of an $(n-1)$-vector and the spectral norm of an $(n-1) \times (n-1)$ matrix.

COROLLARY 3.2. *Let B and A be as in Corollary* 3.1. *Then a sufficient condition for the spectrum of A to be real is*

$$\| B \| + 2 \| \alpha \| \leq |t|. \tag{3.25}$$

Proof. This follows from the fact that (3.25) implies that either (3.23) or (3.24) hold, and Corollary 3.1. □

We conclude this section with another result regarding the perturbation (3.19). This elementary result is independent of Theorem 3.1, and yields further connections between trust region problems and nonsymmetric perturbations.

THEOREM 3.2. *Let B, A, and $\mu(\cdot)$ be as in Corollary* 3.1. *Then $\lambda \in \Lambda$ (that is, λ is a Lagrange multiplier of $\mu(\cdot)$ with respect to S_{n-1}) with $x \in S_\mu(\lambda)$, if and only if λ is an eigenvalue of A with associated eigenvector*

$$\begin{pmatrix} x \\ 1 \end{pmatrix},$$

in which case $||x|| = 1$ and $t = \mu(x)$.

Proof. Upon premultiplying the Lagrange equation

$$Bx - \alpha - \lambda x = 0 \tag{3.26}$$

by x^t and using the fact that $x^t x = 1$, we obtain

$$\mu(x) + \alpha^t x - \lambda = 0, \tag{3.27}$$

i.e., the following eigenvalue-eigenvector equation holds

$$\begin{pmatrix} B - \lambda I & -\alpha \\ \alpha^t & t - \lambda \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = 0. \tag{3.28}$$

Conversely, suppose that the above eigenvalue-eigenvector equation holds, with $t = \mu(x)$. Then the Lagrange equation (3.26) and (3.27) clearly hold. Premultiplying by x^t again and substituting for λ yields $x^t Bx - \alpha^t x - (\mu(x) + \alpha^t x)x^t x = 0$, which implies that $x^t x = 1$. □

4. Applications.

4.1. Exponential nonnegativity. In this subsection it will be seen that the main results of §3 can be applied to characterize those matrices that are exponentially nonnegative with respect to the n-dimensional ice-cream cone and to provide a decomposition theorem for such matrices.

Let us denote the *n-dimensional ice-cream cone* by

$$K_n = \left\{ y \in R^n \ : \ \sum_{i=1}^{n-1} y_i^2 \leq y_n^2, \ y_n \geq 0 \right\}.$$

Equivalently,

$$K_n = \{ y \in R^n \ : \ y^t Q_n y \leq 0, \ y_n \geq 0 \},$$

where $Q_n = \operatorname{diag}(1, 1, \ldots, 1, -1)$. We shall denote the matrix exponential by

$$e^{tA} = \sum_{j=0}^{\infty} (tA)^j / j!,$$

and the boundary of K_n by ∂K_n. The following further notation and terminology will be utilized:

$$\begin{aligned} \Pi(K_n) &= \{ A : AK_n \subset K_n \}. \\ e(K_n) &= \{ A : e^{tA} \subset \Pi(K_n) \ \ \forall t \geq 0 \}. \\ e(\partial K_n) &= \{ A : e^{tA}(\partial K_n) \subset \partial K_n \forall t \geq 0 \}. \end{aligned}$$

These sets will be referred to as the *K_n-nonnegative matrices*, *exponentially K_n-nonnegative matrices*, and *exponentially ∂K_n-invariant matrices*, respectively. It is readily verified that both $\Pi(K_n)$ and $e(\partial K_n)$ are subsets of $e(K_n)$. Although our discussion will be essentially self-contained, the reader is referred to Berman and Plemmons [2] and Berman, Neumann, and Stern [1] for general facts concerning these sets of matrices.

Notice that $A \in e(K_n)$ if and only if for any initial point $y_o \in K_n$, the solution $y(t) = e^{tA}y_o$ of the initial value problem

$$\frac{d}{dt}y(t) = Ay(t); \qquad t \geq 0,$$
$$y(0) = y_o$$

satisfies $y(t) \in K_n$ for all $at \geq 0$. Similarly, $A \in e(\partial K_n)$ means that $y(t) \in \partial K_n$ for all $y_o = y(0) \in \partial K_n$.

We require the following lemma of Stern and Wolkowicz [22], in which $e(K_n)$ and $e(\partial K_n)$ are characterized in terms of tangency-like properties of the vector field $\{Ay\}$ relative to the surface

$$\partial K_n = \{y \in R^n \ : \ y^tQ_ny = 0, \ y_n \geq 0\}.$$

LEMMA 4.1. *Let A be a real $n \times n$ matrix. Then the following hold:* 1. *A necessary and sufficient condition for $A \in e(K_n)$ is*

$$y^tQ_nAy \leq 0 \quad \forall y \in \partial K_n. \tag{4.1}$$

2. *A necessary and sufficient condition for $A \in e(\partial K_n)$ is*

$$y^tQ_nAy = 0 \quad \forall y \in \partial K_n, \tag{4.2}$$

which is in turn equivalent to A being of the form

$$A = \begin{pmatrix} G + aI & g \\ g^t & a \end{pmatrix}, \tag{4.3}$$

for some real $(n-1)$-vector g and real number a, where the $(n-1) \times (n-1)$ matrix G is skew-symmetric.

We next use Corollary 3.1 to characterize $e(K_n)$ in terms of the maximal critical value of a specific trust region problem, as well as in terms of the realness of the spectrum of a certain matrix.

Suppose that $A \in e(K_n)$, or equivalently, that (4.1) holds. Let us partition A as

$$\begin{pmatrix} A_1 & c \\ d^t & a_{nn} \end{pmatrix}. \tag{4.4}$$

Then condition (4.1) becomes

$$x^tA_1x + (c^t - d^t)x - a_{nn} \leq 0 \quad \forall x \in S_{n-1}. \tag{4.5}$$

Let us define

$$B = \frac{A_1 + A_1^t}{2}. \tag{4.6}$$

From (4.5) it follows that (4.1) is equivalent to

$$\mu(x) := x^t Bx - 2\alpha^t x \leq a_{nn} \quad \forall x \in S_{n-1}, \tag{4.7}$$

where

$$\alpha = \frac{d-c}{2}. \tag{4.8}$$

Defining

$$\mu_1 = \max\{\mu(x) \ : \ x^t x = 1\}$$

as in (3.21), we see that (4.7) becomes

$$\mu_1 \leq a_{nn}. \tag{4.9}$$

Now, in view of Corollary 3.1, (4.9) implies that the spectrum of the matrix

$$A_r = \begin{pmatrix} B & (c-d)/2 \\ (d^t - c^t)/2 & a_{nn} \end{pmatrix} \tag{4.10}$$

is real. We shall call A_r the *regularization* of A.

The preceding discussion is summarized in the following result.

THEOREM 4.1. *Let A be a real $n \times n$ matrix. Then the following hold:* 1. *A is exponentially K_n-nonnegative if and only if* (4.9) *holds.*

2. *A necessary condition for A to be exponentially K_n-nonnegative is that the spectrum of A_r be real.*

Example 4.1. In this example,

$$A = \begin{pmatrix} -1 & 1 & 1 \\ 4 & 2 & 3 \\ 0 & 1 & a_{33} \end{pmatrix}.$$

We wish to determine those values of a_{33} for which $A \in e(K_n)$. The regularization of A becomes

$$A_r = \begin{pmatrix} -1 & \frac{5}{2} & -\frac{1}{2} \\ \frac{5}{2} & 2 & -1 \\ \frac{1}{2} & 1 & a_{33} \end{pmatrix}.$$

Therefore

$$\mu(x) = -x_1^2 + 2x_2^2 + 5x_1x_2 - x_1 - 2x_2. \tag{4.11}$$

At this point, one could employ the algorithm of Moré and Sorensen [18] to compute μ_1. Alternatively, one can find an orthogonal diagonalization of B with MATLAB, and then generate the graph of $g_\mu(\cdot)$. The eigenvalues of B are thusly found to be $\lambda_1 = 3.4155$ and $\lambda_2 = -2.4155$, while an orthogonal matrix that diagonalizes B is

$$P = \begin{pmatrix} .8702 & .4927 \\ -.4927 & .8702 \end{pmatrix}.$$

Then

$$P\alpha = P \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix} = \begin{pmatrix} -.0576 \\ 1.1166 \end{pmatrix}.$$

It follows that

$$g_\mu(\lambda) = \lambda - \frac{.0576^2}{(-2.4155 - \lambda)} - \frac{1.1166^2}{(3.4155 - \lambda)}. \tag{4.12}$$

Note that the present example is Case a (since no component of $P^t\alpha$ is zero). We used MATLAB to graphically determine $\mu_1 = 5.67$.

Hence

$$A \in e(K_n) \Longleftrightarrow a_{33} \geq 5.67.$$

Furthermore, if a_{33} satisfies this inequality, then the spectrum of A_r is guaranteed to be real.

Our main objective in the remainder of this subsection is to prove that every exponentially K_n-nonnegative matrix may be (nonuniquely) represented as the sum of a K_n-nonnegative matrix and an exponentially ∂K_n-invariant matrix. Formally, this decomposition result is stated as follows.

THEOREM 4.2. *One has*

$$e(K_n) = \Pi(K_n) + e(\partial K_n). \tag{4.13}$$

In proving this theorem, we make use of Theorem 4.1. We also require the following result, which provides characterizations of $\Pi(K_n)$ and $e(K_n)$ in terms of definiteness conditions. (For a real symmetric matrix C, the notation $C \leq 0$ indicates that C is negative semidefinite.)

THEOREM 4.3. *Let A be a real $n \times n$ matrix.* 1. *Assume that* $\operatorname{rank}(A) > 1$. *Then a necessary and sufficient condition for*

$$A \in \Pi(K_n) \cup \{-\Pi(K_n)\}$$

is the existence of $\mu \geq 0$ such that

$$A^t Q_n A - \mu Q_n \leq 0. \tag{4.14}$$

2. *A necessary and sufficient condition for $A \in e(K_n)$ is the existence of $\gamma \in R$ such that*

$$Q_n A + A^t Q_n - \gamma Q_n \leq 0. \tag{4.15}$$

Part 1 of Theorem 4.3 is due to Loewy and Schneider [14], while part 2 is due to Stern and Wolkowicz [22].

We now shall prove the decomposition theorem.

Proof of Theorem 4.2. Let $A \in e(K_n)$. By part 2 of Theorem 4.1, we know that the spectrum of A_r is real, and we can choose $\delta > 0$ such that all eigenvalues of

$$\tilde{A} = A_r + \delta I$$

are positive. Let Γ be an open disk in the open right-half complex plane, centered at $\psi > 0$, such that the entire spectrum of $\tilde{A}$ lies within Γ. Inside Γ, one can express a branch of the function $f(\lambda) = \lambda^{1/2}$ as

$$f(\lambda) = \sum_{i=0}^{\infty} c_i(\lambda - \psi)^i, \tag{4.16}$$

where the coefficients c_i are all real. According to the theory of matrix functions (see, e.g., [8]), $\tilde{A}$ has a real square root given by

$$\tilde{A}^{1/2} = \sum_{i=0}^{\infty} c_i (\tilde{A} - \psi I)^i. \tag{4.17}$$

Let us write

$$A = \tilde{A} + C, \tag{4.18}$$

where

$$C = \frac{1}{2} \begin{pmatrix} A_1 - A_1^t & c + d \\ c^t + d^t & 0 \end{pmatrix}. \tag{4.19}$$

Then $C \in e(\partial K_n)$, and, in view of Lemma 4.1, it follows that $\tilde{A}$ and A_r are exponentially K_n-nonnegative.

Since $Q_n A_r$ is symmetric, so is $Q_n \tilde{A}$. Then part 2 of Theorem 4.3 implies the existence of $\tilde{\gamma}$ such that

$$Q_n \tilde{A} - \tilde{\gamma} Q_n \leq 0. \tag{4.20}$$

From the fact that the spectrum of $\tilde{A}^{1/2}$ is real and positive, it follows that

$$(\tilde{A}^{1/2})^t [Q_n \tilde{A} - \tilde{\gamma} Q_n] \tilde{A}^{1/2} \leq 0. \tag{4.21}$$

Now, (4.17) implies that the matrix $Q_n \tilde{A}^{1/2}$ is symmetric, and therefore (4.21) yields

$$Q_n \tilde{A}^2 - \tilde{\gamma} Q_n \tilde{A} \leq 0. \tag{4.22}$$

Again using the symmetry of $Q_n \tilde{A}$, it follows that

$$\tilde{A}^t Q_n \tilde{A} - \tilde{\gamma} Q_n \tilde{A} \leq 0. \tag{4.23}$$

We can assume without loss of generality that $\tilde{\gamma} \geq 0$, since δ can be increased, if necessary. Then upon combining (4.20) and (4.23), we arrive at

$$\tilde{A}^t Q_n \tilde{A} - \tilde{\gamma}^2 Q_n \leq 0. \tag{4.24}$$

Since rank($\tilde{\mathrm{A}}$) = n, part 1 of Theorem 4.3 implies

$$\tilde{A} \in \Pi(K_n) \cup \{-\Pi(K_n)\}. \tag{4.25}$$

Then (4.25) yields

$$\tilde{A} e^n \in K_n \cup \{-K_n\},$$

where $e^n = (0, 0, \ldots, 0, 1)^t$. We can assume that δ has been chosen sufficiently large to ensure that

$$\tilde{a}_{nn} = a_{nn} + \delta \geq 0. \tag{4.26}$$

Hence (4.25) and (4.26) imply that $Ae^n \in K_n$. We conclude that

$$\tilde{A} = A - C \in \Pi(K_n). \tag{4.27}$$

Since $C \in e(\partial K_n)$, it follows that

$$A \in \Pi(K_n) + e(\partial K_n). \tag{4.28}$$

This completes the proof. □

Example 4.2. Let

$$A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -2 & 2 \\ 0 & 0 & 0 \end{pmatrix}.$$

Here

$$\langle Ay, Q_3 y \rangle = -(y_2 - 1)^2 \leq 0 \quad \forall y \in K_3,$$

and therefore A is exponentially K_3-nonnegative. Then

$$A_r = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & -1 & 0 \end{pmatrix},$$

and following the proof of Theorem 4.2, $\tilde{A} = A_r + \delta I \in \Pi(K_3)$ provided that δ is chosen sufficiently large. Indeed, if we take $\delta = 2$, then

$$\tilde{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 2 \end{pmatrix}.$$

It is readily checked that $\langle \tilde{A}y, Q_3 \tilde{A}y \rangle = -2(y_2 - 1)^2 \leq 0$ and $(\tilde{A}y)_3 \geq 0$ for all $y \in K_3$. Hence $\tilde{A} \in \Pi(K_3)$, and therefore $A \in \Pi(K_3) + e(\partial K_3)$.

Remark 4.1. Given an $n \times n$ exponentially K_n-nonnegative matrix in the regularized form A_r, define

$$\delta^* = \min\{\delta \in R : A_r + \delta I \in \Pi(K_n)\}.$$

Let us denote the eigenvalues of A_r by $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. It is conjectured that

$$\delta^* = -\frac{(\lambda_1 + \lambda_n)}{2}.$$

This is precisely the minimal value of δ that will ensure that the spectral radius of the matrix $A_r + \delta I$ is in its spectrum. Therefore this conjecture relates to the result of Vandergraft in [23], which asserts that a matrix leaves a proper cone invariant only if its spectral radius is an eigenvalue. (The cone K is said to be *proper* provided that it is closed, convex, possesses nonempty interior, and $K \cap \{-K\} = \{0\}$.) Note that δ^* is generally less than the "sufficiently large" value of δ used in the proof of Theorem 4.2 to ensure various properties, including the existence of a real square root $\tilde{A}^{1/2} = (A_r + \delta I)^{1/2}$. As an illustration, consider the matrix A_r in Example 4.2. The spectrum of A_r is $\{-1, -1, -1\}$, and therefore $\delta^* = 1$. Then

$$A_r + \delta^* I = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 1 \end{pmatrix} \in \Pi(K_3),$$

but this matrix does not possess a real square root.

Remark 4.2. An interesting open problem is to determine whether the decomposition (4.13) holds for any proper cone $K \subset R^n$; that is

$$e(K) = \Pi(K) + e(\partial K). \tag{4.29}$$

It is not difficult to verify (4.29) for the class of ellipsoidal cones; these are the linear homeomorphisms of K_n. Also, results in Schneider and Vidyasagar [19] may be applied to verify (4.29) for the class of polyhedral proper cones, and to show that for a general proper cone K, we have

$$e(K) = \overline{\Pi(K) + e(\partial K)}, \tag{4.30}$$

where the bar denotes closure. Hence (4.30) implies that conjecture (4.29) is equivalent to closedness of $\Pi(K)+e(\partial K)$. The sets $\Pi(K)$ and $e(\partial K)$ can be shown to be a proper cone and a subspace, respectively, in the space of $n \times n$ matrices. Such a sum is not necessarily closed, and the conjecture therefore remains unsettled.

4.2. Interlacing and majorization. Given two real n-vectors x and y with component orderings

$$x_1 \geq x_2 \geq \cdots \geq x_n \tag{4.31}$$

and

$$y_1 \geq y_2 \geq \cdots \geq y_n, \tag{4.32}$$

we say that x is *majorized* by y (notationally, $x \prec y$) provided that

$$\begin{aligned} x_1 &\leq y_1, \\ x_1 + x_2 &\leq y_1 + y_2, \\ &\vdots \\ x_1 + x_2 + \cdots + x_{n-1} &\leq y_1 + y_2 + \cdots + y_{n-1}, \\ x_1 + x_2 + \cdots + x_n &= y_1 + y_2 + \cdots + y_n. \end{aligned}$$

The following is a classical theorem of Schur [20].

THEOREM 4.4. *Let A be a real symmetric $n \times n$ matrix with diagonal elements*

$$a_{11} \geq a_{22} \geq \cdots \geq a_{nn}$$

and eigenvalues

$$\delta_1 \geq \delta_2 \geq \cdots \geq \delta_n.$$

Then

$$(a_{11}, a_{22}, \ldots, a_{nn}) \prec (\delta_1, \delta_2, \ldots, \delta_n). \tag{4.33}$$

One proof of Schur's theorem appearing in Mirsky [16], and attributed there to Schneider, makes use of the interlacing (1.5) obtained upon writing A in the form (1.4). (See also Theorem 9.B.1 in Marshall and Olkin [15] or Theorem 4.3.26 in Horn and Johnson [12].) We next give an analogous "near-majorization" result and proof

for (nonsymmetric) matrices of the form (3.1) satisfying condition (3.4), by applying the "near-interlacing" provided by Theorem 3.1. We first introduce some further terminology and notation.

Given real n-vectors x and y satisfying the orderings (4.31) and (4.32), we say that x is *near-majorized* by y (notationally, $x \tilde{\prec} y$) provided that

$$\begin{aligned} x_2 &\le y_2, \\ x_2 + x_3 &\le y_2 + y_3, \\ &\vdots \\ x_2 + x_3 + \cdots + x_n &\le y_2 + y_3 + \cdots + y_n, \\ x_1 + x_2 + \cdots + x_n &= y_1 + y_2 + \cdots + y_n. \end{aligned}$$

Note that $x \tilde{\prec} y$ implies $x_1 \ge y_1$.

THEOREM 4.5. *Assume that the hypotheses of Theorem* 3.1 *hold, and consider the matrix* $\bar{A}$ *given by* (3.7).

1. *Assume that* $t \ge \bar{\mu}_1$. *Then the spectrum of* $\bar{A}$ *is real,* $t \ge \bar{\gamma}_1$, *and*

$$(t, \bar{\gamma}_1, \bar{\gamma}_2, \ldots, \bar{\gamma}_{\bar{n}}) \tilde{\prec} v,$$

where v *denotes the vector of eigenvalues of* $\bar{A}$ *listed in nonincreasing order.*

2. *Assume that* $t \le \bar{\mu}_m$. *Then the spectrum of* $\bar{A}$ *is real,* $t \le \bar{\gamma}_{\bar{n}}$, *and*

$$-(\bar{\gamma}_1, \bar{\gamma}_2, \ldots, \bar{\gamma}_{\bar{n}}, t) \tilde{\prec} -v.$$

Proof. We only prove part 1 of the theorem. The proof of part 2 is similar and is left to the reader.

In view of Theorem 3.1 (part 4(a) and 4(b)), the eigenvalues of $\bar{A}$ are all real and the ordered spectrum of $\bar{A}$ is given by

$$\bar{\delta}_{\bar{n}-1} \le \bar{\delta}_{\bar{n}-2} \le \cdots \le \bar{\delta}_1 \le \bar{\delta}_a \le \bar{\delta}_b, \tag{4.34}$$

and we have

$$\bar{\gamma}_{\bar{n}} \le \bar{\delta}_{\bar{n}-1} \le \bar{\gamma}_{\bar{n}-1} \le \cdots \le \bar{\delta}_1 \le \bar{\gamma}_1 \le \bar{\delta}_a \le \bar{\delta}_b \le t. \tag{4.35}$$

This yields the system of inequalities

$$\begin{aligned} \bar{\gamma}_1 &\le \bar{\delta}_a, \\ \bar{\gamma}_1 + \bar{\gamma}_2 &\le \bar{\delta}_a + \bar{\delta}_1, \\ &\vdots \\ \bar{\gamma}_1 + \bar{\gamma}_2 + \cdots + \bar{\gamma}_{\bar{n}} &\le \bar{\delta}_a + \bar{\delta}_1 + \cdots + \bar{\delta}_{\bar{n}-1}. \end{aligned}$$

The result now follows from the fact that

$$\begin{aligned} \operatorname{trace}(\bar{A}) &= t + \bar{\gamma}_1 + \bar{\gamma}_2 + \cdots + \bar{\gamma}_{\bar{n}} \\ &= \bar{\delta}_b + \bar{\delta}_a + \bar{\delta}_1 + \cdots + \bar{\delta}_{\bar{n}-1}. \end{aligned}$$

□

In the following corollary to Theorem 4.5, we obtain a near-majorization result for matrices of the form (1.6).

COROLLARY 4.1. *Let B, A, $\mu(\cdot)$, μ_1 and μ_k be as in Corollary* 3.1, *where the diagonal of B is assumed to have the ordering*

$$b_{11} \geq b_{22} \geq \cdots \geq b_{(n-1)(n-1)}. \tag{4.36}$$

In addition, assume that no eigenvector of B is orthogonal to α.

1. *If condition* (3.23) *holds (that is, $t \geq \mu_1$), then the spectrum of A is real, $t \geq b_{11}$, and*

$$(t, b_{11}, b_{22}, \ldots, b_{(n-1)(n-1)}) \tilde{\prec} v,$$

where v is vector of eigenvalues of A listed in nonincreasing order.

2. *If condition* (3.24) *holds (that is, $t \leq \mu_k$), then the spectrum of A is real, $t \leq b_{(n-1)(n-1)}$, and*

$$-(b_{11}, b_{22}, \ldots, b_{(n-1)(n-1)}, t) \tilde{\prec} -v.$$

Proof. We will only prove part 1, with the proof of part 2 being left to the reader.

Since condition (3.6) holds, we have $\hat{A} = \bar{A}$, where $\hat{A}$ and $\bar{A}$ are given by (3.5) and (3.7), respectively. From the proof of Corollary 3.1, we see that the eigenvalues of A are all real, and that the entire ordered spectrum of A is given by (4.34) with $\bar{n} = n - 1$. Upon applying Schur's majorization theorem to B, we obtain

$$\begin{aligned} b_{11} &\leq \bar{\gamma}_1, \\ b_{11} + b_{22} &\leq \bar{\gamma}_1 + \bar{\gamma}_2, \\ &\vdots \\ b_{11} + b_{22} + \cdots + b_{(n-1)(n-1)} &\leq \bar{\gamma}_1 + \bar{\gamma}_2 + \cdots + \bar{\gamma}_{n-1}, \end{aligned}$$

and making use of (4.35) as in the proof of Theorem 4.5, we have that

$$\begin{aligned} \bar{\gamma}_1 &\leq \bar{\delta}_a, \\ \bar{\gamma}_1 + \bar{\gamma}_2 &\leq \bar{\delta}_a + \bar{\delta}_1, \\ &\vdots \\ \bar{\gamma}_1 + \bar{\gamma}_2 + \cdots + \bar{\gamma}_{n-1} &\leq \bar{\delta}_a + \bar{\delta}_1 + \bar{\delta}_2 + \cdots + \bar{\delta}_{n-2}. \end{aligned}$$

The result now follows from the facts that $t \geq \bar{\gamma}_1 \geq b_{11}$ and

$$\begin{aligned} \text{trace(A)} &= t + b_{11} + b_{22} + \cdots + b_{(n-1)(n-1)} \\ &= \bar{\delta}_b + \bar{\delta}_a + \bar{\delta}_1 + \bar{\delta}_2 + \cdots + \bar{\delta}_{n-2}. \end{aligned}$$

□

Notice that the ordering of the diagonal (4.36) can be assumed to hold without loss of generality in Corollary 4.1, since it can always be attained via a permutational similarity.

The following inverse eigenvalue theorem is due to Mirsky [17]; see also Theorem 9.3.B in [15].

THEOREM 4.6. *Suppose we are given real numbers $\gamma_1, \gamma_2, \ldots, \gamma_{n-1}$ and $\delta_1, \delta_2, \ldots, \delta_n$, which satisfy the interlacing property* (1.5); *that is*

$$\delta_1 \geq \gamma_1 \geq \delta_2 \geq \gamma_2 \geq \cdots \geq \gamma_{n-1} \geq \delta_n.$$

Then there exists a real symmetric $n \times n$ matrix of the form

$$A = \begin{pmatrix} D & \eta \\ \eta^t & t \end{pmatrix} \tag{4.37}$$

with $D = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_{n-1})$, *such that the spectrum of* A *is* $\{\delta_1, \delta_2, \ldots, \delta_n\}$.

We will now show that an analogous inverse eigenvalue theorem holds when the interlacing (1.5) is replaced by the types of near-interlacing occurring in Theorem 3.1. The proof closely follows Mirsky, but we include it nevertheless for the sake of completeness. (See also Theorem 7 in [13].)

THEOREM 4.7. *Suppose we are given real numbers* $\gamma_1, \gamma_2, \ldots, \gamma_{n-1}$, $\delta_1, \delta_2, \ldots, \delta_{n-2}$, *and* δ_b, δ_a, *which satisfy*

$$\gamma_1 \geq \delta_1 \geq \gamma_2 \geq \delta_2 \geq \cdots \geq \delta_{n-2} \geq \gamma_{n-1}. \tag{4.38}$$

Assume that one of the following three cases holds:

1. $\delta_b \geq \delta_a \geq \gamma_1$;
2. $\gamma_{n-1} \geq \delta_b \geq \delta_a$;
3. $\gamma_j \geq \delta_j \geq \delta_b \geq \delta_a \geq \gamma_{j+1}$ *for some* j, $1 \geq j \geq n-2$.

Then there exists a real $n \times n$ *matrix of the form*

$$A = \begin{bmatrix} D & \eta \\ -\eta^t & t \end{bmatrix},$$

such that $D = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_{n-1})$ *and such that the spectrum of* A *is* $\{\delta_1, \ldots, \delta_{n-1}\} \cup \{\delta_b, \delta_a\}$.

Proof. The characteristic equation of A is given by

$$\det(\lambda I - A) = \prod_{i=1}^{n-1} (\lambda - \gamma_i) \left[\lambda - t + \sum_{j=1}^{n-1} \left(\frac{\eta_j^2}{\lambda - \gamma_j} \right) \right] = 0. \tag{4.39}$$

We need to choose η and t so that the numbers δ_i, $i = 1, \ldots, n$, are the roots of (4.39), where with some abuse of notation, we refer to δ_b, δ_a as δ_{n-1}, δ_n. First suppose that the δ_i are distinct. Let

$$f(\lambda) = \prod_{i=1}^{n} (\lambda - \delta_i), \qquad g(\lambda) = \prod_{i=1}^{n-1} (\lambda - \gamma_i).$$

By direct verification, or by Lagrange's interpolation formula, we have

$$\frac{f(\lambda)}{g(\lambda)} = \lambda - \left(\sum_{i=1}^{n} \delta_i - \sum_{i=1}^{n-1} \gamma_i \right) + \sum_{k=1}^{n-1} \frac{f(\gamma_k)}{g'(\gamma_k)} \frac{1}{(\lambda - \gamma_k)}. \tag{4.40}$$

Due to the near-interlacing in case 1, that is, when $\delta_b \geq \delta_a \geq \gamma_1$ (or $\delta_{n-1} \geq \delta_n \geq \gamma_1$), we have

$$\begin{aligned} f(\gamma_k) &= \prod_{i=n-1}^{n} (\gamma_k - \delta_i) \prod_{i=1}^{k-1} (\gamma_k - \delta_i) \prod_{i=k}^{n-2} (\gamma_k - \delta_i) \\ &= (-1)^2 \prod_{i=n-1}^{n} |\gamma_k - \delta_i| (-1)^{k-1} \prod_{i=1}^{k-1} |\gamma_k - \delta_i| \prod_{i=k}^{n-2} (\gamma_k - \delta_i) \\ &= (-1)^{k-1} \prod_{i=1}^{n} |\gamma_k - \delta_i| \end{aligned}$$

and

$$g'(\gamma_k) = (-1)^{k-1} \prod_{\substack{i=1 \\ i \neq k}}^{n-1} |\gamma_k - \gamma_i|.$$

It follows that

$$\frac{f(\gamma_k)}{g(\gamma_k)} \geq 0, \quad k = 1, 2, \ldots, n-1.$$

Similarly, this can be shown to hold in the other two cases as well.

Now choose

$$\eta_k^2 = \frac{f(\gamma_k)}{g'(\gamma_k)}, \qquad k = 1, 2, \ldots, n$$

and

$$t = \sum_{i=1}^{n} \delta_i - \sum_{i=1}^{n-1} \gamma_i.$$

Then the eigenvalues of A are the roots of $f(\cdot)$, which are the n values δ_i. (A modification of the proof yields the case of nondistinct δ_i.) □

Remark 4.3. In [11], A. Horn proved the following inverse eigenvalue result, which may be viewed as a converse to Theorem 4.4.

If we are given real numbers

$$a_{11} \geq a_{22} \geq \cdots \geq a_{nn}$$

and

$$\delta_1 \geq \delta_2 \geq \cdots \geq \delta_n,$$

such that the majorization (4.33) holds, then there exists an $n \times n$ real symmetric matrix A with diagonal elements $a_{11}, a_{22}, \ldots, a_{nn}$ and with eigenvalues $\delta_1, \delta_2, \ldots, \delta_n$.

One proof of Horn's theorem is due to Mirsky [17], and relies on Theorem 4.6. (See also Theorem 9.B.2 in [15].) Hence it seems apppropriate to ask whether one can obtain analogous converses to Theorem 4.5 or Corollary 4.1 by utilizing Theorem 4.7. At the present time, this remains an open problem.

Remark 4.4. In this subsection we have seen that known results for symmetric matrices regarding interlacing and majorization can be extended, under certain conditions, to "near-symmetric" matrices, i.e., matrices of the form (3.19). This was possible because these extensions depended more on the realness of the spectrum than on symmetry per se. Other results in the literature regarding symmetric matrices can also be extended to the near-symmetric case by employing the present work; e.g., results on eigenvalue bounds appearing in Wolkowicz and Styan [25].

Acknowledgment. The authors wish to thank Q. Ye for pointing out reference [13].

REFERENCES

[1] A. BERMAN, M. NEUMANN, AND R. STERN, *Nonnegative Matrices in Dynamic Systems*, Wiley-Interscience, New York, 1989.

[2] A. BERMAN AND R. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[3] T. F. COLEMAN AND C. HEMPEL, *Computing a trust region step for a penalty method*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 180–201.

[4] R. FLETCHER, *Practical Methods of Optimization*, Wiley-Interscience, New York, 1987.

[5] G. E. FORSYTHE AND G. H. GOLUB, *On the stationary values of a second-degree polynomial on the unit sphere*, J. SIAM, 13 (1965), pp. 1050–1068.

[6] W. GANDER, *Least squares with a quadratric constraint*, Numer. Math., 36 (1981), pp. 291–307.

[7] W. GANDER, G. H. GOLUB, AND U. VON MATT, *A constrained eigenvalue problem*, Linear Algebra Appl., 114 (1989), pp. 815–839.

[8] F. GANTMACHER, *The Theory of Matrices*, Vol. 1, Chelsea Publishing, New York, 1959.

[9] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.

[10] G. H. GOLUB AND U. VON MATT, *Quadratically contrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.

[11] A. HORN, *Doubly stochastic matrices and the diagonal of a rotation matrix*, Amer. J. Math., 76 (1954), pp. 620–630.

[12] R. HORN AND C. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[13] H. LANGER AND B. NAJMAN, *Some interlacing results for indefinite Hermitian matrices*, Linear Algebra Appl., 69 (1985), pp. 131–154.

[14] R. LOEWY AND H. SCHNEIDER, *Positive operators on the n-dimensional ice-cream cone*, J. Math. Anal. Appl., 49 (1975), pp. 375–392.

[15] A. W. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, New York, 1979.

[16] L. MIRSKY, *Inequalities for normal and Hermitian matrices*, Duke J. Math., 24 (1957), pp. 591–599.

[17] ———, *Matrices with prescribed characteristic roots and diagonal elements*, J. London Math. Soc., 33 (1957), pp. 14–21.

[18] J. J. MORÉ AND D. SORENSEN, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[19] H. SCHNEIDER AND M. VIDYASAGAR, *Cross-positive matrices*, SIAM J. Numer. Anal., 7 (1970), pp. 508–519.

[20] I. SCHUR, *Über eine Klasse von Mittelbildungen mit Angwendungen die Determinanten*, Berlin Math. Ges., 22 (1923), pp. 9–20.

[21] D. SORENSEN, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[22] R. STERN AND H. WOLKOWICZ, *Exponential nonnegativity on the ice-cream cone*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 160–165.

[23] J. VANDERGRAFT, *Spectral properties of matrices which have invariant cones*, SIAM J. Appl. Math., 16 (1968), pp. 1208–1222.

[24] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford Science, Oxford, 1965.

[25] H. WOLKOWICZ AND G. P. H. STYAN, *Bounds for eigenvalues using traces*, Linear Algebra Appl., 29 (1980), pp. 471–506.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 779–795, July 1994

005

THE GENERALIZED ORDER LINEAR COMPLEMENTARITY PROBLEM*

M. SEETHARAMA GOWDA† AND ROMAN SZNAJDER†

Abstract. The generalized order linear complementarity problem (in the setting of a finite dimensional vector lattice) is the problem of finding a solution to the piecewise-linear system

$$x \wedge (M_1 x + q_1) \wedge (M_2 x + q_2) \wedge \cdots \wedge (M_k x + q_k) = 0,$$

where M_i's are linear transformations and q_i's are vectors. This problem is equivalent to the generalized linear complementarity problem considered by Cottle and Dantzig [*J. Combin. Theory*, 8 (1970), pp. 79–90.]. Using degree theory, a comprehensive analysis of existence, uniqueness, and stability aspects of this problem is presented.

Key words. order complementarity problem, piecewise-linear function, block transformations, type

AMS subject classifications. 90C33, 47H11, 46A40

1. Introduction.

For a given matrix $M \in R^{n\times n}$ and a vector $q \in R^n$, the (classical) linear complementarity problem, LCP(M, q), is to find a vector $x \in R^n$ such that

$$x \geq 0, \quad Mx + q \geq 0, \quad \text{and} \quad x^T (Mx + q) = 0. \tag{1}$$

The importance of this problem is well documented in the literature (see, e.g., [40] and [5]).

The above formulation of the LCP deals with the nonnegative orthant R^n_+ and the usual inner product. A generalization of the above problem, appropriately called the topological complementarity problem in [2], deals with a Hilbert space and a closed convex cone. This generalized problem has been well studied in the literature (see, e.g., [1], [7], [28]).

LCP(M, q) can be formulated equivalently as an equation: find x such that

$$x \wedge (Mx + q) = 0, \tag{2}$$

where $x \wedge (Mx + q) = (\min \{x_i, (Mx + q)_i\})$. This is an instance of an order complementarity problem (OCP). Since topology plays no role in this formulation, it is possible to define a generalization of this problem in the setting of a vector lattice. To define this generalization, consider a vector lattice X; that is, X is an ordered vector space in which for any two elements x and y, $x \wedge y = \min \{x, y\}$ and $x \vee y = \max \{x, y\}$ are defined [44]. Corresponding to any function $f : X \to X$, the OCP(f) is to find a vector x in X such that

$$x \wedge f(x) = 0. \tag{3}$$

Interesting results have been obtained in this general setting [1], [2], [18], and [19]. However, it is the mixing of topological and order structures that has yielded more (and perhaps better) results [7], [39], [46].

* Received by the editors October 1, 1992; accepted for publication (in revised form) January 21, 1993.

† Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, Maryland 21228 (gowda@math.umbc.edu and sznajder@math.umbc.edu).

The subject matter of our article is concerned with OCP(f) when f is a minimum of a finite number of affine functions. To be specific, let k be any natural number. Given k linear transformations $M_1, M_2, \ldots, M_k$ from X into itself and k vectors $q_1, q_2, \ldots, q_k$ in X, we write,

$$\mathbf{M} = (M_1, M_2, \ldots, M_k) \quad \text{and} \quad \mathbf{q} = (q_1, q_2, \ldots, q_k). \tag{4}$$

Then the generalized order linear complementarity problem GOLCP($\mathbf{M}, \mathbf{q}$) is to find a vector x in X such that

$$x \wedge (M_1 x + q_1) \wedge (M_2 x + q_2) \wedge \cdots \wedge (M_k x + q_k) = 0. \tag{5}$$

For $k = 1$, this problem was methodically analyzed by Borwein and Dempster [2]. Relying heavily on the order structure, they were able to get many existence results. They showed by an infinite dimensional example why uniqueness (of solution) for every solvable $\mathbf{q}$ does not assert the solvability of all $\mathbf{q}$'s. We show in this article that when X is finite dimensional, as in the classical situation, uniqueness for every solvable $\mathbf{q}$ implies the solvability of every $\mathbf{q}$. In fact, as in the classical situation, we tie this property to certain $\mathbf{P}$-matrices.

Of special importance is when X is the Euclidean space R^n with the usual ordering. It is shown in §6 that the above GOLCP is equivalent to the generalized linear complementarity problem (henceforth called the vertical linear complementarity problem (VLCP), as in [5]) considered by Cottle and Dantzig [4], who established basic existence results via (a modification of) Lemke's algorithm. The uniqueness aspect for the VLCP was considered by Szanc [53]. In recent times, a number of articles dealing with GOLCP (and equivalently, VLCP) have been written [14], [15], [17], [31].

Working still in the Euclidean space R^n with the usual order, we obtain a general problem when x appearing at the beginning of the expression (5) is replaced by an affine function. The resulting piecewise-linear system is an extended GOLCP: find $x \in R^n$ such that

$$(M_0 x + q_0) \wedge (M_1 x + q_1) \wedge (M_2 x + q_2) \wedge \cdots \wedge (M_k x + q_k) = 0. \tag{6}$$

It is clear that if M_0 is invertible, then (6) can be formulated as a GOLCP. This system occurs in certain nonlinear networks [20], [21] and in control theory [50], [51]. Eaves [10] gives a complementary pivoting algorithm to solve the piecewise-linear system (6), and Pang [43] studies this system by formulating it as an implicit complementarity problem. The general problem was also studied by Mangasarian [37], who established the equivalence of this to a certain linear programming problem. A sufficient condition for uniqueness in the extended GOLCP was provided by Isac and Goeleven [31]. We say more about the extended GOLCP in §7.

Our main object in this article is to give a comprehensive analysis of existence, uniqueness, and stability issues connected with the GOLCP. Traditionally, the existence results in the LCP theory are derived via the basic theorem of complementarity [25]. This basic theorem itself is obtained via the theorem of Hartman–Stampacchia on variational inequalities. It is not clear to the authors whether GOLCP can be formulated as a variational inequality problem. As we see, the degree theory approach allows us to prove existence results directly and under relaxed assumptions. Degree theory plays an important role in stability results as well. Some of the results proved in this paper are new even in the LCP setting.

Here is a brief description of various sections. In §3, we introduce block transformations of various types and prove existence results in a general setting. Section

4 deals with the corresponding uniqueness results. Section 5 deals with the stability aspects. In §6, we deal with the usual ordering on R^n and show the equivalence of the GOLCP with the generalized LCP of Cottle and Dantzig. Furthermore, in various subsections of §6, we show how to recover the results of Cottle–Dantzig, Szanc, Ebiefung–Kostreva, and others. Finally, in §7, we deal with the extended GOLCP and completely characterize uniqueness in such problems.

2. Preliminaries. Throughout this paper, we assume that X is a finite dimensional vector lattice. This means that there is a closed convex cone K that induces the order on X: for any two elements x and y, $x \leq y$ if and only if $y - x \in K$ and $x \vee y := \max\{x, y\}$ and $x \wedge y := \min\{x, y\}$ exist in X. Although X is isomorphic to R^n with the usual ordering (cf. [44, Prop., p. 9]), to simplify the notation (and with possible generalizations to infinite dimensional spaces in mind), we work with X and its ordering. The reader can refer to [44] or [36] for properties of order and lattice on a vector space. Let $\mathbf{Y}$ denote the Cartesian product of k copies of X, that is, $\mathbf{Y} = \prod_1^k X$. Because X and $\mathbf{Y}$ are finite dimensional, we may suppose that they have norms defined on them; we denote both norms by the same symbol $\|\cdot\|$. Let $\mathcal{B}$ denote the open unit ball in X. Since $x = x^+ - x^-$ where $x^+ = \max\{x, 0\}$ and $x^- = \max\{-x, 0\}$, we see that $X = K - K$, and so by convex analysis [49], the interior of K is nonempty. To indicate that an element d of X belongs to the interior of K, we use the notation $d \succ 0$. (While dealing with the usual ordering, we use the standard notation $d > 0$ instead of $d \succ 0$. Note, however, that the notation $d > 0$ in the context of a general vector lattice means that $d \geq 0$ and $d \neq 0$ [44].) For a $\mathbf{d} = (d_1, d_2, \ldots, d_k) \in \mathbf{Y}$, we write $\mathbf{d} \succ \mathbf{0}$ when $d_i \succ 0$ for all i and write $\mathbf{d} \geq 0$ when $d_i \geq 0$ for all i. In what follows, $\mathbf{M}$ and $\mathbf{q}$ are given by (4). For brevity, we write

$$x \wedge (\mathbf{M}x + \mathbf{q}) := x \wedge (M_1 x + q_1) \wedge (M_2 x + q_2) \wedge \cdots \wedge (M_k x + q_k),$$

$$x \wedge \mathbf{M}x := x \wedge M_1 x \wedge M_2 x \wedge \cdots \wedge M_k x \quad \text{and} \quad x \vee \mathbf{M}x := x \vee M_1 x \vee M_2 x \vee \cdots \vee M_k x.$$

We use the notation $\mathbf{M}x + \mathbf{q} \geq 0$ ($\succ 0$) to mean $M_i x + q_i \geq 0$ (respectively, $\succ 0$) for all $i = 1, 2, \ldots, k$. A vector x with $x \geq 0, \mathbf{M}x + \mathbf{q} \geq 0$ ($\succ 0$) is called a *feasible* (respectively, *strictly feasible*) vector for GOLCP$(\mathbf{M}, \mathbf{q})$. If there is such a vector, we say that GOLCP$(\mathbf{M}, \mathbf{q})$ is feasible (respectively, strictly feasible). Let

$$\mathcal{F}(\mathbf{M}) = \{\mathbf{q} : \mathbf{M}x + \mathbf{q} \geq 0 \text{ for some } x \geq 0\} \quad \text{and} \quad \mathcal{K}(\mathbf{M}) = \{\mathbf{q} : \mathrm{SOL}(\mathbf{M}, \mathbf{q}) \neq \emptyset\},$$

where SOL$(\mathbf{M}, \mathbf{q})$ denotes the solution set of GOLCP$(\mathbf{M}, \mathbf{q})$.

Borrowing the terminology from the LCP theory, we say that these sets are, respectively (when $\mathbf{M}$ is fixed), the set of all "feasible" $\mathbf{q}$'s and "solvable" $\mathbf{q}$'s. Note that $\mathcal{F}(\mathbf{M})$ is closed, convex, and

$$\operatorname{int} \mathcal{F}(\mathbf{M}) = \{\mathbf{q} : \mathbf{M}x + \mathbf{q} \succ 0 \text{ for some } x \geq 0\}.$$

3. Existence results. As in the LCP theory, our existence results deal with classes of (block) transformations $\mathbf{M}$. Motivated by the LCP theory, we introduce the following definition.

DEFINITION 1. *We say that* $\mathbf{M}$ *is of*

1. type $\mathbf{R}_0$ *if* $x \wedge \mathbf{M}x = 0 \Longrightarrow x = 0$;
2. type $\mathbf{G}$ *if for some* $\mathbf{d} \succ \mathbf{0}, \mathrm{SOL}(\mathbf{M}, \mathbf{d}) = \{0\}$;
3. type $\mathbf{R}$ *if it is of* type $\mathbf{R}_0$ *and* type $\mathbf{G}$;
4. type $\mathbf{E}$ *if* $x \geq 0, x \wedge \mathbf{M}x \leq 0 \Longrightarrow x = 0$;

5. type $\mathbf{P}$ *if* $x \wedge \mathbf{M}x \leq 0 \leq x \vee \mathbf{M}x \Longrightarrow x = 0$;
6. type $\mathbf{Q}$ *if for every* $\mathbf{q}$, GOLCP($\mathbf{M}$, $\mathbf{q}$) *has a solution*;
7. type $\mathbf{Q}_0$ *if* $\mathcal{F}(\mathbf{M}) \subseteq \mathcal{K}(\mathbf{M})$.

Remarks. We use the same symbol to denote the class of $\mathbf{M}$'s of a given type. For example, type $\mathbf{Q}$ denotes the set of all $\mathbf{M}$'s which are of type $\mathbf{Q}$. When X is R^n with the usual ordering and $k = 1$, the definitions 1, 2, 3, 6, and 7, reduce, respectively, to those of $\mathbf{R}_0$-matrices, $\mathbf{G}$-matrices, $\mathbf{R}$-matrices, $\mathbf{Q}$-matrices, and $\mathbf{Q}_0$-matrices [5]; definitions 4 and 5 reduce to those of strictly semimonotone and $\mathbf{P}$-matrices, respectively (see Theorems 3 and 4 below).

To illustrate the above definitions, we give two examples dealing with R^2 and the usual ordering. More examples are given in §6.

Example 1. Let $\mathbf{M} = (M_1, M_2)$ where

$$M_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \qquad M_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

It is easily verified that $\mathbf{M}$ is of type $\mathbf{R}_0$. Note that neither M_1 nor M_2 is an $\mathbf{R}_0$-matrix.

Example 2. Let $\mathbf{M} = (M_1, M_2)$ where

$$M_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad M_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

It is easily verified that $\mathbf{M}$ is of type $\mathbf{E}$ and hence of types $\mathbf{R}$ and $\mathbf{R}_0$ by Proposition 1 below.

PROPOSITION 1. type $\mathbf{P} \subseteq$ type $\mathbf{E} \subseteq$ type $\mathbf{R} \subseteq$ type $\mathbf{R}_0$.

Proof. If $x \geq 0$, then for any vector $y, 0 \leq x \vee y$. This observation shows that type $\mathbf{P} \subseteq$ type $\mathbf{E}$. Since $x \geq 0$ and $x \wedge \mathbf{M}x \leq 0$ when $x \wedge (\mathbf{M}x + t\mathbf{d}) = 0, t \in R_+, \mathbf{d} \succ 0$, we have the inclusion type $\mathbf{E} \subseteq$ type $\mathbf{R}$. The last inclusion type $\mathbf{R} \subseteq$ type $\mathbf{R}_0$ follows from the definition. □

Corresponding to the problem GOLCP($\mathbf{M}, \mathbf{q}$), we define two functions $F_{(\mathbf{M},\mathbf{q})}$ and $F_{\mathbf{M}}$ by

$$F_{(\mathbf{M},\mathbf{q})}(x) = x \wedge (\mathbf{M}x + \mathbf{q}) \quad \text{and} \quad F_{\mathbf{M}}(x) = x \wedge \mathbf{M}x.$$

We see that the problem GOLCP($\mathbf{M}, \mathbf{q}$) can be formulated as an equation $F_{(\mathbf{M},\mathbf{q})}(x) = 0$. It is now possible to study the GOLCP using degree theory. Suppose that $\mathbf{M}$ is of type $\mathbf{R}_0$ so that the zero vector is the only solution of $F_{\mathbf{M}}(x) = 0$. Let Ω be any bounded open set in X containing the zero vector. Then the integer $\deg(F_{\mathbf{M}}, \Omega, 0)$ (the degree of $F_{\mathbf{M}}$ over Ω relative to zero) is defined [35]. Furthermore, this is independent of the bounded open set Ω. We call this integer the GOLCP-degree of $\mathbf{M}$ and denote it by GOLCP-deg $\mathbf{M}$.

We remark that in the context of the (classical) LCP, the GOLCP-degree (simply called the LCP-degree) of an $\mathbf{R}_0$-matrix M can also be computed by using the mapping $x \mapsto x^+ - Mx^-$ [5, Chap. 6], [12].

Before stating our first existence result, we give a necessary and sufficient condition for boundedness of SOL ($\mathbf{M}, \mathbf{q}$) for all $\mathbf{q}$. We omit the proof as it is identical to the one in the classical case. Note that the solution set SOL ($\mathbf{M}, \mathbf{q}$) may be empty for some particular $\mathbf{q}$.

PROPOSITION 2. $\mathbf{M}$ *is of* type $\mathbf{R}_0$ *if and only if for all* $\mathbf{q}$, SOL ($\mathbf{M}$, $\mathbf{q}$) *is bounded.*

Remark. When $\mathbf{M}$ is of type $\mathbf{R}_0$, it is easily seen that the solution sets of GOLCP($\mathbf{M}, \mathbf{q}$) are uniformly bounded as $\mathbf{q}$ varies over a bounded set in $\mathbf{Y}$.

Here is our first existence result.

THEOREM 1. *Suppose that* $\mathbf{M}$ *is of* type $\mathbf{R}_0$ *and* GOLCP-degree *of* $\mathbf{M}$ *is nonzero. Then* $\mathbf{M}$ *is of type* $\mathbf{Q}$.

Proof. Let $\mathbf{q}$ be arbitrary. For any t in the interval [0, 1], consider

$$F_{(\mathbf{M},t\mathbf{q})}(x) = x \wedge (\mathbf{M}x + t\mathbf{q}).$$

In view of the remark above, the set $\{x : F_{(\mathbf{M},t\mathbf{q})}(x) = 0 \text{ for some } t \in [0,1]\}$ is bounded in X and hence contained in some bounded open set, say, Ω. Note that $F_{(\mathbf{M},t\mathbf{q})}$ is a homotopy connecting the mappings $F_{(\mathbf{M},\mathbf{q})}$ and $F_{\mathbf{M}}$. Using the homotopy invariance property of the degree [35, Thm. 2.1.2], we see that $\deg(F_{(\mathbf{M},\mathbf{q})}, \Omega, 0) = \deg(F_{\mathbf{M}}, \Omega, 0) =$ GOLCP-deg$\mathbf{M}$. Since this last integer is assumed to be nonzero, by the well-known property of the degree [35, Thm. 2.1.1], the equation $F_{(\mathbf{M},\mathbf{q})}(x) = 0$ has a solution in Ω, i.e., GOLCP$(\mathbf{M}, \mathbf{q})$ has a solution. Since $\mathbf{q}$ is arbitrary, the result follows. □

Our next result is the GOLCP analog of Karamardian's result [33] for regular matrices. Karamardian based his proof on an existence theorem of Hartman–Stampacchia on variational inequalities, which in turn is based on the Brouwer fixed point theorem. The proof of our result below is based on degree theory.

THEOREM 2. *Suppose that* $\mathbf{M}$ *is of* type $\mathbf{R}$. *Then its* GOLCP-degree *is one and hence it is of* type $\mathbf{Q}$. *In particular, if* $\mathbf{M}$ *is of* type $\mathbf{P}$ *or of* type $\mathbf{E}$, *then it is of* type $\mathbf{Q}$.

Proof. For t in the interval [0, 1], consider

$$F_{(\mathbf{M},t\mathbf{d})}(x) = x \wedge (\mathbf{M}x + t\mathbf{d}),$$

where $\mathbf{d}$ is as in the definition of the class type $\mathbf{R}$. As in the proof of the previous theorem, we have GOLCP-deg $\mathbf{M} = \deg(F_{\mathbf{M}}, \Omega, 0) = \deg(F_{(\mathbf{M},\mathbf{d})}, \Omega, 0)$, where Ω is any bounded open set containing zero. Since $\mathbf{d} \succ 0$, we have for all x near zero, $\mathbf{M}x + \mathbf{d} \succ 0$ and $F_{(\mathbf{M},\mathbf{d})}(x) = x$. This implies that $\deg(F_{\mathbf{M}}, \Omega, 0) = 1$; cf. [35, Thm. 1.1.4]. Now Theorem 1 completes the proof. □

4. Uniqueness results. The following theorem characterizes uniqueness in GOLCPs. Another equivalent characterization using $\mathbf{P}$-matrices is given in §6.

THEOREM 3. *The following statements are equivalent*:

(a) $\mathbf{M}$ *is of* type $\mathbf{P}$;

(b) *for every* $\mathbf{q}$, GOLCP$(\mathbf{M}, \mathbf{q})$ *has at most one solution*;

(c) *for every* $\mathbf{q}$, GOLCP$(\mathbf{M}, \mathbf{q})$ *has exactly one solution*;

Proof. The proof of the equivalence (a) $\iff$ (b) is standard (see, e.g., [2, Thm. 2.14]); for the sake of completeness, we include a proof. Assume (a). If x and y are two solutions of GOLCP$(\mathbf{M}, \mathbf{q})$, then with $z = x - y$, we easily verify that

$$z \wedge \mathbf{M}z \leq 0 \leq z \vee \mathbf{M}z, \tag{7}$$

which, in view of (a) gives $z = 0$. Hence, (a) $\Longrightarrow$ (b). Now suppose that (a) fails to hold. Let $z \neq 0$ satisfy (7). Define $\mathbf{q}$ by $q_j = (M_j z)^+ - M_j(z^+)$ $(j = 1, 2, \ldots, k)$ and note that $q_j = (M_j z)^- - M_j(z^-)$ $(j = 1, 2, \ldots, k)$. Since (7) is equivalent to $z^+ \wedge (M_1 z)^+ \wedge \cdots \wedge (M_k z)^+ = 0 = z^- \wedge (M_1 z)^- \wedge \cdots \wedge (M_k z)^- = 0$, we see that z^+ and z^- are two (distinct) solutions of GOLCP$(\mathbf{M}, \mathbf{q})$ contradicting (b). Hence, (b) $\Longrightarrow$ (a). Clearly, (c) $\Longrightarrow$ (b) $\Longrightarrow$ (a). From the implication (a) $\Longrightarrow$ (b) and Theorem 2, we get (a) $\Longrightarrow$ (c). This completes the proof. □

Remarks. As pointed out in the proof, the implication (a) $\Longrightarrow$ (b) was proved by Borwein and Dempster. They also give an example to show that in the infinite dimensional setting, the converse (b) $\Longrightarrow$ (a) is false.

THEOREM 4. *The following statements are equivalent:*

(a) $\mathbf{M}$ *is of* type $\mathbf{E}$;

(b) *For every* $\mathbf{q} \geq \mathbf{0}$, *the zero vector is the only solution* GOLCP$(\mathbf{M}, \mathbf{q})$.

Proof. Suppose (a) holds and x is a solution of GOLCP$(\mathbf{M}, \mathbf{q})$ where $\mathbf{q} \geq \mathbf{0}$. It follows that $x \geq 0$ and $x \wedge (\mathbf{M}x) \leq x \wedge (\mathbf{M}x + \mathbf{q}) = 0$. We see that $x = 0$. On the other hand, suppose (b) holds and let z be a solution of the system $z \geq 0, z \wedge \mathbf{M}z \leq 0$. Then z is a solution of GOLCP$(\mathbf{M}, \mathbf{q})$ where $\mathbf{q}$ is defined by $q_j = (M_j z)^+ - (M_j z)$ $(j = 1, 2, \ldots, k)$. Since $\mathbf{q} \geq \mathbf{0}$, we get $z = 0$. □

In the classical LCP theory, statement (b) in Theorem 4 describes strictly semimonotone matrices (defined by the condition: for all $0 \neq x \geq 0$, there is an index i such that $x_i(Mx)_i > 0$). Analogous to the classes of semimonotone matrices and $\mathbf{P}_0$-matrices, we introduce two classes of block transformations $\mathbf{M}$. Let I denote the identity mapping on X and let

$$\mathbf{M} + \varepsilon\mathbf{I} := (M_1 + \varepsilon I, M_2 + \varepsilon I, \ldots, M_k + \varepsilon I).$$

We say that $\mathbf{M}$ is of type $\mathbf{E}_0$ ($\mathbf{P}_0$) if $\mathbf{M} + \varepsilon\mathbf{I}$ is of type $\mathbf{E}$ (respectively, type $\mathbf{P}$) for every $\varepsilon > 0$. It is clear from Theorems 3 and 4 that type $\mathbf{P}_0 \subseteq$ type $\mathbf{E}_0$. We have the following theorem.

THEOREM 5. *Consider the statements:*

(a) $\mathbf{M}$ *is of* type $\mathbf{E}_0$;

(b) *for every* $\mathbf{q} \succ \mathbf{0}$, *the zero vector is the only solution of* GOLCP$(\mathbf{M},\mathbf{q})$.

Then (a) $\Longrightarrow$ (b).

Remark. We show in §6.3 that the reverse implication (b) $\Longrightarrow$ (a) holds when X is R^n with the usual ordering. Isomorphism considerations then allow us to get the reverse implication in the general case as well.

Proof. Suppose $\mathbf{q} \succ \mathbf{0}$ and $x \wedge (\mathbf{M}x + \mathbf{q}) = 0$. Let $\mathbf{x} := (x, x, \ldots, x)$ and $\varepsilon > 0$ be small so that $\mathbf{q} - \varepsilon\mathbf{x} \succ \mathbf{0}$. Then $x \geq 0$ and $x \wedge (\mathbf{M}x + \varepsilon x) \leq x \wedge (\mathbf{M}x + \mathbf{q}) = 0$, which implies that $x = 0$ in view of (a). □

Remark. The above theorem shows that type $\mathbf{E}_0 \subseteq$ type $\mathbf{G}$ and therefore type $\mathbf{E}_0 \cap$ type $\mathbf{R}_0 \subseteq$ type $\mathbf{R}$.

5. Stability. In the stability aspect of GOLCP, we are interested in the solution behavior as the data changes. First, we deal with the behavior of the entire solution set. In the definition below, $\|\mathbf{M}\|$ refers to the norm of $\mathbf{M}$ as a linear operator from $\mathbf{Y}$ into itself; however, we can use any norm.

DEFINITION 2. *The problem* GOLCP$(\mathbf{M}, \mathbf{q})$ *is said to be stable if for every* $\varepsilon > 0$ *there exists a* $\delta > 0$ *such that*

$$\mathrm{SOL}\,(\mathbf{M}', \mathbf{q}') \cap (\mathrm{SOL}(\mathbf{M}, \mathbf{q}) + \varepsilon\mathcal{B}) \neq \emptyset$$

for all $(\mathbf{M}', \mathbf{q}')$ *with* $\|\mathbf{M}' - \mathbf{M}\| + \|\mathbf{q}' - \mathbf{q}\| < \delta$.

The following theorem is the basis of our stability analysis.

THEOREM 6. *Suppose that* SOL$(\mathbf{M}, \mathbf{q})$ *is nonempty and bounded. Suppose that for some open set* Ω *containing* SOL$(\mathbf{M}, \mathbf{q})$, $\deg(F_{(\mathbf{M},\mathbf{q})}, \Omega, 0)$ *is nonzero. Then* GOLCP$(\mathbf{M}, \mathbf{q})$ *is stable.*

Proof. For any given ε, we consider the open set $\mathcal{D} = \mathrm{SOL}(\mathbf{M}, \mathbf{q}) + \varepsilon\mathcal{B}$. Without loss of generality, we can assume that $\mathcal{D} \subset \Omega$. Since there are no solutions of

$F_{(\mathbf{M},\mathbf{q})}(x) = 0$ in $\Omega \setminus \mathcal{D}$, by the excision property of the degree, we have deg $(F_{(\mathbf{M},\mathbf{q})}, \mathcal{D}, 0) = \deg\,(F_{(\mathbf{M},\mathbf{q})}, \Omega, 0) \neq 0$. Now for a suitable $\delta > 0$, we have

$$\sup_{\mathcal{D}} \left\| F_{(\mathbf{M},\mathbf{q})}\,(x) - F_{(\mathbf{M}',\mathbf{q}')}\,(x) \right\| < \operatorname{dist}\left(0, F_{(\mathbf{M},\mathbf{q})}\,(\partial D)\right)$$

for all $(\mathbf{M}', \mathbf{q}')$ with $\|\mathbf{M}' - \mathbf{M}\| + \|\mathbf{q}' - \mathbf{q}\| < \delta$. By the nearness property of the degree [35, Thm. 2.1.2], $\deg(F_{(\mathbf{M}',\mathbf{q}')}, \mathcal{D}, 0) = \deg(F_{(\mathbf{M},\mathbf{q})}, \mathcal{D}, 0) \neq 0$. Hence, the equation $F_{(\mathbf{M}',\mathbf{q}')}(x) = 0$ has a solution in $\mathcal{D}$. The stated conclusion follows. □

To illustrate the above theorem as well as for future reference, we formulate a definition. We say that $\mathbf{M}$ is of type $\mathbf{D}$ if there is a $\mathbf{d}$ such that (a) $\mathrm{SOL}(\mathbf{M}, \mathbf{d})$ is nonempty and bounded, and (b) $\deg(F_{(\mathbf{M},\mathbf{d})}, \Omega, 0) \neq 0$, where Ω is some bounded open set containing $\mathrm{SOL}(\mathbf{M}, \mathbf{d})$.

Note that these two conditions imply the stability of $\mathrm{GOLCP}(\mathbf{M}, \mathbf{d})$. In particular, when $\mathbf{M}$ is of type $\mathbf{D}$, the vector $\mathbf{d}$ that appears in the definition of $\mathbf{D}$ is necessarily in the interior of $\mathcal{F}(\mathbf{M})$. Now recall that $\mathbf{M}$ is of type $\mathbf{G}$ if there is a $\mathbf{d} \succ 0$ such that $\mathrm{SOL}(\mathbf{M}, \mathbf{d}) = \{0\}$. It follows that for such an $\mathbf{M}, F_{(\mathbf{M},\mathbf{d})}(x) = x$ for all x near the zero vector. So, $\deg(F_{(\mathbf{M},\mathbf{d})}, \Omega, 0) = 1$ where Ω is any bounded open set containing the zero vector in R^n. This establishes the inclusion type $\mathbf{G} \subseteq$ type $\mathbf{D}$.

The theorem given below is new even in the classical LCP theory. For positive semidefinite matrices, this result was proved in the LCP setting by Robinson [47] via generalized equations and maximal monotone multifunctions. Using degree theory, Gowda and Pang presented various generalizations of Robinson's result for the affine variational inequality problem [27]. As we see below, the Gowda–Pang analysis goes through even for GOLCP.

THEOREM 7. *Let* $\mathbf{M}$ *be of* type $\mathbf{P}_0$ *and* $\mathrm{SOL}(\mathbf{M}, \mathbf{q})$ *be nonempty and bounded. Then* $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$ *is stable.*

Proof. Let Ω be a bounded open set in X containing $\mathrm{SOL}(\mathbf{M}, \mathbf{q})$. Let $x^* \in \mathrm{SOL}(\mathbf{M}, \mathbf{q})$. For any t with $0 \leq t \leq 1$, define $(\mathbf{M}_t, \mathbf{q}_t)$ by

$$\mathbf{M}_t = \mathbf{M} + t\mathbf{I}, \qquad \mathbf{q}_t = \mathbf{q} - t\mathbf{x}^*,$$

where $\mathbf{x}^* := (x^*, x^*, \ldots, x^*)$. Then for $t > 0, x^* \in \mathrm{SOL}(\mathbf{M}_t, \mathbf{q}_t)$ and $\mathbf{M}_t$ is of type $\mathbf{P}$. Therefore, $\mathrm{SOL}(\mathbf{M}_t, \mathbf{q}_t) = \{x^*\}$ for all $t > 0$. Clearly, we have a homotopy between $(\mathbf{M}, \mathbf{q})$ and $(\mathbf{M}_1, \mathbf{q}_1)$. Hence, $\deg(F_{(\mathbf{M},\mathbf{q})}, \Omega, 0) = \deg(F_{(\mathbf{M}_1,\mathbf{q}_1)}, \Omega, 0)$. Since $\mathbf{M}_1$ is of type $\mathbf{P}$, it is of type $\mathbf{R}$ by Proposition 1. By Theorem 2, its GOLCP-degree is one. By following the proof of Theorem 1, we see that $\deg(F_{(\mathbf{M}_1,\mathbf{q}_1)}, \Omega, 0) = 1$. Therefore, $\deg(F_{(\mathbf{M},\mathbf{q})}, \Omega, 0) \neq 0$. Now the stability follows from the previous theorem. □

We now formulate the definition of stability at a solution point.

DEFINITION 3. *Let* x^* *be a solution of* $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$. *The problem* $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$ *is said to be stable at* x^* *if* x^* *is an isolated solution of* $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$ *and for every* $\varepsilon > 0$ *there exists a* $\delta > 0$ *such that*

$$\mathrm{SOL}\,(\mathbf{M}', \mathbf{q}') \cap (x^* + \varepsilon\mathcal{B}) \neq \emptyset$$

for all $(\mathbf{M}', \mathbf{q}')$ *with* $\|\mathbf{M}' - \mathbf{M}\| + \|\mathbf{q}' - \mathbf{q}\| < \delta$.

In the case of the classical LCP, the stability at a solution point is fairly well understood [23]–[26], [29], [30]. Here, we present a result for GOLCP. By modifying the proofs of Theorems 6 and 7, we are led to the following. We omit the details.

THEOREM 8. *Suppose that* x^* *is an isolated solution of* $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$. *Suppose that there is a bounded open set* Ω *containing only one solution of* $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$, *namely,* x^* *and* $\deg(F_{(\mathbf{M},\mathbf{q})}, \Omega, 0)$ *is nonzero. Then* $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$ *is stable at* x^*.

THEOREM 9. *Let* $\mathbf{M}$ *be of type* $\mathbf{P}_0$ *and let* x^* *be an isolated solution of* GOLCP$(\mathbf{M}, \mathbf{q})$. *Then* GOLCP$(\mathbf{M}, \mathbf{q})$ *is stable at* x^*.

An interesting consequence can be obtained from the above theorem. Suppose that $\mathbf{M}$ is of type $\mathbf{P}_0$ and there are two isolated solutions, say, x^* and y^* for GOLCP$(\mathbf{M}, \mathbf{q})$. From the above theorem, GOLCP$(\mathbf{M}, \mathbf{q})$ is stable at both the solutions. But if $\mathbf{M}$ is perturbed to $\mathbf{M}'$, which is of type $\mathbf{P}$, then the problem GOLCP$(\mathbf{M}', \mathbf{q})$ must have solutions near x^* as well as y^*. This clearly contradicts the uniqueness property of $\mathbf{M}'$. We conclude that for GOLCP$(\mathbf{M}, \mathbf{q})$, there can be at most one isolated solution. This means that *when* $\mathbf{M}$ *is of type* $\mathbf{P}_0$ *and* $\mathbf{q}$ *is arbitrary,* SOL$(\mathbf{M}, \mathbf{q})$ *either is empty, or a singleton set, or an infinite set.*

In the LCP setting, Theorem 9 reduces to a result of Gowda [23]. In [26], Gowda and Pang generalize this LCP result by imposing a condition only on a submatrix of the given matrix. Such a detailed analysis is certainly possible for GOLCP and is dealt with elsewhere.

6. Results for R^n with the usual ordering. In this section, we assume that X is R^n with the usual order. In this setting, we prove existence, uniqueness, and stability results. We think of vectors in R^n as column vectors. For $x \in R^n, x_i$ denotes the ith component. For $x, y \in R^n$, the Hadamard product $x * y$ is the vector whose ith component is $x_i y_i$. We recall that the (standard) notation $r > 0$ means that the vector r is positive ($r \succ 0$).

6.1. The VLCP. In this subsection, we describe the vertical LCP [5] and show that it is equivalent to GOLCP.

Consider a rectangular matrix N of order $m \times n$ with $m \geq n$, and let p be an m-vector. Suppose that N and p are partitioned in the form

$$N = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{bmatrix}, \qquad p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}, \tag{8}$$

where each $N_i \in R^{m_i \times n}$ and $p_i \in R^{m_i}$ with $\sum_{i=1}^{n} m_i = m$. Then VLCP(N, p) is to find a vector $z \in R^n$ such that

$$z \geq 0,\ Nz + p \geq 0, \quad \text{and} \quad z_i \prod_{j=1}^{m_i} (N_i z + p_i)_j = 0 \qquad i = 1, 2, \ldots, n. \tag{9}$$

We show that this problem can be formulated as a GOLCP. Let $k = \max\{m_i : i = 1, \ldots, n\}$. Let N_i^j denote the jth row of the matrix N_i. We define matrices $\tilde{N}_1, \tilde{N}_2, \ldots, \tilde{N}_n$ each of size $k \times n$ in the following way. For any i, the jth row of $\tilde{N}_i$ is N_i^j if $j \leq m_i$ and N_i^1 (the first row of N_i) if $j > m_i$. Correspondingly, we define vectors $\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_n$ each of size $k \times 1$. This construction leads to the rectangular (block) matrix $\tilde{N}$ and the vector $\tilde{p}$. It is clear that VLCP$(\tilde{N}, \tilde{p})$ is equivalent to the VLCP(N, p). Now let M_i be the matrix of size $n \times n$ whose jth row is the ith row in the matrix $\tilde{N}_j$. (For example, M_1 is formed by considering the first row in each of the matrices $\tilde{N}_1, \tilde{N}_2, \ldots, \tilde{N}_n$.) Similarly, let q_i be the vector of size $1 \times n$ whose jth component is the ith component in the vector $\tilde{p}_j$. It is easily verified that the VLCP$(\tilde{N}, \tilde{p})$ is equivalent to GOLCP$(\mathbf{M}, \mathbf{q})$.

To see that every GOLCP can be formulated as a VLCP, consider GOLCP$(\mathbf{M}, \mathbf{q})$. Let N_i be the $k \times n$-matrix whose jth row is the ith row of the matrix M_j. (For example,

N_1 is formed by considering the first row in each of the matrices $M_1, M_2, \ldots, M_k$.) Correspondingly, we define the $k \times n$ vector p_i. This construction leads to the pair (N, p), and the corresponding VLCP is easily seen to be equivalent to GOLCP$(\mathbf{M}, \mathbf{q})$.

6.2. Representative matrices. Let $(\mathbf{M}, \mathbf{q})$ be given by (4) where each M_i is an $n \times n$ real matrix and q_i is an $n \times n$ vector. We say that the pair $(A, a) \in R^{n \times n} \times R^n$ is a *representative* of $(\mathbf{M}, \mathbf{q})$ if for each $j = 1, 2, \ldots, n$, the jth row of A is the jth row of some M_i and jth component of a is the jth component of the corresponding q_i. In other words, $A^j \in \{M_1^j, M_2^j, \ldots, M_k^j\}$, where the superscript refers to the corresponding row vector, and so on. We say that matrix A is a *representative matrix* of $\mathbf{M}$ and a is a representative vector of $\mathbf{q}$.

It is immediate that $x \in \mathrm{SOL}(\mathbf{M}, \mathbf{q})$ if and only if x is a feasible vector for GOLCP$(\mathbf{M}, \mathbf{q})$ and $x \in \mathrm{SOL}(A, a)$ for some representative (A, a) of $(\mathbf{M}, \mathbf{q})$. We record this formally in the following proposition.

PROPOSITION 3. *It holds that*

$$\mathrm{SOL}\,(\mathbf{M}, \mathbf{q}) = \mathcal{N}\,(\mathbf{M}, \mathbf{q}) \cap (\cup\ \mathrm{SOL}\,(A, a))\,, \tag{10}$$

where

$$\mathcal{N}\,(\mathbf{M}, \mathbf{q}) = \{x\ :\ x \geq 0,\ \mathbf{M}x + \mathbf{q} \geq 0\}$$

and the union is over all representatives (A, a) *of* $(\mathbf{M}, \mathbf{q})$.

Proposition 3 leads to three important observations. First, the set $\mathrm{SOL}(\mathbf{M}, \mathbf{q})$ is a union of finite number of polyhedral (convex) sets. Second, $\mathcal{K}(\mathbf{M})$ is closed. (This follows from the facts that for any given $\mathbf{M}$, the set of all $\mathbf{q}$'s with $\mathcal{N}(\mathbf{M}, \mathbf{q}) \neq \emptyset$ is closed and in the case of classical LCP, for any given matrix A, the set of all a's with $\mathrm{SOL}(A, a) \neq \emptyset$ is closed.) Third, the graph of the mapping $\mathbf{\Phi} : \mathbf{q} \mapsto \mathrm{SOL}(\mathbf{M}, \mathbf{q})$ is a union of finite number of polyhedral sets, i.e., $\mathbf{\Phi}$ is a polyhedral multifunction [48]. This third observation leads to the following proposition.

PROPOSITION 4. *For any given* $\mathbf{M}$, *consider the mapping* $\mathbf{\Phi}(\mathbf{q}) := \mathrm{SOL}(\mathbf{M}, \mathbf{q})$. *Let* $\mathcal{E}$ *be a compact subset of* $\prod_1^k R^n$. *If* $\mathbf{\Phi}(\mathbf{q})$ *is bounded for each* $\mathbf{q} \in \mathcal{E}$, *then* $\mathbf{\Phi}$ *is (uniformly) bounded on* $\mathcal{E}$, *i.e., the set* $\bigcup_{\mathbf{q} \in \mathcal{E}} \mathbf{\Phi}(\mathbf{q})$ *is bounded.*

Proof. It follows from Robinson's locally upper Lipschitzian property of $\mathbf{\Phi}$ [48] that the mapping $\mathbf{\Phi}$ is upper semicontinuous at any $\mathbf{q}$, i.e., given any $\mathbf{q}$ and $\varepsilon > 0$, there exists a neighborhood $\mathbf{V}$ of $\mathbf{q}$ such that

$$\Phi\,(\mathbf{q}') \subseteq \Phi\,(\mathbf{q}) + \varepsilon \mathcal{B}$$

for all $\mathbf{q}'$ in $\mathbf{V}$ where $\mathcal{B}$ is the open unit ball in R^n. Since we are assuming that $\mathbf{\Phi}(\mathbf{q})$ is bounded for each $\mathbf{q} \in \mathcal{E}$, a standard argument, involving the compactness and upper semicontinuity notions, gives the desired result. □

We say that $\mathbf{M}$ has the $\mathbf{T}$-*property* if every representative matrix of $\mathbf{M}$ is a $\mathbf{T}$-matrix where $\mathbf{T}$ denotes a class of matrices. For example, $\mathbf{M}$ has the $\mathbf{P}$-property if every representative matrix of $\mathbf{M}$ is a $\mathbf{P}$-matrix. (Recall that a square matrix A is a $\mathbf{P}$-matrix if every principal minor of A is positive; equivalently [5], $z * Az \leq 0 \Longrightarrow z = 0$.) The standard classes in the LCP theory are the classes of copositive matrices, $\mathbf{R}_0$-matrices, $\mathbf{R}$-matrices, copositive-plus matrices, $\mathbf{E}_1$-matrices, and so on. We refer the reader to [5] for detailed information on these classes.

PROPOSITION 5. *If* $\mathbf{M}$ *has the* $\mathbf{R}_0$*-property, then it is of* type $\mathbf{R}_0$.

Proof. Suppose that $\mathbf{M}$ has the $\mathbf{R}_0$-property and let x be a vector such that $x \wedge \mathbf{M}x = 0$. Since the ordering is the usual ordering, this leads to $x_i \wedge (M_1 x)_i \wedge$

$(M_2x)_i \cdots \wedge (M_kx)_i = 0$ for each $i = 1, 2, \ldots, n$. Corresponding to say i, let $l(i)$ be an index such that $(M_{l(i)}x)_i = (M_1x)_i \wedge (M_2x)_i \cdots \wedge (M_kx)_i$. Let A be the representative matrix of $\mathbf{M}$ whose ith row is $M^i_{l(i)}$. Clearly, $x \wedge Ax = 0$. By assumption, A is an $\mathbf{R}_0$-matrix and so $x = 0$. Thus, $\mathbf{M}$ is of type $\mathbf{R}_0$. □

Remarks. Example 1 shows that the converse statement in the above proposition may not hold. There does not seem to be any connection between type $\mathbf{R}$ and the $\mathbf{R}$-property (see example below). However, suppose there is a positive vector e in R^n such that for every representative matrix A of $\mathbf{M}$ and for every $t \geq 0, \mathrm{SOL}(A, te) = \{0\}$. Then it is easily verified that $\mathbf{M}$ is of type $\mathbf{R}$. In particular, if $\mathbf{M}$ has the $\mathbf{E}_0$-property and the $\mathbf{R}_0$-property, then it is of type $\mathbf{R}$.

Before moving on to the uniqueness aspect, we present two important examples.

Example 3. Let $\mathbf{M} = (M_1, M_2)$ where

$$M_1 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix}, \quad \text{and} \quad e = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

It is easily verified that $\mathrm{SOL}(\mathbf{M}, t\mathbf{e}) = \{0\}$, where t is any nonnegative real number and $\mathbf{e} = (e, e)$. (While verifying this, we have used the fact that the matrix

$$A = \begin{bmatrix} 1 & -2 \\ 1 & -1 \end{bmatrix}$$

is an $\mathbf{R}$-matrix [5, p. 194].) This means that $\mathbf{M}$ is of type $\mathbf{R}$ and hence of type $\mathbf{Q}$ by Theorem 2. However, M_1 is neither an $\mathbf{R}_0$-matrix nor a $\mathbf{Q}$-matrix, i.e., $\mathbf{M}$ has neither the $\mathbf{R}$-property nor the $\mathbf{Q}$-property.

Example 4. Let $\mathbf{M} = (M_1, M_2)$ and $\mathbf{q} = (q_1, q_2)$, where

$$M_1 = \begin{bmatrix} 2 & -3 \\ 1 & -1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} -\frac{1}{2} & 1 \\ 2 & -1 \end{bmatrix}, \quad q_1 = \begin{bmatrix} 6 \\ 3 \end{bmatrix}, \quad \text{and} \quad q_2 = \begin{bmatrix} -1 \\ 3 \end{bmatrix}.$$

By employing complementary cones [5], it can be verified that $\mathbf{M}$ has the $\mathbf{Q}$-property but that $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$ has no solution, i.e., $\mathbf{M}$ is not of type $\mathbf{Q}$. Note also that $\mathbf{M}$ has the $\mathbf{R}_0$-property.

6.3. Uniqueness.

THEOREM 10. $\mathbf{M}$ *has the* $\mathbf{P}$*-property if and only if it is of* type $\mathbf{P}$.

Proof. Suppose that $\mathbf{M}$ is of type $\mathbf{P}$ and let A be a representative matrix of $\mathbf{M}$. To show that A is a $\mathbf{P}$-matrix, we start with a vector x such that $x_i(Ax)_i \leq 0$ for all i and show that $x = 0$. If $x_i > 0$, then $A^ix \leq 0$ and hence $(x \wedge \mathbf{M}x)_i \leq 0 \leq (x \vee \mathbf{M}x)_i$. If $x_i < 0$, then $A^ix \geq 0$ and hence $(x \wedge \mathbf{M}x)_i \leq 0 \leq (x \vee \mathbf{M}x)_i$. Finally, if $x_i = 0$, then $(x \wedge \mathbf{M}x)_i \leq 0 \leq (x \vee \mathbf{M}x)_i$. Therefore, $x \wedge \mathbf{M}x \leq 0 \leq x \vee \mathbf{M}x$. Since $\mathbf{M}$ is of type $\mathbf{P}, x = 0$. We have shown that each representative of $\mathbf{M}$ is a $\mathbf{P}$-matrix and so $\mathbf{M}$ has the $\mathbf{P}$-property.

To see the converse, assume that $\mathbf{M}$ has the $\mathbf{P}$-property. Let x be a vector such that $x \wedge \mathbf{M}x \leq 0 \leq x \vee \mathbf{M}x$. We construct a representative matrix A as follows. For any index j, let A^j be a vector in the set $\{M_1^j, M_2^j, \ldots, M_k^j\}$ such that $A^jx \leq 0$ if $x_j > 0$ and $A^jx \geq 0$ if $x_j \leq 0$. Since A is a $\mathbf{P}$-matrix (by assumption) and $x*(Ax) \leq 0$, we see that $x = 0$. Therefore $\mathbf{M}$ is of type $\mathbf{P}$. □

Remarks. Combining Theorems 3 and 10 we see that $\mathrm{GOLCP}(\mathbf{M}, \mathbf{q})$ *has a unique solution for all* $\mathbf{q}$ *if and only if every representative matrix of* $\mathbf{M}$ *is a* $\mathbf{P}$*-matrix.* This result for VLCPs was proved earlier by Cottle and Dantzig [4], who proved the existence of solution when $\mathbf{M}$ has the $\mathbf{P}$-property, and by Szanc [53], who proved that

$\mathbf{M}$ has the $\mathbf{P}$-property if and only if every VLCP corresponding to $\mathbf{M}$ has at most one solution.

By slightly modifying the proof of the above theorem, we can show that $\mathbf{M}$ is of type $\mathbf{E}$ (type $\mathbf{E}_0$, type $\mathbf{P}_0$) if and only if $\mathbf{M}$ has the $\mathbf{E}$-property (respectively, $\mathbf{E}_0$-property, $\mathbf{P}_0$-property). We omit the details.

We end this subsection by proving the reverse implication (b) $\Longrightarrow$ (a) in Theorem 5. Because of the previous remarks, we need to show that if (b) holds, then every representative matrix of $\mathbf{M}$ is an $\mathbf{E}_0$-matrix. Without loss of generality, consider the matrix M_1 in $\mathbf{M}$. Let $q_1 > 0$ in R^n. If the problem LCP(M_1, q_1) has two solutions, say, x and y, then we can take $r > 0$ and define $q_i = (-M_i x) \vee (-M_i y) \vee r$ for $i = 2, 3, \ldots, k$. Clearly, $\mathbf{q} \succ 0$ and GOLCP$(\mathbf{M}, \mathbf{q})$ has two solutions. So, if (b) holds, then LCP(M_1, q_1) has a unique solution for all $q_1 > 0$, that is, M_1 is an $\mathbf{E}_0$-matrix.

6.4. The $\mathbf{E}_1$-property. Our next two sections deal with existence results not covered by Theorems 1 and 2. As in the classical LCP, we deal with the classes of $\mathbf{E}_1$-matrices (also called $\mathbf{L}_2$-matrices) and copositive matrices. Recall that a matrix A is an $\mathbf{E}_1$-matrix if for each nonzero solution $v \in \mathrm{SOL}(A, 0)$, there exist two nonnegative diagonal matrices Γ and Λ such that $\Gamma v \neq 0$ and $A^T(\Gamma v) + \Lambda A v = 0$. In [4], Cottle and Dantzig show via a modification of Lemke's algorithm that when $\mathbf{M}$ has the copositive-plus property, feasibility of the VLCP implies its solvability. The extension of this result given below deals with an $\mathbf{M}$ that has the $\mathbf{E}_1$-property. In the classical setting, results for $\mathbf{E}_1$-matrices are proved using either the Lemke algorithm (as in Eaves [10] and [39]) or using the basic theorem of complementarity (as in [25] and [5]). As we see below, degree theory allows us to deal with the $\mathbf{E}_1$-property in a comprehensive way, even allowing a generalization of the classical LCP result of Moré [39].

THEOREM 11. *Suppose that* $\mathbf{M}$ *has the* $\mathbf{E}_1$*-property and is of* type $\mathbf{D}$. *Suppose further that* GOLCP$(\mathbf{M}, \mathbf{q})$ *is strictly feasible. Then* SOL$(\mathbf{M}, \mathbf{q})$ *is nonempty and bounded.*

Proof. We first show that for any $\mathbf{p} \in \mathrm{int}\, \mathcal{F}(\mathbf{M})$, SOL$(\mathbf{M}, \mathbf{p})$ is bounded. Assume the contrary. By Proposition 3, there exists a representative pair (A, a) of $(\mathbf{M}, \mathbf{p})$ such that $\mathcal{N}(\mathbf{M}, \mathbf{p}) \cap \mathrm{SOL}(A, a)$ is unbounded. Hence, there exists an $x \in \mathrm{SOL}(A, a)$ and a direction $v \neq 0$ such that $x + \lambda v \in \mathrm{SOL}(A, a)$ for all $\lambda \geq 0$, i.e., $(x + \lambda v) \wedge \{A(x + \lambda v) + a\} = 0$ for all $\lambda \geq 0$. It is easily verified that $v \wedge Av = 0, x * Av = 0$, and $v * (Ax + a) = 0$. Since $\mathbf{M}$ has the $\mathbf{E}_1$-property, A is an $\mathbf{E}_1$-matrix and so there exists two nonnegative diagonal matrices Λ and Γ such that $\Gamma v \neq 0$ and $A^T(\Gamma v) + \Lambda A v = 0$. It follows from $v * (Ax + a) = 0$ that $(\Gamma v)^T (Ax + a) = 0$. Since $\mathbf{M}s + \mathbf{p} \succ 0$ for some $s \geq 0$, we can write $a = r - As$ with $r > 0$. We now have

$$0 = -(\Lambda A v)^T x + (\Gamma v)^T r + (\Lambda A v)^T s.$$

In view of $x * Av = 0$, the first term in the above equality is zero. Since Av and Λ are nonnegative, the third term is nonnegative. Since $\Gamma v \neq 0$ and $r > 0$, we reach a contradiction. Hence, SOL$(\mathbf{M}, \mathbf{p})$ is bounded.

To show that SOL$(\mathbf{M}, \mathbf{q})$ is nonempty, we consider the homotopy defined by $\mathbf{q}_t := (1 - t)\mathbf{d} + t\mathbf{q}$, where $\mathbf{d}$ is the vector that corresponds to $\mathbf{M}$ in the definition of $\mathbf{D}$. It is clear that $\mathbf{q}_t \in \mathrm{int}\, \mathcal{F}(\mathbf{M})$ for all t and (by the above argument) SOL$(\mathbf{M}, \mathbf{q}_t)$ is bounded for each t. By Proposition 4, SOL$(\mathbf{M}, \mathbf{q}_t)$ is uniformly bounded, say, by an open set Ω' that includes the open set Ω appearing in the definition of $\mathbf{d}$. By the homotopy invariance and the excision properties of the degree, $\deg(F_{(\mathbf{M},\mathbf{q})}, \Omega', 0) =$

$\deg(F_{(\mathbf{M},\mathbf{d})}, \Omega, 0) \neq 0$. This means that the equation $F_{(\mathbf{M},\mathbf{q})}(x) = 0$ has a solution in Ω', i.e., GOLCP$(\mathbf{M}, \mathbf{q})$ has a solution. This completes the proof. □

THEOREM 12. *Suppose that* $\mathbf{M}$ *has the* $\mathbf{E}_1$*-property and is of* type $\mathbf{D}$. *Then* $\mathbf{M}$ *is of* type $\mathbf{Q}_0$, *i.e., for all* $\mathbf{q} \in \mathcal{F}(\mathbf{M}), \mathrm{SOL}(\mathbf{M}, \mathbf{q})$ *is nonempty. Moreover, the following are equivalent for any* $\mathbf{q}$:

(a) GOLCP$(\mathbf{M}, \mathbf{q})$ *is stable*;

(b) $\mathbf{q} \in \text{int } \mathcal{F}(\mathbf{M})$, *i.e.*, GOLCP$(\mathbf{M}, \mathbf{q})$ *is strictly feasible*;

(c) SOL$(\mathbf{M}, \mathbf{q})$ *is nonempty and bounded.*

Proof. It was observed earlier that $\mathcal{K}(\mathbf{M})$ is closed and $\mathcal{K}(\mathbf{M}) \subseteq \mathcal{F}(\mathbf{M})$. The previous theorem says that int $\mathcal{F}(\mathbf{M}) \subseteq \mathcal{K}(\mathbf{M})$. Since $\mathcal{F}(\mathbf{M})$ is a closed convex set, it follows that $\mathcal{K}(\mathbf{M}) = \mathcal{F}(\mathbf{M})$. Therefore, $\mathbf{M}$ is of type $\mathbf{Q}_0$. The implication (a) $\Longrightarrow$ (b) follows from the definition of stability. The implication (b) $\Longrightarrow$ (c) follows from the previous theorem. Now suppose that $\mathrm{SOL}(\mathbf{M}, \mathbf{q})$ is nonempty and bounded. Let $\mathbf{d}$ be the vector that appears in the definition of $\mathbf{D}$. Consider the homotopy joining $(\mathbf{M}, \mathbf{q})$ and $(\mathbf{M}, \mathbf{d})$ defined by $\mathbf{q}_t := (1-t)\mathbf{d} + t\mathbf{q}$. Since $\mathbf{q} \in \mathcal{F}(\mathbf{M})$, we have $\mathbf{q}_t \in \text{int } \mathcal{F}(\mathbf{M})$ for all $t \in [0, 1)$. $\mathrm{SOL}(\mathbf{M}, \mathbf{q}_t)$ is bounded for all t; for $t < 1$ by the previous theorem and for $t = 1$ by the assumption. We proceed as in the proof of the previous theorem to conclude that GOLCP$(\mathbf{M}, \mathbf{q})$ is stable. This gives the implication (c) $\Longrightarrow$ (a). □

Remarks. It is clear that in the above theorem, $\mathbf{q}$ is in the boundary of $\mathcal{F}(\mathbf{M}, \mathbf{q})$ if and only if $\mathrm{SOL}(\mathbf{M}, \mathbf{q})$ is unbounded (which is equivalent to saying that there is a solution ray for GOLCP$(\mathbf{M}, \mathbf{q})$).

Note that the conclusions of Theorems 11 and 12 hold if we assume (instead of the $\mathbf{E}_1$-property) that for every $p \in \text{int } \mathcal{F}(\mathbf{M}), \mathrm{SOL}(\mathbf{M}, \mathbf{p})$ is bounded.

When $k = 1$, Theorem 12 covers various known results. For example, when M is in the class $\mathbf{G} \cap \mathbf{E}_1$, the implication (c) $\Longrightarrow$ (a) (which holds since type $\mathbf{G} \subseteq$ type $\mathbf{D}$) is a result of Doverspike [8]. For M in the same class, the equivalence (b) $\Longleftrightarrow$ (c), when stated in terms of boundary of $\mathcal{F}(\mathbf{M})$ and solution rays, reduces to the result of Eagambaram and Mohan [9]. See [3], [38], and [5] for related results.

Interestingly enough, Theorem 12 covers (when $k = 1$) matrices outside of the class $\mathbf{G}$. To see this, consider a matrix M for which there is a vector d that is nondegenerate with respect to M (i.e., $x + Mx + d > 0$ for all $x \in \mathrm{SOL}(M, d)$) and $\sum \text{sgn det } M_{\alpha\alpha} \neq 0$, where $M_{\alpha\alpha}$ is the submatrix of M corresponding to the nonzero indexes of a solution x of LCP(M, d) and the summation is over all solutions x of LCP(M, d). (This sum is precisely the local degree of M at d [5]. This sum is nonzero, for example, when LCP(M, d) consists of odd number of solutions.) If M is also in the class $\mathbf{E}_1$, then Theorem 12 applies.

6.5. The $\mathbf{G}^{\#}$-property. In the classical LCP theory, copositive matrices have interesting LCP properties. In this subsection, we see how degree theory can be employed to get a generalization of the celebrated result of Lemke [34].

We say that $\mathbf{M}$ has the *sharp property* if every representative of $\mathbf{M}$ has the sharp property, i.e., for any representative A of $\mathbf{M}$,

$$x \in \mathrm{SOL}\,(A, 0) \Rightarrow \left(A + A^T\right) x \geq 0. \tag{11}$$

We note that when $k = 1$, this condition is shared by copositive matrices, $\mathbf{R}_0$-matrices, and symmetric matrices (see [5], p. 695). Now consider an $\mathbf{M}$ having the sharp property (11). Let

$$\mathcal{C}\,(\mathbf{M}) := \left\{\mathbf{q} : a \in (\mathrm{SOL}(A, 0))^* \text{ for all representatives } (A, a) \text{ of } (\mathbf{M}, \mathbf{q})\right\},$$

where for any set S in R^n, $S^* = \{x : x^T s \geq 0 \text{ for all } s \in S\}$ is the dual cone of S.

We note that $\mathcal{C}(\mathbf{M})$ is convex and every $\mathbf{d} \succ 0$ is in $\mathcal{C}(\mathbf{M})$. We claim that for any $\mathbf{q} \in \text{int } \mathcal{C}(\mathbf{M})$, SOL($\mathbf{M}$, $\mathbf{q}$) is bounded. If not, we proceed as in the proof of Theorem 11 to get a representative (A, a) of $(\mathbf{M}, \mathbf{q})$, solution x and a direction v such that $(x + \lambda v)^T(Ax + \lambda Ax + a) = 0$ for all $\lambda \geq 0$. This leads to $x^T a + (A + A^T)x = 0$. In view of the sharp property, $x^T a \leq 0$. We reach a contradiction to $\mathbf{q} \in \text{int } \mathcal{C}(\mathbf{M})$.

THEOREM 13. *Let* $\mathbf{M}$ *have the sharp property* (11) *and suppose that for some* $\mathbf{d} \in \text{int } \mathcal{C}(\mathbf{M})$ *and for some bounded open set* Ω *containing* SOL($\mathbf{M}, \mathbf{d}$), $\deg(F_{(\mathbf{M},\mathbf{d})}, \Omega, 0) \neq 0$. *Then for all* $\mathbf{q} \in \text{int } \mathcal{C}(\mathbf{M})$, GOLCP($\mathbf{M}, \mathbf{q}$) *is stable. Moreover, for all* $\mathbf{q} \in \mathcal{C}(\mathbf{M})$, GOLCP($\mathbf{M}, \mathbf{q}$) *has a solution.*

Proof. We fix a $\mathbf{q} \in \text{int } \mathcal{C}(\mathbf{M})$ and consider the homotopy joining $(\mathbf{M}, \mathbf{d})$ and $(\mathbf{M}, \mathbf{q})$ defined by $\mathbf{q}_t := (1 - t)\mathbf{d} + t\mathbf{q}$. Since int $\mathcal{C}(\mathbf{M})$ is convex, $\mathbf{q}_t \in \text{int } \mathcal{C}(\mathbf{M})$ and SOL($\mathbf{M}, \mathbf{q}_t$) is bounded for each t. By Proposition 4, SOL($\mathbf{M}, \mathbf{q}_t$) is uniformly bounded for all t. We proceed as in the second part of the proof of Theorem 11 and get the stability of GOLCP($\mathbf{M}, \mathbf{q}$). Since $\mathcal{K}(\mathbf{M})$ is closed, the inclusion int $\mathcal{C}(\mathbf{M}) \subseteq \mathcal{K}(\mathbf{M})$ shows that $\mathcal{C}(\mathbf{M}) \subseteq \mathcal{K}(\mathbf{M})$. The second part of the theorem follows. □

By definition, $\mathbf{M}$ has the $\mathbf{G}^{\#}$-property if it has both the sharp property and the $\mathbf{G}$-property. The following result is immediate.

COROLLARY 1. *Suppose that* $\mathbf{M}$ *has the* $\mathbf{G}^{\#}$*-property. Then for all* $\mathbf{q} \in \text{int } \mathcal{C}(\mathbf{M})$, GOLCP($\mathbf{M}, \mathbf{q}$) *is stable. Moreover, for all* $\mathbf{q} \in \mathcal{C}(\mathbf{M})$, GOLCP($\mathbf{M}, \mathbf{q}$) *has a solution.*

Remarks. It is easily seen that $\mathbf{M}$ has the $\mathbf{G}^{\#}$-property if $\mathbf{M}$ has the copositive-plus property, and so Corollary 1 is applicable. When $k = 1$, the first part in the above corollary covers Theorem 7.5.8 and Exercise 7.6.6 in [5]. Furthermore, for $k = 1$, the second part in the above corollary was proved in [25] using the basic theorem of complementarity of Eaves. Note that this second part, for copositive matrices, is precisely the result of Lemke [34]. The above corollary can further be specialized to copositive plus matrices to obtain well-known results of Mangasarian [38] and Cottle [3]. We omit the details.

7. The extended GOLCP. In this section, we indicate how to analyze the extended GOLCP (EGOLCP). Given

$$\mathbf{B} = (B_0, B_1, B_2, \ldots, B_k) \quad \text{and} \quad \mathbf{b} = (b_0, b_1, b_2, \ldots, b_k), \tag{12}$$

where each B_j is an $n \times n$ matrix and b_j is an n-vector, the EGOLCP($\mathbf{B}, \mathbf{b}$) is to find $x \in R^n$ such that

$$F_{(\mathbf{B},\mathbf{b})}(x) := (B_0 x + b_0) \wedge (B_1 x + b_1) \wedge (B_2 x + b_2) \wedge \cdots \wedge (B_k x + b_k) = 0. \tag{13}$$

We write $F_{\mathbf{B}}(x) := F_{(\mathbf{B},\mathbf{0})}(x)$. If $F_{\mathbf{B}}(x) = 0$ implies that $x = 0$, then $\deg(F_{\mathbf{B}}, \Omega, 0)$ is defined and is independent of Ω, where Ω is any bounded open set containing the origin in R^n. We call this number the EGOLCP-degree of $\mathbf{B}$. The result below is similar to Theorem 1. We omit the proof.

THEOREM 14. *Suppose that* $F_{\mathbf{B}}(x) = 0$ *only when* $x = 0$, *and* EGOLCP-degree *of* $\mathbf{B}$ *is nonzero. Then for all* $\mathbf{b}$, *the* EGOLCP($\mathbf{B}, \mathbf{b}$) *has a solution.*

As in §6.2, we say that an $n \times n$ matrix B is a *representative* matrix of $\mathbf{B}$ if the ith row $(i = 1, 2, \ldots, k)$ of B comes from the set $\{B_0^i, B_1^i, \ldots, B_k^i\}$ where, of course, the superscript refers to the corresponding row. We note immediately that if every representative matrix of $\mathbf{B}$ is nonsingular, then the implication $F_{\mathbf{B}}(x) = 0 \Longrightarrow x = 0$ holds.

The next two results address the question of calculating the EGOLCP-degree of $\mathbf{B}$.

THEOREM 15. *Suppose that* $F_{\mathbf{B}}(x) = 0$ *only when* $x = 0$ *and that* B_0 *is nonsingular. Let* $M_i = B_i(B_0)^{-1}$ *for* $i = 1, \ldots, k$. *Then* $\mathbf{M}$ *given by* (4) *is of* type $\mathbf{R}_0$ *and*

$$\text{EGOLCP-deg } \mathbf{B} = \text{sgn det } (B_0)\, \text{GOLCP-deg } \mathbf{M}.$$

Proof. The first part of the conclusion follows immediately from the relation $F_{\mathbf{B}}(B_0^{-1}x) = F_{\mathbf{M}}(x)$. The second part follows from the equality $\deg(F_{\mathbf{M}}, \Omega, 0) = \text{sgn det } (B_0) \deg(F_{\mathbf{B}}, \Omega, 0)$, which is a consequence of the multiplication property of the degree [35, Thm. 2.3.1]. □

THEOREM 16. *Suppose that* $F_{\mathbf{B}}(x) = 0$ *only when* $x = 0$. *Then for all* $\tilde{\mathbf{B}} = (\tilde{B}_0, \tilde{B}_1, \ldots, \tilde{B}_k)$ *sufficiently close to* $\mathbf{B}$ *with* $\tilde{B}_0$ *invertible, we have*

$$\text{EGOLCP-deg } \mathbf{B} = \text{sgn det } \left(\tilde{B}_0\right) \text{GOLCP-deg } \tilde{\mathbf{M}},$$

where $\tilde{\mathbf{M}} = (\tilde{M}_1, \tilde{M}_2, \ldots, \tilde{M}_k)$ *and* $\tilde{M}_i = \tilde{B}_i(\tilde{B}_0)^{-1}$ $(i = 1, 2, \ldots, k)$.

Proof. As in the proof of Theorem 6, we can appeal to the nearness property of the degree and conclude that EGOLCP-deg $\mathbf{B}$ =EGOLCP-deg $\tilde{\mathbf{B}}$ whenever $\tilde{\mathbf{B}}$ is sufficiently close to $\mathbf{B}$. When $\tilde{B}_0$ is invertible, we can use the previous theorem to get the desired conclusion. □

As an illustration of the above result, we state the following corollary.

COROLLARY 2. *Suppose that* (i) $F_{\mathbf{B}}(x) = 0$ *only when* $x = 0$, *and* (ii) *for some* $\mathbf{b}^*$ *with* $b_0^* = 0$ *and* $b_i^* > 0, i = 1, 2, \ldots, k$, *the zero vector is the only solution of* EGOLCP$(\mathbf{B}, \mathbf{b}^*)$. *Then for every* $\mathbf{b}$, EGOLCP$(\mathbf{B}, \mathbf{b})$ *has a solution.*

Proof. To see this, first observe (by the usual normalization arguments) that the above two conditions hold if $\mathbf{B}$ is perturbed (slightly) to $\tilde{\mathbf{B}}$ for which Theorem 16 is applicable. The resulting $\tilde{\mathbf{M}}$ is of type $\mathbf{R}$, and so, by Theorem 2, EGOLCP-deg $\mathbf{B}$ is nonzero. Now applying Theorem 14 we get the desired result. □

We now turn to the uniqueness issue. The result below completely characterizes the uniqueness in EGOLCPs.

THEOREM 17. *Let* $\mathbf{B}$ *be given by* (12) *and when* B_0 *is invertible, let* $\mathbf{M}$ *be given by* (4), *where* $M_i = B_i(B_0)^{-1}, i = 1, 2, \ldots, k$. *Then the following conditions are equivalent.*

1. *For every* $\mathbf{b}$, EGOLCP$(\mathbf{B}, \mathbf{b})$ *has a unique solution.*
2. *The implication* $B_0x \wedge B_1x \wedge \cdots \wedge B_kx \leq 0 \leq B_0x \vee B_1x \vee \cdots \vee B_kx \Longrightarrow x = 0$ *holds.*
3. *For every* $\mathbf{b}$, EGOLCP$(\mathbf{B}, \mathbf{b})$ *has at most one solution.*
4. B_0 *is invertible and* $\mathbf{M}$ *has the* $\mathbf{P}$*-property.*
5. *All representative matrices of* $\mathbf{B}$ *have the same (nonzero) determinantal sign.*

Proof. 1 $\Longrightarrow$ 2: Similar to the proof of the implication (b) $\Longrightarrow$ (a) in Theorem 3.

2 $\Longrightarrow$ 3: Similar to the proof of the implication (a) $\Longrightarrow$ (b) in Theorem 3.

3 $\Longrightarrow$ 4: Suppose that $B_0x = 0$. Define $\mathbf{b}$ by $b_0 = -B_0x^+$ $(= -B_0x^-)$, $b_i = (-B_ix^+) \vee (-B_ix^-)$ for $i = 1, 2, \ldots, k$. Then x^+ and x^- are solutions of EGOLCP$(\mathbf{B}, \mathbf{b})$. From condition 3, $x = 0$, and hence B_0 is invertible. Now for any $\mathbf{q}$ given by (4), u is a solution of GOLCP$(\mathbf{M}, \mathbf{q})$ if and only if $v = B_0^{-1}u$ is a solution of EGOLCP$(\mathbf{B}, \mathbf{b})$, where $b_0 = 0$ and $b_i = q_i$ for $i = 1, 2, \ldots, k$. Hence, we see from Condition 3 that GOLCP$(\mathbf{M}, \mathbf{q})$ as at most one solution for every $\mathbf{q}$. The $\mathbf{P}$-property of $\mathbf{M}$ follows from Theorems 3 and 10.

4 $\Longrightarrow$ 1: Clearly, x is a solution of EGOLCP$(\mathbf{B}, \mathbf{b})$ if and only if $y = B_0x + b_0$ is a solution of GOLCP$(\mathbf{M}, \mathbf{q})$, where $\mathbf{q}$ is given by (4) with $q_i = b_i - M_ib_0, i = 1, 2, \ldots, k$. Now the desired implication follows from Theorem 3.

4 $\iff$ 5: It is easily seen that for any $n \times n$ matrix C, any representative matrix of

$$\mathbf{B}C := (B_0C, B_1C, \ldots, B_kC)$$

looks like BC where B is a representative matrix of $\mathbf{B}$. Hence, when C is nonsingular, all representative matrices of $\mathbf{B}C$ will have the same nonzero determinantal sign if and only if all representative matrices of $\mathbf{B}$ have the same nonzero determinantal sign. The equivalence of Conditions 4 and 5 is seen by taking $C = B_0^{-1}$ and observing that $\mathbf{M}$ has the $\mathbf{P}$-property if and only if all representative matrices of $(I, M_1, M_2, \ldots, M_k)$ have the same positive determinantal sign. □

Remarks. In some practical situations, for example, in nonlinear network theory [20], it is essential to know when a piecewise-linear function on R^n is a homeomorphism. Many researchers, including Kuhn and Löwen, Rheinboldt and Vandergraft, Kojima and Saigal, and Schramm, studied this problem (see the recent paper of Ralph [45] for precise references). We now show that condition 5 in Theorem 17 (which is typical in the above works) implies that for each $\mathbf{b}$, the piecewise-linear function $F_{(\mathbf{B},\mathbf{b})}$, given by (13), is a homeomorphism of R^n. This can be seen easily by observing that the equation $F_{(\mathbf{B},\mathbf{b})}(x) = y$ is equivalent to EGOLCP$(\mathbf{B}, \mathbf{b} - \mathbf{y})$, where $\mathbf{y} := (y, y, \ldots, y)$, and by appealing to the Invariance of Domain Theorem.

In some important practical problems involving the EGOLCP, one encounters [50], [51], [41], via discretization procedures, $\mathbf{Z}$-matrices. (Recall that these are matrices having nonpositive off-diagonal entries.) The following corollary deals with these matrices.

COROLLARY 3. *Consider a* $\mathbf{B}$ *given by* (12) *in which each* B_i *is a* $\mathbf{Z}$*-matrix. Suppose that one of the following conditions holds.*

(a) *Each* B_i *has positive diagonal and is strictly diagonally dominant.*

(b) *There exists a positive vector* e *in* R^n *such that* $B_ie > 0$ *for all* $i = 0, 1, \ldots, k$.

Then for all $\mathbf{b}$, EGOLCP$(\mathbf{B}, \mathbf{b})$ *has a unique solution.*

Proof. Let B be any representative matrix of $\mathbf{B}$. Obviously, B is a $\mathbf{Z}$-matrix. Moreover, under (a), B will have positive diagonal entries and is strictly diagonally dominant, and when (b) holds, $Be > 0$. In either situation, B is a $\mathbf{P}$-matrix [5], and hence the determinant of B is positive. The equivalence of Conditions 1 and 5 in Theorem 17 gives the desired result. □

Remarks. The conditions (a) and (b) of Corollary 3 were considered, in the context of EGOLCP, by Isac and Goeleven [31] and Goeleven [22], respectively. They showed that for each $\mathbf{b}$, EGOLCP$(\mathbf{B}, \mathbf{b})$ has at most one solution but they did not prove the existence of solutions.

Concluding remarks. In this paper, we have presented an analysis of existence, uniqueness, and stability aspects of the GOLCP and its extension, the EGOLCP. This analysis, which is based on degree theory, is by no means complete. For example, we have not treated $\mathbf{Z}$-matrices, row (column) sufficient matrices, and their counterparts in the general theory. Note, however, that there are results for such matrices and their generalizations [5], [2], [15]. Although our study is rather limited, we believe that, on the whole, the GOLCP theory is very similar to the classical LCP theory.

Acknowledgment. We thank one of the referees for pointing out an error in an earlier version of Theorem 15 and for indicating the given simplified proof in the present version.

REFERENCES

[1] J. M. BORWEIN, *Alternative theorems in general complementarity problems*, Lecture Notes in Economics and Mathematical Systems 259, Springer-Verlag, Berlin, New York, 1985, pp. 194–203.

[2] J. M. BORWEIN AND M. A. H. DEMPSTER, *The linear order complementarity problem*, Math. Oper. Res., 14 (1989), pp. 534–558.

[3] R. W. COTTLE, *Solution rays for a class of complementarity problems*, Math. Programming Stud., No. 1 (1974), pp. 59–70.

[4] R. W. COTTLE AND G. B. DANTZIG, *A generalization of the linear complementarity problem*, J. Combinatorial Theory, 8 (1970), pp. 79–90.

[5] R. W. COTTLE, J.-S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[6] R. W. COTTLE, J.-S. PANG, AND V. VENKATESWARAN, *Sufficient matrices and the linear complementarity problem*, Linear Algebra Appl., 114/115 (1989), pp. 231–249.

[7] C. W. CRYER AND M. A. H. DEMPSTER, *Equivalence of linear complementarity problems and linear programs in vector lattice Hilbert spaces*, SIAM J. Control Optim., 18 (1980), pp. 76–90.

[8] R. D. DOVERSPIKE, *Some perturbation results for the linear complementarity problem*, Math. Programming, 23 (1982), pp. 181–192.

[9] N. EAGAMBARAM AND S. R. MOHAN, *On strongly degenerate complementary cones and solution rays*, Math. Programming, 44 (1989), pp. 77–83.

[10] B. C. EAVES, *The linear complementarity problem*, Management Sci., 17 (1971), pp. 612–634.

[11] ———, *Solving piecewise linear convex equations*, Math. Programming Stud., No. 1 (1974), pp. 96–119.

[12] B. C. EAVES, F. J. GOULD, H. O. PEITGEN, AND M. J. TODD, EDS., *Homotopy Methods and Global Convergence*, Plenum Press, New York, 1983.

[13] A. A. EBIEFUNG, *The Generalized Linear Complementarity Problem and Its Applications*, Ph.D. thesis, Clemson University, Clemson, SC, 1991.

[14] A. A. EBIEFUNG AND M. M. KOSTREVA, *Global solvability of generalized linear complementarity problems and a related class of polynomial complementarity problems*, in Recent Advances in Global Optimization, C. Floudas and P. Pardalos, eds., Princeton University Press, Princeton, NJ, 1991.

[15] ———, **Z**-*Matrices and the Generalized Linear Complementarity Problem*, Tech. report #608, Department of Mathematical Sciences, Clemson University, Clemson, SC, 1991.

[16] ———, *The Generalized Leontief Input-Output Model and Its Application to the Choice of New Technology*, Ann. Oper. Res., 44 (1993), pp. 161–172.

[17] ———, *Generalized* $\mathbf{P}_0$- *and* **Z**-*Matrices*, Linear Algebra Appl., 195 (1993), pp. 165–179.

[18] T. FUJIMOTO, *An extension of Tarski's fixed point theorem and its application to isotone complementarity problems*, Math. Programming, 28 (1984), pp. 116–118.

[19] ———, *Nonlinear complementarity problems in a function space*, SIAM J. Control Optim., 18 (1980), pp. 621–623.

[20] T. FUJISAWA AND E. S. KUH, *Piecewise-linear theory of nonlinear networks*, SIAM J. Appl. Math., 22 (1972), pp. 307–328.

[21] T. FUJISAWA, E. S. KUH, AND T. OHTSUKI, *A sparse matrix method for analysis of piecewise-linear resistive networks*, IEEE Trans. Circuit Theory, 19 (1972), pp. 571–584.

[22] D. GOELEVEN, *A Uniqueness Theorem for the Generalized Linear Complementarity Problem*, Tech. report, Départment de Mathématiques, Facultés universitaires N.-D. de la Paix, Namur, Belgique, 1992.

[23] M. S. GOWDA, *Applications of degree theory to linear complementarity problems*, Math. Oper. Res., 18 (1993), pp. 868–879.

[24] M. S. GOWDA AND J.-S. PANG, *On solution stability of the linear complementarity problem*, Math. Oper. Res., 17 (1991), pp. 77–83.

[25] ———, *The basic theorem of complementarity revisited*, Math. Programming, 58 (1993), pp. 161–177.

[26] ———, *Stability Analysis of Variational Inequalities and Nonlinear Complementarity Problems, via the Mixed Linear Complementarity Problem and Degree Theory*, Math. Oper. Res., to appear.

[27] M. S. GOWDA AND J.-S. PANG, *On the boundedness and stability of solutions to the affine variational inequality problem*, SIAM J. Control Optim., 32 (1994), pp. 421–441.
[28] M. S. GOWDA AND T. I. SEIDMAN, *Generalized linear complementarity problems*, Math. Programming, 46 (1990), pp. 329–340.
[29] C. D. HA, *Stability of the linear complementarity problem at a solution point*, Math. Programming, 31 (1985), pp. 327–338.
[30] ———, *Application of degree theory in stability of the complementarity problem*, Math. Oper. Res., 12 (1987), pp. 368–376.
[31] G. ISAC AND D. GOELEVEN, *The implicit general order complementarity problem, models and iterative methods*, Ann. Oper. Res., 44 (1993), pp. 63–92.
[32] G. ISAC AND M. M. KOSTREVA, *The generalized order complementarity problem*, J. Optim. Theory Appl., 71 (1991), pp. 517–534.
[33] S. KARAMARDIAN, *An existence theorem for the complementarity problem*, J. Optim. Theory Appl., 19 (1976), pp. 227–232.
[34] C. E. LEMKE, *Bimatrix equilibrium points and mathematical programming*, Management Sci., 11 (1965), pp. 681–689.
[35] N. G. LLOYD, *Degree Theory*, Cambridge University Press, Cambridge, 1978.
[36] W. A. J. LUXEMBURG AND A. C. ZAANEN, *Riesz Spaces*, North-Holland, Amsterdam, 1971.
[37] O. L. MANGASARIAN, *Generalized linear complementarity problems as linear programs*, W. Oettli and F. Steffeus, eds., Methods Oper. Res., 31 (1979), pp. 393–402.
[38] ———, *Characterization of bounded solutions of linear complementarity problems*, Math. Programming Stud., 19 (1982), pp. 153–166.
[39] J. J. MORÉ, *Coercivity conditions in nonlinear complementarity problems*, SIAM Rev., 16 (1974), pp. 1–15.
[40] K. G. MURTY, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann-Verlag, West Berlin, 1987.
[41] K. P. OH, *The formulation of the mixed lubrication problem as a generalized nonlinear complementarity problem*, J. Tribology, 108 (1986), pp. 598–604.
[42] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
[43] J.-S. PANG, *The implicit complementarity problem*, in Nonlinear Programming, Vol. 4, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1981, pp. 487–518.
[44] A. L. PERESSINI, *Ordered Topological Vector Spaces*, Harper & Row, New York, 1967.
[45] D. RALPH, *A new proof of Robinson's homeomorphism theorem for* PL*-normal maps*, Linear Algebra Appl., 178 (1992), pp. 249–260.
[46] R. C. RIDDEL, *Equivalence of nonlinear complementarity problems and least element problems in Banach lattices*, Math. Oper. Res., 6 (1981), pp. 462–474.
[47] S. M. ROBINSON, *Generalized equations and their solutions, Part* I*: Basic theory*, Math. Programming Stud., 10 (1979), pp. 128–141.
[48] ———, *Some continuity properties of polyhedral multifunctions*, Math. Programming Study, 14 (1981), pp. 206–214.
[49] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[50] M. SUN, *Monotonicity of Mangasarian's iterative algorithm for generalized linear complementarity problems*, J. Math. Anal. Appl., 144 (1989), pp. 474–485.
[51] ———, *Singular control problems in bounded intervals*, Stochastics, 21 (1987), pp. 303–344.
[52] ———, *Singular stochastic control problems solved by a sparse simplex method*, IMA J. Math. Control Inform., 6 (1989), pp. 27–38.
[53] B. P. SZANC, *The Generalized Complementarity Problem*, Ph.D. thesis, Rensselear Polytechnic Institute, Troy, NY, 1989.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 796–803, July 1994

006

A MATRIX APPROACH TO FINDING A SET OF GENERATORS AND FINDING THE POLAR (DUAL) OF A CLASS OF POLYHEDRAL CONES *

CAROLYN PILLERS DOBLER†

Abstract. A set of generators for a polyhedral cone is found by using a new representation for the polar (dual) cone. Examples from order-restricted statistical inference are chosen to illustrate this method.

Key words. polyhedral cone, generators, polar cone, dual cone, order-restricted statistical inference

AMS subject classifications. primary 52B99; secondary 15A39

1. Introduction. Mathematical quantities or objects can often be expressed by more than one representation. The object of interest in this paper is a polyhedral cone or a "cone which is a polyhedron" [1, p. 55]. A polyhedral cone is a special type of closed, convex cone and has at least two representations. It can be expressed as the solution set of a system of homogeneous linear inequalities:

$$C = \{\mathbf{x} \in \Re^k : \mathbf{a}_i'\mathbf{x} \leq 0, \quad i = 1, 2, \ldots, m\} = \{\mathbf{x} \in \Re^k : A'\mathbf{x} \leq \mathbf{0}\}, \tag{1}$$

where $\mathbf{a}_i'$ is the ith row of A'. Note that for any given system of homogeneous linear inequalities, the matrix A is not unique. Another representation of a polyhedral cone is the set of all nonnegative linear combinations of a finite set of generating vectors [1, Thm. of Minkowski]:

$$\begin{aligned} C &= \left\{ \mathbf{x} \in \Re^k : \mathbf{x} = \sum_{i=1}^{p} \lambda_i \mathbf{g}_i, \quad \forall \lambda_i \geq 0, \quad i = 1, 2, \ldots, p \right\} \\ &= \{\mathbf{x} \in \Re^k : \mathbf{x} = G\boldsymbol{\lambda} \quad \forall \boldsymbol{\lambda} \geq \mathbf{0}\}, \end{aligned} \tag{2}$$

where $\mathbf{g}_i$ is the ith column of G. The columns of G are a set of generators for C. Note that we say "a" set of generators rather that "the" set of generators, because a set of generators is not unique. Also note any set of generators in scale invariant, that is, if $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_p\}$ is a set of generators for C and α is any positive scalar, then $\{\alpha\mathbf{g}_1, \alpha\mathbf{g}_2, \ldots, \alpha\mathbf{g}_p\}$ is also.

Polyhedral cones arise in many practical problems, particularly in order-restricted statistical inference (wherein constraints on a set of parameters are often defined by polyhedral cones) and in linear programming. In some of these problems, a homogeneous system of linear inequalities, as in (1), is specified, but one needs to find a set of generators, as in (2). In this paper, we discuss a matrix approach for finding a set of generators for a particular class of polyhedral cones. That is, we provide an answer to

*Received by the editors May 28, 1991; accepted for publication (in revised form) February 3, 1993.

†Department of Mathematics and Computer Science, Gustavus Adolphus College, St. Peter, Minnesota 56082 (dobler@gac.edu).

the question: If a matrix A in (1) is specified, how can we find a matrix G to satisfy (2)? In fact, the answer to this question also provides the answer to another question: How can we find the polar (dual) of a polyhedral cone?

2. Motivation and examples. Why is the identification of a set of generators of a polyhedral cone important? A set of generators is useful for determining order-preserving functions [2]. In order-restricted statistical inference, a set of generators can be used to verify one of the conditions that the isotonic regression must satisfy [3, Example 1.3.2]. An important use of a set of generators is in the creation of contrast statistics for hypothesis testing problems involving order restrictions [3, §§4.2 and 4.3], [4], and [5]. In minimization and maximization problems, one often needs to find the closest point in a polyhedral cone to a specified point $\mathbf{x} \in \Re^k$; this closest point is called the projection of $\mathbf{x}$ onto C. There are many algorithms available for computing the projection. In [6], an algorithm is presented where C is expressed in representation (2), and thus one can compute the projection from a set of generators. For certain polyhedral cones, this algorithm may be more efficient than existing algorithms because the algorithm converges in finitely many linear steps.

An example of a polyhedral cone using both representations (1) and (2) is provided by the "simple tree" polyhedral cone [3, Example 1.3.2]. The simple tree polyhedral cone is the set of all vectors in $\Re^k$ satisfying the simple tree partial order, that is, $x_1 \leq x_i, i = 2, 3, \ldots, k$. Expressing the simple tree cone in representation (1), the matrix A is a $k \times (k-1)$ matrix with (i,j)th entry: $a_{ij} = 1$ if $i = 1, a_{ij} = -1$ if $i = j+1$, and $a_{ij} = 0$ otherwise. The matrix G in representation (2) is a $k \times (k+1)$ matrix with ith column $\mathbf{g}_i$: for $i = 1, 2, \ldots, k-1, g_{i,i+1} = 1$ and $g_{ij} = 0$ for $j \neq i+1$; $\mathbf{g}_k = \mathbf{1}'$ (the vector of ones), and $\mathbf{g}_{k+1} = -\mathbf{g}_k$. We see that for any vector $\mathbf{x}$ that follows the simple tree order, $\mathbf{x}$ can be written as $\mathbf{x} = G\boldsymbol{\lambda} = \sum_{i=1}^{k+1} \lambda_i \mathbf{g}_i$ for some $\lambda_i \geq 0, i = 1, 2, \ldots, k+1$. In particular, $\lambda_i = x_i - x_1 \geq 0$ for $i = 1, 2, \ldots, k-1$, and if $x_1 < 0, \lambda_k = 0$ and $\lambda_{k+1} = -x_1 > 0$, whereas if $x_1 \geq 0, \lambda_k = x_1 \geq 0$ and $\lambda_{k+1} = 0$.

Several examples of polyhedral cones appearing in the literature are the simple linear order [3, §1.2] and [4], the star-shaped order [3, §5.3], the unimodal (umbrella) order [3, §5.5], the positive orthant [7, Chap. 14, §C], and the positive orthant restricted to the linear subspace where the sum of the components of the vector is zero [7, Chap. 14, §C].

3. Background. The standard reference for polyhedral cones is [1, Chap. 2]. A discussion of polyhedral cones in the context of order-restricted statistical inference appears in [3, §2.7].

Define a weighted inner product as $\langle \mathbf{x}, \mathbf{y} \rangle_W = \mathbf{x}' W \mathbf{y}$, for $\mathbf{x}, \mathbf{y} \in \Re^k$, where W is a $k \times k$ positive definite weight matrix. (Note that since W is positive definite, it must also be symmetric.) In a statistical context, the weight matrix W is often the variance–covariance matrix, or it may be a diagonal matrix, with diagonal entries being the sample sizes or reciprocals of the sample sizes. The notation $\mathbf{x} \leq \mathbf{y}$ denotes component-wise inequality, that is, $x_i \leq y_i, i = 1, 2, \ldots, k$.

The question of interest is: Given a polyhedral cone in its inequality representation, (1), with a matrix A specified, how can we find a matrix G in (2) so we can identify a set of generators of the cone? When A is $k \times k$ and full rank, a set of generators is given by the columns of $-A^{-1}$ [2], but a more general solution does not exist in the literature. Many of the examples in the literature simply provide a set of generators and then show that any vector in the polyhedral cone can be written as a nonnegative linear combination of these generators, but it is not indicated *how* these generators are obtained. This process appears to be largely trial and error. We now

discuss a more systematic way of finding a set of generators for a class of polyhedral cones.

The key to the relationship between A in (1) and G in (2) is provided by the polar, or dual, cone. The polar cone of a polyhedral cone C is

$$(3) \qquad C^P = \{\mathbf{y} \in \Re^k : \langle \mathbf{x},\ \mathbf{y}\rangle_W \leq 0 \quad \forall \mathbf{x} \in C\} = \{y \in \Re^k : \mathbf{x}'W\mathbf{y} \leq 0 \quad \forall \mathbf{x} \in C\}.$$

Geometrically, the polar (dual) cone is the set of all vectors in $\Re^k$ that make an obtuse angle with every vector in C. The polar, C^P, is also a polyhedral cone, and $C^{PP} = C$ [1].

For any C expressed in representation (1), an alternative representation for the polar is

$$(4) \qquad C^P = \{\mathbf{y} \in \Re^k : \mathbf{y} = W^{-1}A\boldsymbol{\lambda}, \quad \forall \boldsymbol{\lambda} \geq \mathbf{0}\}$$

[1, p. 55]. From this representation, it is obvious that the columns of $W^{-1}A$ are a set of generators for C^P.

Since C^P is a polyhedral cone, it can be written in an inequality representation, that is, $C^P = \{\mathbf{y} \in \Re^k : G'\mathbf{y} \leq \mathbf{0}\}$. By (1), (4), and the fact that $C^{PP} = C$, we see that $C = \{\mathbf{x} \in \Re^k : \mathbf{x} = W^{-1}G\boldsymbol{\alpha}$, for all $\boldsymbol{\alpha} \geq \mathbf{0}\}$, and a set of generators for C are the columns of $W^{-1}G$. To reach our goal of finding a set of generators for C, we express C^P in its inequality representation, (1), where the matrix G defining the inequalities of C^P is somehow related to A, a specified matrix defining the inequalities of C.

To facilitate reaching our goal, let us rewrite C and C^P in a slightly different form. Note that any polyhedral cone is determined by either linear inequalities, linear equalities, or both. Separating the inequalities and equalities, an alternative form of (1) is

$$(5) \qquad C = \{\mathbf{x} \in \Re^k : A'\mathbf{x} \leq \mathbf{0} \quad \text{and} \quad B'\mathbf{x} = \mathbf{0}\},$$

where A is $k \times m_1$ and B is $k \times m_2$.

An alternative form of C^P corresponding to (4) is

$$(6) \qquad C^P = \{\mathbf{y} \in \Re^k : \mathbf{y} = W^{-1}(A\boldsymbol{\lambda} + B\mathbf{r}), \quad \forall \boldsymbol{\lambda} \geq \mathbf{0}, \quad \mathbf{r} \in \Re^{m_2}\}.$$

DEFINITION 3.1. *A polyhedral cone in the form of* (5) *is in* full rank *form if* rank $(A) = m_1$ *and* rank $(B) = m_2$.

Note that for a polyhedral cone to be in full rank form, the rows of $A'(B')$ must be linearly independent and the number of inequality (equality) constraints, m_1 (m_2), must be less than or equal to the dimension of the space, k. Not all polyhedral cones can be written in full rank form. When trying to express a polyhedral cone in full rank form, it is often helpful to first eliminate all redundant constraints. Let the columns of A (B) be expressed as a set of m_1 (m_2) vectors $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{m_1}\}$ $(\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m_2}\})$. A vector $\mathbf{a}_l$ is redundant if

$$\{\mathbf{x} \in \Re^k : \mathbf{a}_i'\mathbf{x} \leq 0,\ i \in \{1, 2, \ldots, m_1\} \text{ and } \mathbf{b}_j'\mathbf{x} = 0,\ j \in \{1, 2, \ldots, m_2\}\}$$
$$= \{\mathbf{x} \in \Re^k : \mathbf{a}_i'\mathbf{x} \leq 0,\ i \in \{1, 2, \ldots, m_1\} - \{l\} \text{ and } \mathbf{b}_j'\mathbf{x} = 0,\ j \in \{1, 2, \ldots, m_2\}\}.$$

A vector $\mathbf{b}_l$ is redundant by a similar definition. Hence, a constraint (either $\mathbf{a}_l'\mathbf{x} \leq 0$ or $\mathbf{b}_l'\mathbf{x} = 0$) is redundant if the vector $\mathbf{a}_l$ $(\mathbf{b}_l)$ is redundant. An algorithm for removing redundant constraints is found in [8].

DEFINITION 3.2. *A polyhedral cone in form* (5) *is in* nonredundant *form if all redundant constraints have been removed.*

An example of a polyhedral cone that cannot be written in full rank form is the simple loop order [3, p. 84] when $k = 4$: $C = \{\mathbf{x} \in \Re^4 : x_1 \leq x_2 \leq x_4 \text{ and } x_1 \leq x_3 \leq x_4\}$. Then

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

which contains no redundant columns. Since rank $(A) = 3 \neq 4 = m_1$, the simple loop polyhedral cone does not have a full rank form. We discuss the nonfull rank case more fully in §6.

The set of all polyhedral cones that can be written in full rank form is called the class of full rank polyhedral cones. The main result in the next section provides a relationship between C and C^P, both in representation (5), for the class of full rank polyhedral cones that satisfy one additional condition.

4. The main result. In the following, I_m is the $m \times m$ identity matrix and $0_{m_1 \times m_2}$ is an $m_1 \times m_2$ matrix of zeros. Also recall that W is a positive definite weight matrix that defines the weighted inner product. The following lemma provides the existence of a matrix needed in the theorem.

LEMMA 4.1. *Let A be a $k \times m$ matrix,* rank $(A) = m$. *Then there exists a matrix G_A such that $G_A A = I_m$.*

Proof. Note that G_A' is the solution to $A'G_A' = I_m$. Its existence can be seen by considering the m sets of simultaneous equations $A'\mathbf{g}_{A_i} = \mathbf{e}_i$, where $\mathbf{g}_{A_i}$ is the ith column of G_A' and $\mathbf{e}_i$ is the column vector with ith component one, all other components zero. Since rank (A') = rank $(A) = m, A'\mathbf{g}_{A_i} = \mathbf{e}_i$ has a solution [9, Thm. 7.2.4]. Hence, $A'G_A' = I_m$ has a solution. The solution is unique if and only if $k =$ rank (A') = rank (A) [9, Cor. 7.3.1.4]. Thus, a unique G_A exists if and only if A is a $k \times k$ full rank matrix. □

THEOREM 4.2. *Let C be a full-rank polyhedral cone in representation* (5), *with G_A and G_B any matrices satisfying $G_A A = I_{m_1}$ and $G_B B = I_{m_2}$. If, in addition, $G_A B = 0_{m_1 \times m_2}$, then*

$$(7) \qquad C^P = \{\mathbf{y} \in \Re^k : -G_A W\mathbf{y} \leq \mathbf{0} \quad \textit{and} \quad (I - AG_A - BG_B)\,W\mathbf{y} = \mathbf{0}\}.$$

Furthermore, if $G_B A = 0_{m_2 \times m_1}$ and $m_1 + m_2 = k$, then $C^P = \{\mathbf{y} \in \Re^k : -G_A W\mathbf{y} \leq \mathbf{0}\}$.

Proof. We know that G_A and G_B exist from Lemma 4.1. Recall from (3) and (6) that $C^P = \{\mathbf{y} \in \Re^k : \mathbf{x}'W\mathbf{y} \leq 0 \text{ for all } \mathbf{x} \in C\} = \{\mathbf{y} \in \Re^k : \mathbf{y} = W^{-1}[A \quad B]\begin{bmatrix}\boldsymbol{\lambda} \\ \mathbf{r}\end{bmatrix} \text{ for all } \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{r} \in \Re^{m_2}\}$. Let C^* be as in (7), and we want to show that $C^* = C^P$. First, note if $G_B A = 0_{m_2 \times m_1}$, then $AG_A + BG_B$ is idempotent. Furthermore, if $m_1 + m_2 = k$, then rank $(AG_A + BG_B) = k$ and hence $AG_A + BG_B = I_k$.

Consider $\mathbf{y} \in C^*$. If $m_1 + m_2 = k$ and $G_B A = 0_{m_2 \times m_1}$, then for all $\mathbf{x} \in C$,

$$\mathbf{x}'W\mathbf{y} = \mathbf{x}'\,(AG_A + BG_B)\,W\mathbf{y} = \mathbf{x}'AG_AW\mathbf{y} + \mathbf{x}'BG_BW\mathbf{y} \leq 0,$$

because $\mathbf{x}'A \leq \mathbf{0}', G_AW\mathbf{y} \geq \mathbf{0}$, and $\mathbf{x}'B = \mathbf{0}'$. If $m_1 + m_2 \neq k$, then $\mathbf{y} = W^{-1}(AG_A + BG_B)W\mathbf{y}$. For all $\mathbf{x} \in C$, $\mathbf{x}'W\mathbf{y} = \mathbf{x}'WW^{-1}(AG_A + BG_B)W\mathbf{y} = \mathbf{x}'AG_AW\mathbf{y} + \mathbf{x}'BG_BW\mathbf{y} \leq 0$. Hence, $\mathbf{y} \in C^P$, so $C^* \subset C^P$.

Consider $\mathbf{y} \in C^P$. Then $\mathbf{y} = W^{-1}A\boldsymbol{\lambda} + W^{-1}B\mathbf{r}$ for some $\boldsymbol{\lambda} \geq \mathbf{0}$ and $\mathbf{r} \in \Re^{m_2}$. Now, $G_AW\mathbf{y} = G_AWW^{-1}A\boldsymbol{\lambda}+G_AWW^{-1}B\mathbf{r} = \boldsymbol{\lambda} \geq \mathbf{0}$ or $-G_AW\mathbf{y} \leq \mathbf{0}$. If $m_1+m_2 = k$ and $G_BA = 0_{m_2\times m_1}$, then $AG_A + BG_B = I_k$, so that $(I - AG_A - BG_B)W\mathbf{y} = \mathbf{0}$, and hence $\mathbf{y} \in C^*$. If $m_1 + m_2 \neq k$, then we know that $\mathbf{y} = W^{-1}A\boldsymbol{\lambda} + W^{-1}B\mathbf{r}$ has a solution, implying $W\mathbf{y} = A\boldsymbol{\lambda} + B\mathbf{r}$ has a solution. The solution must satisfy

$$[A \quad B]\begin{bmatrix} G_A \\ G_B \end{bmatrix} W\mathbf{y} = W\mathbf{y}$$

[9, Thm. 7.2.3]. Therefore, $(I - AG_A - BG_B)W\mathbf{y} = \mathbf{0}$ and $\mathbf{y} \in C^*$, so $C^P \subset C^*$. Hence, $C^* = C^P$. □

To use the theorem to find a set of generators for C: (i) Solve the system of equations $G_AA = I, G_BB = I$, and $G_AB = 0$. The system of equations may not have a unique solution. Often, G_A (or G_B) is of a simple form such as $(A'A)^{-1}A'$ or $(A'DA)^{-1}A'D$, for some $k \times k$ full rank matrix D. Although G_B always exists, there may not be a matrix G_A to satisfy the system of equations defined by $G_AA = I$ and $G_AB = 0$. (Note that if $m_1 + m_2 = k$, we solve the system $G_AA = I, G_AB = 0, G_BB = I$, and $G_BA = 0$.)

(ii) Define a set of generators by the unduplicated rows of the matrices $-G_A, I - AG_A - BG_B$, and $-(I - AG_A - BG_B)$.

To see that a set of generators arises from this process, we simply use the form of C^P in (7) to obtain a set of generators for C as the columns of $-W^{-1}(G_AW)'$ and $\pm W^{-1}((I - AG_A - BG_B)W)'$. But since W is positive definite and symmetric, a set of generators is given by the columns of $-G_A^{'}$ and $\pm(I - AG_A - BG_B)'$ or the rows of their transposes.

Note that if $A = 0$, then C is a linear subspace. C^P is the orthogonal complement of C, and the rows of $(I - BG_B)$ are a set of spanning vectors for C. Since $G_B = (B'B)^{-1}B'$ satisfies the condition of the theorem, a set of spanning vectors is provided by the rows of $(I - B(B'B)^{-1}B')$.

Also note that if $B = 0$, then C consists entirely of inequality constraints, and the only condition for C to satisfy is that the inequality constraints are independent and the number of inequality constraints (m_1) is less than the dimension of the space (k).

Although our aim was to find a set of generators for a polyhedral cone, we see that Theorem 4.2 also provides the solution to finding the polar (dual) cone. This situation is of interest in its own right, because an optimization problem involving a polyhedral cone may be easier to solve in terms of the dual cone [3, §1.7], [10], and [11]. Examples of this use of Theorem 4.2 are given in the next section.

Theorem 4.2 also provides a vehicle for expressing C in its inequality representation (1) if a set of generators are known. When a set of generators for C are known, then we can express C^P in its inequality/equality representation (7). Now, just apply Theorem 4.2, with C^P playing the role of C. Since $C^{PP} = C$, the polar of C^P in representation (7) is just C! An example of this procedure is also given in the next section.

5. Special cases and examples. We now consider special cases in which the computation of G_A and G_B is relatively straightforward. In the examples in this section, unless otherwise specified, $k = 3$ for ease of computation and of presentation, although all the examples easily extend when $k > 3$. We also assume that W is an arbitrary positive definite weight matrix, unless otherwise stated.

If $m_1 = k, B = 0$, then $G_A = A^{-1}$, and the rows of $-A^{-1}$ are a set of generators for C. This is the situation discussed in [3].

If $m_1 < k, B = 0$, then $G_A = (A'A)^{-1}A'$, the Moore–Penrose inverse of A, is a solution to $G_A A = I$. The rows of $-(A'A)^{-1}A'$ and $\pm(I - A(A'A)^{-1}A')$ are a set of generators for C. For example, consider the simple tree ordering from the example in §2 when $k = 3$. Then $C = \{\mathbf{x} \in \Re^3 : x_1 \leq x_2 \text{ and } x_1 \leq x_3\}$ and

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad G_A = \left(\frac{1}{3}\right)\begin{bmatrix} -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \quad \text{and} \quad I - AG_A = \left(\frac{1}{3}\right)\mathbf{1}\mathbf{1}'.$$

Since a set of generators is scale invariant, a set of generators for C are $\mathbf{g}_1 = [-1, 2, -1]', \mathbf{g}_2 = [-1, -1, 2]', \mathbf{g}_3 = [1, 1, 1]'$, and $\mathbf{g}_4 = [-1, -1, -1]'$. Note that this set of generators is different than the one in the example in §2. Since $\mathbf{x} = \sum_{i=1}^{4} \lambda_i \mathbf{g}_i$ for all $\mathbf{x} \in C, \lambda_1 = (x_2 - x_1)/3 \geq 0, \lambda_2 = (x_3 - x_1)/3 \geq 0$, and if $x_1 < 0, \lambda_3 = (x_2 + x_3 - 2x_1)/3 \geq 0$ and $\lambda_4 = -x_1 \geq 0$, whereas if $x_1 \geq 0, \lambda_3 = (x_1 + x_2 + x_3)/3 \geq 0$ and $\lambda_4 = 0$. This illustrates that a set of generators need not be unique. In addition, Theorem 4.2 provides the polar of C. If $W = I, C^P = (\mathbf{y} \in \Re^3 : y_1 + y_3 - 2y_2 \geq 0, y_1 + y_2 - 2y_3 \geq 0, y_1 + y_2 + y_3 = 0) = (\mathbf{y} \in \Re^3 : y_2 \leq 0, y_3 \leq 0, y_1 + y_2 + y_3 = 0)$. Now, suppose we only know a set of generators of C, $(\mathbf{g}_1, \ldots, \mathbf{g}_4)$, as given above. Since $C^P = (\mathbf{y} \in \Re^3 : y_2 \leq 0, y_3 \leq 0, y_1 + y_2 + y_3 = 0)$,

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

and $B = \mathbf{1}'$. Since $m_1 = 2$ and $m_2 = 1$, so that $m_1 + m_2 = 3$, we solve the system of equations $G_A A = I, G_A B = 0, G_B B = I$, and $G_B A = 0$. The unique solution is

$$G_A = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix},$$

and $G_B = [1 \quad 0 \quad 0]$, so $C = C^{PP} = \{\mathbf{x} \in \Re^3 : -G_A \mathbf{x} \leq \mathbf{0}\} = \{\mathbf{x} \in \Re^3 : x_1 \leq x_2 \text{ and } x_1 \leq x_3\}$, which is the simple tree cone.

The next two examples illustrate polyhedral cones with both inequality and equality constraints. First, suppose $m_1 < k$ and $B = W\mathbf{1}$, where again W is an arbitrary positive definite weight matrix. In this context, B represents a vector of weights, and we are restricting our original polyhedral cone to a linear subspace where the weighted sum of the components of the vectors is zero. An example of this situation appears in [5], where a multiple contrast statistic is defined for the treatments versus control hypothesis testing situation. In this problem, any contrast vector must satisfy the simple tree order, and the weighted sum of its components must be zero. If $A'\mathbf{1} = \mathbf{0}$, then $G_A = (A'W^{-1}A)^{-1}A'W^{-1}$ and $G_B = (\mathbf{1}'W\mathbf{1})^{-1}\mathbf{1}'$. A set of generators are the rows of $-G_A$ and $\pm[I - A(A'W^{-1}A)^{-1}A'W^{-1} - W\mathbf{1}(\mathbf{1}'W\mathbf{1})^{-1}\mathbf{1}']$.

A second example involves the umbrella order constrained to the "symmetry" subspace, that is, $C = \{\mathbf{x} \in \Re^3 : x_1 \leq x_2 \leq x_3, \text{ and } x_1 = x_3\}$, as in [12]. If we let

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

and $B' = [-1 \quad 0 \quad 1]$, then C is in full rank form, but it contains redundant constraints. Furthermore, the system of equations $G_A A = I$ and $G_A B = 0$ does not have

a solution because the rows of A' and B' are linearly dependent. Since $x_1 = x_3$, the constraint defined by the second row of A' is redundant and can be ignored. Letting $A' = [1 \quad -1 \quad 0]$, and $B' = [-1 \quad 0 \quad 1]$, then a solution to $G_A A = I, G_A B = 0$, and $G_B B = 1$ is $G_A = [0 \quad -1 \quad 0]$ and $G_B = [0 \quad 0 \quad 1]$. Then $-G_A = [0 \quad 1 \quad 0]$ and

$$I - AG_A - BG_B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and a set of generators is $\mathbf{g}_1 = [0,1,0]', \mathbf{g}_2 = [1,1,1]'$, and $\mathbf{g}_3 = -\mathbf{g}_2$. For any $\mathbf{x} \in C, \mathbf{x} = \sum_{i=1}^{3} \lambda_i \mathbf{g}_i$, where $\lambda_1 = x_2 - x_1 \geq 0$, and if $x_2 \geq 0$, then $\lambda_2 = x_2 \geq 0$ and $\lambda_3 = x_2 - x_1 \geq 0$, whereas if $x_2 < 0$, then $x_1 < 0$ so that $\lambda_2 = 0$ and $\lambda_3 = -x_1 > 0$. Further, $C^P = \{\mathbf{y} \in \Re^k : -G_A \mathbf{y} \leq 0 \text{ and } (I - AG_A - BG_B)\mathbf{y} = \mathbf{0}\} = \{\mathbf{y} \in \Re^3 : y_2 \leq 0 \text{ and } y_1 + y_2 + y_3 = 0\}$. This example illustrates the importance of eliminating all redundant constraints from A and B before trying to apply the theorem.

6. Nonfull rank case. The main problem with Theorem 4.2 is its restriction to full rank cones. Although most practical problems involve full rank cones, situations may arise when the cone is not full rank, as illustrated by the simple loop order discussed in §3. The following algorithm, suggested by a referee, can be used to find a set of generators for a nonfull rank cone.

ALGORITHM 6.1. Assume that C is expressed in representation (1) and that all constraints are nonredundant. Furthermore, rank $(A) > m$.

Step 1. Find, if possible, a subspace $S \subset C$, where S has codimension r. Let $\mathbf{a}_i^* = P(\mathbf{a}_i | S^\perp), i = 1,2,\ldots,m$, that is, project $\mathbf{a}_i$ onto $S^\perp$, the orthogonal complement of S.

Step 2. Let K_j be a subset of $\{\mathbf{a}_1^*, \mathbf{a}_2^*, \ldots, \mathbf{a}_m^*\}$ of size $r-1$, where $j = 1,2,\ldots, \binom{m}{r-1}$. Find a vector $\mathbf{b}_j$ such that $\mathbf{b}_j' \mathbf{c}_i = 0$ for all $\mathbf{c}_i \in K_j$ and $\mathbf{b}_j' \mathbf{d} = 0$ for all $\mathbf{d} \in S$. The set

$$\left\{\mathbf{b}_1 \mathbf{b}_2, \ldots, \mathbf{b}_{\binom{m}{r-1}}, -\mathbf{b}_1, -\mathbf{b}_2, \ldots, -\mathbf{b}_{\binom{m}{r-1}}\right\}$$

contains a set of generators for C.

Step 3. For each vector $\mathbf{b}_j, j = 1,2,\ldots,\binom{m}{r-1}$, compute $\mathbf{b}_j' \mathbf{a}_i$ for $i = 1,2,\ldots,m$. If $\mathbf{b}_j' \mathbf{a}_i < 0$ for all $\mathbf{a}_i^* \notin K_j$, then $\mathbf{b}_j$ is a generator of C. If $\mathbf{b}_j' \mathbf{a}_i > 0$ for all $\mathbf{a}_i^* \notin K_j$, then $-\mathbf{b}_j$ is a generator of C. If $\mathbf{b}_j' \mathbf{a}_i > 0$ for some $\mathbf{a}_i^* \notin K_j$ and $\mathbf{b}_j' \mathbf{a}_{i'} < 0$ for some $\mathbf{a}_{i'}^* \notin K_j$, then $\mathbf{b}_j$ is not a generator of C.

If the constraints of C are written "in order," then one need not search all subsets K_j.

For example, consider the simple loop order when $k = 4$. Then $\mathbf{a}_1 = (1 \quad -1 \quad 0 \quad 0), \mathbf{a}_2 = (1 \quad 0 \quad -1 \quad 0), \mathbf{a}_3 = (0 \quad 1 \quad 0 \quad -1)$, and $\mathbf{a}_4 = (0 \quad 0 \quad 1 \quad -1)$. Now $S = (\mathbf{x} \in \Re^4 : \sum_{i=1}^{4} x_i = 0)$, which is generated by $(1 \quad 1 \quad 1 \quad 1)$ and codim $(S) = 3$. Furthermore, $S^\perp = \{\mathbf{y} \in \Re^4 : y_1 + y_2 + y_3 + y_4 = 0\}$, so $\mathbf{a}_i \in S^\perp$ and hence $\mathbf{a}_i^* = \mathbf{a}_i$. Rather than consider all subsets of $\{\mathbf{a}_i\}$ of size $r - 1 = 2$, note that dim $(C^P) = 3$ and the faces of C^P are two dimensional, generated by pairs of $\{\mathbf{a}_i\}$. The pairs generating these faces are $\{\mathbf{a}_1, \mathbf{a}_2\}, \{\mathbf{a}_2, \mathbf{a}_3\}, \{\mathbf{a}_3, \mathbf{a}_4\}$, and $\{\mathbf{a}_4, \mathbf{a}_1\}$, so we only need to consider these four pairs in Step 2 of Algorithm 6.1 rather than all six pairs. It is easy to see that $\mathbf{b}_1 = (-1,-1,-1,3), \mathbf{b}_2 = (-1,1,-1,1), \mathbf{b}_3 = (-3,1,1,1)$, and $\mathbf{b}_4 = (-1,-1,1,1)$ are a set of generators of the simple loop order.

7. Summary. Theorem 4.2 provides an alternative representation for the polar (dual) of a polyhedral cone. This representation serves three purposes: (i) finding a set

of generators for a polyhedral cone, (ii) finding the polar (dual) of a polyhedral cone, and (iii) expressing a polyhedral cone in its inequality form when a set of generators is known. To use the theorem, the polyhedral cone must be in full-rank form. As seen by the "simple loop" example, not all polyhedral cones in applications can be written in full rank form. In addition, it is important to remove all redundant constraints before attempting to use this method. The examples in §5 illustrate how Theorem 4.2 can be applied to many polyhedral cones arising in practical situations, particularly in order-restricted statistical inference.

REFERENCES

[1] J. STOER AND C. WITZGALL, *Convexity and Optimization in Finite Dimensions* I, Springer-Verlag, Berlin, 1970.

[2] A. W. MARSHALL, B. W. WALKUP, AND R. J. B. WETS, *Order-preserving functions: Applications to majorization and order statistics*, Pacific J. Math., 23 (1967), pp. 569–584.

[3] T. ROBERTSON, F. T. WRIGHT, AND R. DYKSTRA, *Order Restricted Statistical Inference*, John Wiley and Sons, Ltd., Chichester, 1988.

[4] R. P. ABELSON AND J. W. TUKEY, *Efficient utilization of non-numerical information in quantitative analysis: General theory and the case of simple order*, Ann. Math. Stat., 34 (1963), pp. 1347–1369.

[5] H. MUKERJEE, T. ROBERTSON, AND F. T. WRIGHT, *Comparisons of several treatments with a control using multiple contrasts*, J. Amer. Statist. Assoc., 82 (1987), pp. 902–910.

[6] D. R. WILHEMSON, *A nearest point algorithm for convex polyhedral cones and applications to positive linear approximation*, Math. Comp., 30 (1976), pp. 48–57.

[7] A. W. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization and its Applications*, Academic Press, Orlando, FL, 1979.

[8] R. J. B. WETS AND C. WITZGALL, *Algorithms for frames and lineality spaces of cones*, J. Res. Nat. Bureau of Standards, 71 (1967), pp. 1–7.

[9] F. A. GRAYBILL, *Matrices with Applications in Statistics*, Wadsworth, Belmont, CA, 1983.

[10] T. ROBERTSON AND F. T. WRIGHT, *Likelihood ratio tests for and against a stochastic ordering between multinomial populations*, Ann. Statist., 9 (1981), pp. 1248–1257.

[11] R. L. DYKSTRA, *Dual convex cones of order restrictions with applications*, in Inequalities in Statistics and Probability, Instit. Math. Statist., Hayward, CA, 1984, pp. 228–235.

[12] T. ROBERTSON, *On testing symmetry and unimodality*, in Advances in Order Restricted Statistical Inference, Springer-Verlag, Berlin, 1986, pp. 231–248.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 804–823, July 1994

007

A UNIFORM APPROACH FOR THE FAST COMPUTATION OF MATRIX-TYPE PADÉ APPROXIMANTS *

BERNHARD BECKERMANN† AND GEORGE LABAHN‡

Abstract. Recently, a uniform approach was given by B. Beckermann and G. Labahn [*Numer. Algorithms*, 3 (1992), pp. 45–54] for different concepts of matrix-type Padé approximants, such as descriptions of vector and matrix Padé approximants along with generalizations of simultaneous and Hermite Padé approximants. The considerations in this paper are based on this generalized form of the classical scalar Hermite Padé approximation problem, *power Hermite Padé approximation.* In particular, this paper studies the problem of computing these new approximants.

A recurrence relation is presented for the computation of a basis for the corresponding linear solution space of these approximants. This recurrence also provides bases for particular subproblems. This generalizes previous work by Van Barel and Bultheel and, in a more general form, by Beckermann. The computation of the bases has complexity $\mathcal{O}(\sigma^2)$, where σ is the order of the desired approximant and requires no conditions on the input data. A second algorithm using the same recurrence relation along with divide-and-conquer methods is also presented. When the coefficient field allows for fast polynomial multiplication, this second algorithm computes a basis in the superfast complexity $\mathcal{O}(\sigma \log^2 \sigma)$. In both cases the algorithms are reliable in exact arithmetic. That is, they never break down, and the complexity depends neither on any normality assumptions nor on the singular structure of the corresponding solution table. As a further application, these methods result in fast (and superfast) reliable algorithms for the inversion of striped Hankel, layered Hankel, and (rectangular) block-Hankel matrices.

Key words. vector Padé approximant, Hermite Padé approximant, simultaneous Padé approximant, matrix Padé approximant, Hankel matrices

AMS subject classifications. 65D05, 41A21, CR: G.1.2

1. Introduction.

Let $\mathbf{F} = (f_1, \ldots, f_m)^T$ (with $m \geq 2$) be an m-tuple of formal power series with coefficients from a field $\mathbb{K}$ (typically a subfield of either the real or complex numbers) and $\mathbf{n} = (n_1, \ldots, n_m)$ an m-tuple of integers, $n_l \geq -1$. A *Hermite Padé approximant* for $\mathbf{F}$ of type $\mathbf{n}$ is a nontrivial tuple $\mathbf{P} = (P_1, \ldots, P_m)$ of polynomials P_l over $\mathbb{K}$ having degrees bounded by the n_l such that

$$\mathbf{P}(z) \cdot \mathbf{F}(z) = P_1(z)f_1(z) + \cdots + P_m(z)f_m(z) = c_N z^N + c_{N+1} z^{N+1} + \cdots, \tag{1}$$

with $N = n_1 + \cdots + n_m + m - 1$.

The *Hermite Padé approximation problem* was introduced in 1873 by Hermite and has been studied widely by several authors (for a bibliography, see, e.g. [2]–[4] or [25]). Note that when $m = 2, \mathbf{F} = (f, -1)^T$, Eq. (1) is the same as

$$P_1(z)f(z) - P_2(z) = O\left(z^{n_1+n_2+1}\right),$$

and hence as a special case we have the classical Padé approximation problem for a power series f. Hermite Padé approximation also includes other classical approximation problems such as algebraic approximants ($\mathbf{F} = (1, f, f^2, \ldots, f^{m-1})^T$) (e.g. [23] for the special case $m = 2$) and G^3J approximants ($m = 3, \mathbf{F} = (f', f, 1)^T$). We refer the reader to [1, pp. 32–40] for additional examples. More generally, there is the

* Received by the editors April 16, 1992; accepted for publication (in revised form) February 25, 1993.

† Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA-M3, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq Cedex, France.

‡ Department of Computing Science, University of Waterloo, Waterloo, Ontario, Canada (glabahn@daisy.waterloo.edu).

M-Padé approximation problem that requires that $\mathbf{P} \cdot \mathbf{F}$ vanishes at a given set of knots $z_0, z_1, \ldots, z_{N-1}$, counting multiplicities ([2]–[4], [20], [21]). The case where all the z_i are equal to 0 is just the Hermite Padé problem.

Hermite also defined a second type of approximant to a vector of power series, the so-called *simultaneous Padé approximants* and used them in his proof of the transcendence of e. Close connections between these two approximation problems have been pointed out in [7], [14], [16], [17], [21].

In recent years, several vector and matrix generalizations of these approximation problems have been given (see §2). The aim of this paper is to study a uniform approach not only to Hermite Padé and simultaneous Padé approximants but also to their matrix-type generalizations. To this end, we consider the following generalized scalar Hermite Padé approximation problem [5].

DEFINITION 1.1. *Let* $\sigma \geq 0, s > 0, n_1, \ldots, n_m$ *be integers,* $n_l \geq -1$ *and* $\mathbf{n} = (n_1, \ldots, n_m)$. *Then a* power Hermite Padé approximant *(PHPA)* $\mathbf{P} = (P_1, \ldots, P_m)$ *of type* $(\mathbf{n}, \sigma, s)$ *consists of scalar polynomials* P_l *having degrees bounded by the* n_l *with*

$$(2)\quad R(z) = \mathbf{P}(z^s) \cdot \mathbf{F}(z) = P_1(z^s) f_1(z) + \cdots + P_m(z^s) f_m(z) = c_\sigma z^\sigma + c_{\sigma+1} z^{\sigma+1} + \cdots,$$

that is, has order σ. *The power series* R *is referred to as the* s-residual.

The power s appearing in Definition 1.1 provides a method of converting a vector problem into a scalar problem (see §2). By defining these approximants in a similar way to Hermite Padé approximants we can borrow from the (successful) computational techniques for the Hermite Padé problem used in [2], [4], [25]. Of course the classical Hermite Padé approximation problem is included by setting $s = 1$ and $\sigma = \|\mathbf{n}\| - 1$, where the norm of multiindices $\mathbf{n} = (n_1, \ldots, n_m) \in (\mathbb{N}_0 \cup \{-1\})^m$ is defined by $\|\mathbf{n}\| := (n_1 + 1) + \cdots + (n_m + 1)$. Note that, by equating coefficients, (2) results in a system of homogeneous linear equations. By comparing the number of unknowns to equations, one can conclude that there exists at least $\|\mathbf{n}\| - \sigma$ PHPAs of type $(\mathbf{n}, \sigma, s)$ that are linearly independent over $\mathbb{K}$.

Section 2 gives examples of matrix-type generalizations of existing approximation problems. These are shown to be special cases of the PHPA problem for various values of s and σ. In §3 we provide a recursive algorithm to efficiently and reliably solve the PHPA problem in exact arithmetic. Some interesting properties of our algorithm along with a cost analysis are given in §4. It is shown that the algorithm is at least as fast or faster than existing methods for special cases. Thus, our results provide a uniform method of computing matrix-type generalizations of Padé approximation problems. Section 5 gives an example of the use of this algorithm in the context of square-matrix Padé approximants. Section 6 considers a modification of our algorithm that combines divide-and-conquer techniques along with the recurrence relation of §3. When the field $\mathbb{K}$ allows fast polynomial multiplication, the resulting new algorithm solves the PHPA problem with superfast complexity. Finally, the paper closes with a discussion of a number of research directions that follow from our work.

For purposes of presentation, we adopt the following notations. Let $\mathcal{S}$ be a space with scalars from $\mathbb{K}$, for instance, $\mathcal{S} = \mathbb{K}^{(p \times q)}$, the space of $p \times q$ matrices over $\mathbb{K}$. Then $\mathcal{S}[z]$ denotes the set of polynomials in z with coefficients from $\mathcal{S}$, whereas $\mathcal{S}[[z]]$ represents the set of formal power series in z with coefficients from $\mathcal{S}$. Multiindices and PHPAs are denoted in boldface letters; they are both $(1 \times m)$ row vectors. Also, throughout this paper the parameter s and the multiindex $\mathbf{n}$ are fixed. The algorithm of §3 follows along an m-dimensional "diagonal" path $(\mathbf{n}(\delta))_{\delta \in \mathbb{Z}}$ induced by $\mathbf{n}$, which

is defined as follows:

$$(3)\quad \delta \in \mathbb{Z}, \mathbf{n} = (n_1, \ldots, n_m) : \mathbf{n}(\delta) = \left(n_1', \ldots, n_m'\right) \quad \text{with } n_l' = \max\{-1, n_l + \delta\}.$$

This notion allows us to discuss not only one approximation problem corresponding to $\mathbf{n} = \mathbf{n}(0)$ but also simultaneously all subproblems associated with $\mathbf{n}(\delta), \delta < 0$ (cf. Table 3). Finally, the set of all PHPAs of type $(\mathbf{n}(\delta), \sigma, s)$ is denoted as $\mathcal{L}_\delta^\sigma$; it is a finite-dimensional space over $\mathbb{K}$.

Parallel to and independent of [5] and our present work, another uniforming approach was proposed in [26] by Van Barel and Bultheel based on the concept of vector M-Padé approximation. Their approach does not reduce to a simple scalar concept as does the notion of our PHPAs. However, their approach does have the advantage of handling matrix rational interpolation and is seen as complementary to this paper.

2. Matrix-type Padé approximants as special PHPAs. In this section we give examples of a number of matrix-type generalizations of classical Padé approximation problems. Let A be a $p \times q$ matrix of power series over $\mathbb{K}$ and suppose that $r \in \mathbb{N}$ and $M, N \in \mathbb{N}_0$.

Example 2.1 (Right-hand square and rectangular Matrix-Padé forms). Find $P \in \mathbb{K}^{(p\times r)}[z], Q \in \mathbb{K}^{(q\times r)}[z]$, with $\deg P \le M$, $\deg Q \le N$, and the columns of Q being linearly independent over $\mathbb{K}$ such that

$$A(z) \cdot Q(z) - P(z) = z^{M+N+1} \cdot R(z),$$

with $R \in \mathbb{K}^{(p\times r)}[[z]]$.

Example 2.2 (Left-hand square and rectangular Matrix-Padé forms). Find $P \in \mathbb{K}^{(r\times q)}[z], Q \in \mathbb{K}^{(r\times p)}[z]$, with $\deg P \le M$, $\deg Q \le N$, and the rows of Q being linearly independent over $\mathbb{K}$ such that

$$Q(z) \cdot A(z) - P(z) = z^{M+N+1} \cdot R(z),$$

with $R \in \mathbb{K}^{(r\times q)}[[z]]$.

When $p = q = r = 1$ this is the classical scalar Padé approximation problem. When $p = q = r > 1$ these are square right-hand or left-hand matrix Padé approximants [19]. In the rectangular ($p \neq q$) case, two natural matrix Padé approximations occur when either $p = r$ or $q = r$. Both of these rectangular-matrix types of Padé forms are used, for example, to compute the inverse of matrices partitioned into a rectangular-block Hankel structure [18].

We remark that, in the examples where $Q(z)$ is square, it is of special interest to determine those cases where we can form a Padé fraction $P(z)\cdot Q(z)^{-1}$ or $Q(z)^{-1}\cdot P(z)$ as an approximant to $A(z)$. In both cases we are therefore interested in necessary and sufficient conditions under which $Q(z)$ is nonsingular.

Motivated by the well-known connections between left-hand and right-hand square matrix Padé forms and by inversion formulas of block Hankel-like matrices, one of the authors [17] introduced for $p, \mu \in \mathbb{N}$ and $\rho_0, \ldots, \rho_\mu \ge 0, \rho = \rho_0 + \cdots + \rho_\mu, A_0, \ldots, A_\mu \in \mathbb{K}^{p\times p}[[z]]$.

Example 2.3 (Matrix Hermite Padé form). Find polynomials $P_0, \ldots, P_\mu \in \mathbb{K}^{p\times p}[z]$ with $\deg P_l \le \rho_l - 1, 0 \le l \le \mu$, and

$$A_0(z)P_0(z) + \cdots + A_\mu(z)P_\mu(z) = z^{\rho-1} \cdot R(z),$$

$R \in \mathbb{K}^{p\times p}[[z]]$ such that the matrix $[P_0, \ldots, P_\mu] \in \mathbb{K}^{p\times(\mu+1)p}[z]$ has full rank over $\mathbb{K}$.

Example 2.4 (Matrix simultaneous Padé form). Find polynomials $Q_0, \ldots, Q_\mu \in \mathbb{K}^{p\times p}[z]$ with $\deg Q_l \le \rho - \rho_l, 0 \le 1 \le \mu$, and

$$Q_0(z)A_l(z) - Q_l(z)A_0(z) = z^{\rho+1} \cdot R_l(z),$$

$1 \le l \le \mu, R_l \in \mathbb{K}^{p\times p}[[z]]$ such that the matrix $[Q_0, \ldots, Q_\mu] \in \mathbb{K}^{p\times(\mu+1)p}[z]$ has full rank over $\mathbb{K}$.

Beside the classical scalar simultaneous Padé approximants ($p = 1, A_0(z) = 1$), Example 2.4 also includes the *simultaneous partial Padé approximation problem* where we have prescribed poles and zeros for the approximants [8]. Following [22], the question of irreducible Hermite Padé forms is of special interest, i.e., we also require that $[P_0(0), \ldots, P_\mu(0)] \in \mathbb{K}^{p\times(\mu+1)p}$ is different from zero (or moreover has full rank over $\mathbb{K}$). Similarly, in Example 2.4 we are interested in approximants where $Q_0(0)$ is a nonsingular matrix.

We remark that the order conditions in Examples 2.1–2.4 are all such that at least one solution exists for each approximation problem. In addition, the so-called *weak matrix Hermite Padé* and *weak matrix simultaneous Padé forms* are connected to Examples 2.3 and 2.4 (see [17]). In this case the order conditions are weakened to allow for more linearly independent solutions. Other examples of matrix-type generalizations of Padé approximants include Hermite Padé [11] and simultaneous Padé systems [12], [17]. These, however, only exist in certain cases.

Note that the matrix simultaneous Padé form is closely connected to a rectangular matrix Hermite Padé form if the interpolation conditions are written as follows:

$$\begin{bmatrix} A_1^T(z) \\ A_2^T(z) \\ A_3^T(z) \\ \vdots \\ A_\mu^T(z) \end{bmatrix} \cdot Q_0^T(z) + \begin{bmatrix} -A_0^T(z) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot Q_1^T(z) + \cdots + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ -A_0^T(z) \end{bmatrix} \cdot Q_\mu^T(z) = z^{\rho+1} \cdot \begin{bmatrix} R_1^T(z) \\ R_2^T(z) \\ R_3^T(z) \\ \vdots \\ R_\mu^T(z) \end{bmatrix}.$$

All examples given here are special cases of so-called vector Hermite Padé approximants.

Example 2.5 (vector Hermite Padé approximant). Let $m, s, \tau \in \mathbb{N}_0$, $m, s \ge 2, G_1, \ldots, G_m \in \mathbb{K}^{s\times 1}[[z]]$ and let $\mathbf{n}$ be a multiindex. Find linearly independent polynomial tuples $(P_1, \ldots, P_m), P_l \in \mathbb{K}[z]$ with $\deg P_l \le n_l, 1 \le l \le m$ such that

$$G_1(z)P_1(z) + \cdots + G_m(z)P_m(z) = z^\tau \cdot R(z),$$

with $R \in \mathbb{K}^{s\times 1}[[z]]$.

By setting

$$\text{for } 1 \le l \le m: \quad f_l(z) = (1, z, z^2, \ldots, z^{s-1}) \cdot G_l(z^s), \tag{4}$$

we see that computing vector Hermite Padé approximants of type $(\mathbf{n}, \tau)$ and dimension s is equivalent to the determination of PHPAs of type $(\mathbf{n}, \tau s, s)$, i.e. of the solution set $\mathcal{L}_0^{\tau s}$. Indeed, the above technique of converting a vector problem to a scalar problem via the raising of z to the sth power provides the motivation for Definition 1.1.

In Table 1, we list the particular choices of $m, \mathbf{n}, s, \sigma, \mathbf{F}$ with respect to Examples 2.1, 2.2, and 2.3. Instead of Example 2.4, we consider the special case of scalar simultaneous Padé approximation.

TABLE 1
Specification of the PHPA parameters used in (2) *for some matrix-type Padé approximation problems.*

Example	m	s	σ	$\mathbf{n}$, $\mathbf{F}$	Number solutions*
Classical Hermite Padé	m	1	$\|\mathbf{n}\| - 1$	$(n_1, \ldots, n_m)$, $\mathbf{F}^T(z) = (f_1(z), \ldots, f_m(z))$	1
2.1	$p+q$	p	$p(M+N+1)$	$(M, \ldots, M, N, \ldots, N)$, $\mathbf{F}^T(z) = (1, z, \ldots, z^{p-1}) \cdot (\mathbf{I}, -A(z^p))$	r
2.2	$p+q$	q	$q(M+N+1)$	$(M, \ldots, M, N, \ldots, N)$, $\mathbf{F}(z) = \begin{pmatrix} \mathbf{I} \\ -A(z^q) \end{pmatrix} \cdot (1, z, \ldots, z^{q-1})^T$	r
2.3	$p(\mu+1)$	p	$p(\rho-1)$	$(\rho_0 - 1, \ldots, \rho_0 - 1, \ldots, \rho_\mu - 1, \ldots, \rho_\mu - 1)$, $\mathbf{F}^T(z) = (1, z, \ldots, z^{p-1}) \cdot (A_0(z^p), \ldots, A_\mu(z^p))$	p
2.4 with $p = 1, A_0 = 1$	$\mu+1$	μ	$\mu(\rho+1)$	$(\rho - \rho_0, \ldots, \rho - \rho_\mu)$, $f_1(z) = -\sum_{1 \le j \le \mu} z^{j-1} A_j(z^\mu)$, $j \ge 1 : f_{j+1}(z) = z^{j-1}$	1

* Number of PHPA solutions required to construct the corresponding matrix-type Padé approximant.

3. Recursive computation of PHPA bases. In this section, we construct systems of m PHPAs by recurrence on σ. This allows us to describe all the PHPAs of type $(\mathbf{n}(\delta), \sigma, s), \delta \le 0$, when a fixed $s, \mathbf{F}$, and $\mathbf{n}$ are given. Therefore, we not only solve the Hermite Padé approximation problem of type $\mathbf{n}$ or the corresponding matrix-type Padé approximation problem (see §2) but also all subproblems of type $\mathbf{n}(\delta), \delta \le 0$ (cf. (3)) belonging to a "diagonal path" in the solution table. The recurrence formula and the resulting algorithm do not require any assumptions on the input data $\mathbf{F}$. Moreover, the algorithm is fast, i.e. it always has a complexity of $\mathcal{O}(\|\mathbf{n}\|^2)$ arithmetic operations, whereas the classical Gaussian algorithm, applied on the corresponding system of linear equations, has complexity $\mathcal{O}(\|\mathbf{n}\|^3)$ because it does not take into account the special structure of the matrix of coefficients. Finally, our method is also reliable, which in this context means that it also recognizes insoluble problems or gives representations if the solution sets of type $\mathbf{n}(\delta), \delta < 0$ are multidimensional (assuming that exact arithmetic is available). We remark that our algorithm does not consider the case of floating point arithmetic and hence does not consider the issue of numerical stability in the presence of roundoff errors.

Several fast algorithms for special cases of PHPAs are well known, but most of them require a normal or perfect solution table (i.e., PHPAs of different type are distinct). As far as we know, only the methods proposed in [19] for square matrix Padé approximation and the Jacobi–Perron continued fraction algorithms of [6] for simultaneous Padé approximation and [2], [4], [11], [12], [25] for scalar Hermite Padé approximation are also reliable. All of them still require slight assumptions on the input data ($A(0)$ regular, $\mathbf{F}(0)$ nontrivial); moreover, the algorithms of [11], [12], [19] might reach a complexity $\mathcal{O}(\|\mathbf{n}\|^3)$ if none of the subproblems of type $\mathbf{n}(\delta), \delta < 0$ has a unique solution.

For the special case $s = 1$ (i.e., scalar Hermite Padé approximation), the recurrence formula of our new algorithm is similar to that used in [2], [4], [25]. The *fast Gaussian algorithm* [2, §5] is motivated by the close connections to the factorization of the corresponding matrix of coefficients via the Gaussian algorithm with partial pivoting; a "special rule" reduces the complexity to $\mathcal{O}(\|\mathbf{n}\|^2)$. It provides solutions

to all subproblems on the diagonal path $(\mathbf{n}(\delta))_{\delta \le 0}$. The methods of [2], [4] both are developed for the more general M-Padé approximation problem (arbitrary interpolation knots); moreover, by the algorithm given in [4] we can compute solutions by recurrence on "arbitrary paths" or "staircases" $(\mathbf{n}_k)$ where the multiindex $\mathbf{n}_{k+1}$ differs from $\mathbf{n}_k$ by increasing one component (also the decreasing of a second component is allowed). Parallel to [2], [4], Van Barel and Bultheel proposed a fast, reliable method for computing Hermite Padé approximants on diagonal paths [25]. Their version is similar to [2] but notationally less complicated. The ideas developed in [26] for a recursive computation of vector M-Padé approximants have close connections to [4], [25]. The authors propose three alternative "basic steps" that include considerable freedom in solving certain subproblems.

There seems to be no connection between the methods described above and the reliable Jacobi–Perron continued fraction algorithm of [6] for simultaneous Padé approximation. For this approximation problem, using our formalism we obtain a more compact method with at most the same complexity; in addition, we get more information about singular cases.

Before describing bases for PHPA solution sets let us introduce the following definition.

DEFINITION 3.1 (defect, order). *The defect of a* $\mathbf{P} = (P_1, \dots, P_m) \in \mathbb{K}^m[z]$ *(with respect to the fixed multiindex* $\mathbf{n} = (n_1, \dots, n_m)$*) is*

$$\operatorname{dct} \mathbf{P} := \min_l \{n_l + 1 - \deg P_l\},$$

where the zero polynomial has degree $-\infty$*. The order of* $\mathbf{P}$ *(with respect to* $s \in \mathbb{N}$ *and* $\mathbf{F}$*) is defined by*

$$\operatorname{ord} \mathbf{P} := \sup \{\sigma \in \mathbb{N}_0 : \mathbf{P}(z^s) \cdot \mathbf{F}(z) = z^\sigma \cdot R(z) \quad \text{with } R \in \mathbb{K}[[z]]\}.$$

The definition of the defect is a natural extension of that found in the case of the M-Padé problem (cf. [3], [4]) and its special case of rational interpolation (some authors use a slightly different definition). The defect is also closely connected to the τ-degree of [25].

Using Definition 3.1, we get an equivalent characterization for PHPA solution sets:

$$\text{(5)} \quad \text{For } \sigma \in \mathbb{N}_0, \quad \delta \in \mathbb{Z} \cup \{+\infty\} : \mathcal{L}_\delta^\sigma = \{\mathbf{P} \in \mathbb{K}^m[z] : \operatorname{dct} \mathbf{P} > -\delta, \operatorname{ord} \mathbf{P} \ge \sigma\}.$$

Now we are able to describe so-called σ-bases of PHPAs.

DEFINITION 3.2 (σ-bases). *Let* $\sigma \in \mathbb{N}_0$*. The system* $\mathbf{P}_1, \dots, \mathbf{P}_m \in \mathbb{K}^m[z]$ *is called a* σ*-basis if and only if*:

(a) $\mathbf{P}_1, \dots, \mathbf{P}_m \in \mathcal{L}_{+\infty}^\sigma$, *i.e.*, $\operatorname{ord} \mathbf{P}_l \ge \sigma$.

(b) *For each* $\delta \in \mathbb{Z} \cup \{+\infty\}$ *and for each* $\mathbf{Q} \in \mathcal{L}_\delta^\sigma$ *there exists one and only one tuple of polynomials* $(\alpha_1, \dots, \alpha_m), \deg \alpha_l < \operatorname{dct} \mathbf{P}_l + \delta$ *such that* $\mathbf{Q} = \alpha_1 \cdot \mathbf{P}_1 + \dots + \alpha_m \cdot \mathbf{P}_m$.

Note that, as a consequence of Definition 3.2, a σ-basis $\mathbf{P}_1, \dots, \mathbf{P}_m$ must be linearly independent with respect to polynomial coefficients. Moreover, we have

$$\text{(6)} \quad \mathcal{L}_\delta^\sigma = \operatorname{span}\left\{z^j \cdot \mathbf{P}_l : 1 \le l \le m, 0 \le j < \operatorname{dct} \mathbf{P}_l + \delta\right\},$$

$$\text{(7)} \quad \dim \mathcal{L}_\delta^\sigma = \max\{\operatorname{dct} \mathbf{P}_1 + \delta, 0\} + \dots + \max\{\operatorname{dct} \mathbf{P}_m + \delta, 0\}.$$

The existence of σ-bases for the case $s = 1$ was given in [2]–[4], [25] and for the case $s > 1$ in [5]. Before giving an algorithm for their computation, let us state some simple rules for the defect and order of linear combinations of PHPAs.

LEMMA 3.3. *For* $\mathbf{P}, \mathbf{Q} \in \mathbb{K}^m[z], c \in \mathbb{K} \setminus \{0\}$:

(8)

$$\operatorname{dct}(c \cdot \mathbf{P}) = \operatorname{dct} \mathbf{P}, \quad \operatorname{dct}(\mathbf{P} + \mathbf{Q}) \geq \min\{\operatorname{dct} \mathbf{P}, \operatorname{dct} \mathbf{Q}\} \quad \operatorname{dct}(z \cdot \mathbf{P}) = \operatorname{dct} \mathbf{P} - 1,$$

(9)

$$\operatorname{ord}(c \cdot \mathbf{P}) = \operatorname{ord} \mathbf{P}, \quad \operatorname{ord}(\mathbf{P} + \mathbf{Q}) \geq \min\{\operatorname{ord} \mathbf{P}, \operatorname{ord} \mathbf{Q}\}, \quad \operatorname{ord}(z \cdot \mathbf{P}) = \operatorname{ord} \mathbf{P} + s.$$

Proof. The proof is left to the reader. □

From the characterization (5), it is clear that $\mathcal{L}_\delta^\sigma \subset \mathcal{L}_{\delta+1}^\sigma$ and $\mathcal{L}_\delta^{\sigma+1} \subset \mathcal{L}_\delta^\sigma$. In addition, if $\mathbf{P} \in \mathcal{L}_\delta^\sigma \setminus \mathcal{L}_\delta^{\sigma+1}$, i.e. $\operatorname{ord} \mathbf{P} = \sigma$, then from (8), (9) it is easy to see that for each $\mathbf{Q} \in \mathcal{L}_\delta^\sigma$ there exists a $c \in \mathbb{K}$ such that $\mathbf{Q} - c \cdot \mathbf{P} \in \mathcal{L}_\delta^{\sigma+1}$. This proves the statement

$$\mathcal{L}_\delta^{\sigma+1} \subset \mathcal{L}_\delta^\sigma, \qquad \dim \mathcal{L}_\delta^{\sigma+1} \geq \dim \mathcal{L}_\delta^\sigma - 1 \tag{10}$$

and already gives an idea about the computation of σ-bases by recurrence on the order as proposed in the procedure FPHPS (*fast power Hermite Padé solver*) below. We show in Theorem 3.4 that this method is both correct and produces the desired σ-bases.

FPHPS ALGORITHM

INPUT: $m \geq 2, s \in \mathbb{N}, \mathbf{F} = (f_1, \ldots f_m)^T$, multiindex $\mathbf{n} = (n_1, \ldots, n_m)$

INITIALIZATION: Let for $\sigma = 0, l = 1, \ldots, m$:
$d_{l,0} = n_l, \mathbf{P}_{l,0} = (0, \ldots, 0, 1, 0, \ldots, 0)$($l$th unit vector)

RECURSIVE STEP: For $\sigma = 0, 1, 2, \ldots$:
Let for $l = 1, \ldots, m$: $c_{l,\sigma} = z^{-\sigma} \cdot \mathbf{P}_{l,\sigma}(z^s) \cdot \mathbf{F}(z)|_{z=0}$ and $\Lambda_\sigma = \{l \ : \ c_{l,\sigma} \neq 0\}$
CASE $\Lambda_\sigma = \{\}$, then for $l = 1, \ldots, m$:
$\mathbf{P}_{l,\sigma+1} = \mathbf{P}_{l,\sigma}, d_{l,\sigma+1} = d_{l,\sigma}$
CASE $\Lambda_\sigma \neq \{\}$, then let $\pi = \pi_\sigma \in \Lambda_\sigma$ be defined by
$d_{\pi,\sigma} = \max\{d_{l,\sigma} \ : \ l \in \Lambda_\sigma\}$
and compute for $l = 1, \ldots, m$:
$l \in \Lambda_\sigma, l \neq \pi$: $\mathbf{P}_{l,\sigma+1} = \mathbf{P}_{l,\sigma} - \frac{c_{l,\sigma}}{c_{\pi,\sigma}} \cdot \mathbf{P}_{\pi,\sigma}, d_{l,\sigma+1} = d_{l,\sigma}$
$l \notin \Lambda_\sigma$: $\mathbf{P}_{l,\sigma+1} = \mathbf{P}_{l,\sigma}, d_{l,\sigma+1} = d_{l,\sigma}$
$l = \pi$: $\mathbf{P}_{\pi,\sigma+1} = z \cdot \mathbf{P}_{\pi,\sigma}, d_{\pi,\sigma+1} = d_{\pi,\sigma} - 1$

OUTPUT: For $\sigma = 0, 1, 2, \ldots$:
σ-bases $\mathbf{P}_{1,\sigma}, \ldots, \mathbf{P}_{m,\sigma}$ with $\operatorname{dct} \mathbf{P}_{l,\sigma} = d_{l,\sigma} + 1, l = 1, \ldots, m$, i.e.
for all δ: $\mathcal{L}_\delta^\sigma = \{\alpha_1 \cdot \mathbf{P}_{1,\sigma} + \cdots + \alpha_m \cdot \mathbf{P}_{m,\sigma} \ : \ \deg \alpha_l \leq d_{l,\sigma} + \delta\}$.

THEOREM 3.4 (Feasibility of method FPHPS). *Method* FPHPS *is well defined and gives the specified results.*

Proof. We show the assertion by induction on σ for a fixed δ.

The case $\sigma = 0$ follows immediately from the definition of $\mathcal{L}_\delta^0$. Hence, suppose $\sigma \geq 0$ and that the algorithm is correct for σ. We show that the algorithm produces the correct output for $\sigma + 1$. Note that by assumption $\operatorname{ord} \mathbf{P}_{l,\sigma} \leq \sigma$, i.e., its s-residual takes the form

$$\mathbf{P}_{l,\sigma}(z^s) \cdot \mathbf{F}(z) = z^\sigma \cdot R_l(z) \quad \text{with } R_l \in \mathbb{K}[[z]].$$

Hence, $c_{l,\sigma} = R_l(0)$ and the recurrence step is well defined. By construction we have

$$\operatorname{ord} \mathbf{P}_{l,\sigma+1} \geq \sigma + 1 \quad \text{and} \quad \operatorname{dct} \mathbf{P}_{l,\sigma+1} \geq d_{l,\sigma+1} + 1.$$

Moreover, it is easy to see that with $\mathbf{P}_{1,\sigma},\dots,\mathbf{P}_{m,\sigma}$ also $\mathbf{P}_{1,\sigma+1},\dots,\mathbf{P}_{m,\sigma+1}$ are linearly independent with respect to polynomial coefficients.

Consider first the case when $\mathcal{L}_\delta^\sigma = \mathcal{L}_\delta^{\sigma+1}$. By assumption, each $\mathbf{Q} \in \mathcal{L}_\delta^{\sigma+1}$ then has a representation

$$\mathbf{Q} = \alpha_1 \cdot \mathbf{P}_{1,\sigma} + \cdots + \alpha_m \cdot \mathbf{P}_{m,\sigma}, \quad \deg\ \alpha_l < \operatorname{dct} \mathbf{P}_{l,\sigma} + \delta.$$

This is already a suitable linear combination of $\mathbf{P}_{1,\sigma+1},\dots,\mathbf{P}_{m,\sigma+1}$. To see this, note that with $\alpha_l \neq 0$, we get $\operatorname{dct} \mathbf{P}_{l,\sigma} + \delta > 0$ and $\mathbf{P}_{l,\sigma} \in \mathcal{L}_\delta^\sigma = \mathcal{L}_\delta^{\sigma+1}$; hence, $c_{l,\sigma} = 0$ and $\mathbf{P}_{l,\sigma+1} = \mathbf{P}_{l,\sigma}$.

The case where $\mathcal{L}_\delta^\sigma \neq \mathcal{L}_\delta^{\sigma+1}$ is also easy to handle. Let

$$\mathcal{L}_\delta := \{\alpha_1 \cdot \mathbf{P}_{1,\sigma+1} + \cdots + \alpha_m \cdot \mathbf{P}_{m,\sigma+1} : \deg\ \alpha_l < \operatorname{dct} \mathbf{P}_{l,\sigma+1} + \delta\},$$

so that in view of Lemma 3.3 we have $\mathcal{L}_\delta \subset \mathcal{L}_\delta^{\sigma+1}$. On the other hand, the dimension of $\mathcal{L}_\delta$ can be estimated as follows:

$$\begin{aligned}
\dim\ L_\delta &= \max\ \{\operatorname{dct} \mathbf{P}_{1,\sigma+1} + \delta, 0\} + \cdots + \max\ \{\operatorname{dct} \mathbf{P}_{m,\sigma+1} + \delta, 0\}, \\
&\geq \max\ \{d_{1,\sigma+1} + 1 + \delta, 0\} + \cdots + \max\ \{d_{m,\sigma+1} + 1 + \delta, 0\}, \\
&\geq \max\ \{d_{1,\sigma} + 1 + \delta, 0\} + \cdots + \max\ \{d_{m,\sigma} + 1 + \delta, 0\} - 1, \\
&= \dim\ \mathcal{L}_\delta^\sigma - 1 = \dim\ \mathcal{L}_\delta^{\sigma+1},
\end{aligned}$$

where for the last two equalities we have applied (7) and (10). Consequently, $\mathcal{L}_\delta = \mathcal{L}_\delta^{\sigma+1}$, and we have equality in the estimation above. For all $\Delta \geq \delta$ we also have that $\mathcal{L}_\Delta^{\sigma+1} \neq \mathcal{L}_\Delta^\sigma$, since by definition $\emptyset \neq \mathcal{L}_\delta^\sigma \setminus \mathcal{L}_\delta^{\sigma+1} \subset \mathcal{L}_\Delta^\sigma \setminus \mathcal{L}_\Delta^{\sigma+1}$. Therefore, the above equations are also valid if we replace δ by $\Delta \geq \delta$. Choosing Δ sufficiently large, we can conclude that $\operatorname{dct} \mathbf{P}_{l,\sigma+1} = d_{l,\sigma+1} + 1$ for $l = 1,\dots,m$ which proves the theorem. □

4. Some properties of the FPHPS algorithm. In this section, we discuss some properties of the σ-bases obtained by the procedure FPHPS. In particular, we are interested in simple conditions describing whether some PHPAs are irreducible and whether given PHPAs $\mathbf{P}_1,\dots,\mathbf{P}_\lambda$ (and its values at zero) are linearly independent with respect to polynomial coefficients (and constant coefficients, respectively)—questions that as explained in §2 naturally arise in the context of matrix Padé approximation. In addition, for multidimensional solution sets, we classify PHPAs having "best" approximation properties, i.e., maximal order and/or minimal degree. The complexity of method FPHPS is determined at the end of this section.

Let Λ_σ and π_σ (for a given σ) be defined as in the FPHPS algorithm. As given in all applications of Table 1, in the sequel we only discuss the case $s \leq m$ and $\Lambda_0 \neq \{\ \},\dots,\Lambda_{s-1} \neq \{\ \}$. This is equivalent to the fact that the matrix $(\mathbf{F}(0), \mathbf{F}'(0),\dots,\mathbf{F}^{(s-1)}(0))$ has full rank. In Theorem 4.1, we summarize some facts about reducible PHPAs. These results are generalizations of ideas appearing in [25].

THEOREM 4.1. (a) *For all $\sigma \geq s$, we have* $\operatorname{card} \Lambda_\sigma \geq 1$, *more precisely*

$$\begin{aligned}
&\pi_{\sigma-s} \in \Lambda_\sigma \subset L_\sigma \cup \{\pi_{\sigma-s}\}, \\
&\text{where } L_\sigma := \{1,\dots,m\} \setminus \{\pi_{\sigma-s}, \pi_{\sigma-s+1},\dots,\pi_{\sigma-1}\}.
\end{aligned} \tag{11}$$

(b) *Let U denote the $(m-s)$ dimensional subspace of vectors that are orthogonal to all $\mathbf{F}(0), \mathbf{F}'(0),\dots,\mathbf{F}^{(s-1)}(0)$. Then for $\sigma \geq s$*

$$\operatorname{span}\ \{\mathbf{P}_{l,\sigma}(0) : l \in L_\sigma\} = U \quad \text{and for all } l \notin L_\sigma : \mathbf{P}_{l,\sigma}(0) = 0. \tag{12}$$

Proof. Note that in the FPHPS algorithm we always have ord $\mathbf{P}_{\pi_\sigma,\sigma+1} = \sigma + s$. Therefore, $\pi_\sigma \in \Lambda_{\sigma+s} \neq \{\ \}$ but $\pi_\sigma \notin \Lambda_{\sigma+1}, \ldots, \pi_\sigma \notin \Lambda_{\sigma+s-1}$. This proves (a). The second part of (b) follows directly from the fact that $\mathbf{P}_{l,\sigma+1} = \mathbf{P}_{l,\sigma}$ for all $l \notin L_{\sigma+1} \cup L_\sigma$, and $\mathbf{P}_{l,\sigma+1}(0) = 0$ for $l = \pi_\sigma$. The first assertion of (b) can be shown by a simple recurrence argument on $\sigma \geq s$. Let

$$U_\sigma := \text{span}\,\{\mathbf{P}_{l,\sigma}(0) : l \in L_\sigma\}.$$

Then $U_\sigma \subset U$ since we have ord $\mathbf{P}_{l,\sigma} \geq s$. With $\mathbf{P}_{l,\sigma}(0), l \in L_\sigma$, also the vectors $\mathbf{P}_{l,\sigma+1}(0), l \in L_\sigma \bigcap L_{\sigma+1}$, together with $\mathbf{P}_{\pi_\sigma,\sigma}(0)$ are linearly independent. From the recurrence relations we know that $\mathbf{P}_{\pi_{\sigma-s},\sigma}(0) = 0$ and $\mathbf{P}_{\pi_{\sigma-s},\sigma+1}(0) = c \cdot \mathbf{P}_{\pi_\sigma,\sigma}(0)$ with $c \neq 0$. Consequently, $\mathbf{P}_{l,\sigma+1}(0), l \in L_{\sigma+1}$, are linearly independent that proves part (b). □

Supposing that the vector $\mathbf{F}$ contains only polynomial entries, we expect that the solution set $\mathcal{L}_\delta^\sigma$ becomes stationary for sufficiently large σ. In contrast, due to Theorem 4.1(a) the σ-bases will always change if σ is increased. In fact, we observe that for large σ, the nonconstant part of the σ-basis described by the sets Λ_σ consists only of approximants with defect smaller than $-\delta$ and that, for sufficiently large σ, for the representation (6) of the solution set $\mathcal{L}_\delta^\sigma$ we need at most $(m - s)$ elements of the σ-basis.

Theorem 4.1(b) yields a simple criterion determining whether the solution set $\mathcal{L}_\delta^\sigma$ contains an irreducible element. By definition, the components of an element of the σ-basis can only have a common factor that vanishes at zero. Hence, there exists an element $\mathbf{P}$ of $\mathcal{L}_\delta^\sigma$ being irreducible, i.e., $\mathbf{P}(0) \neq 0$, if and only if there is an $l \in L_\sigma$ with $d_{l,\sigma} \geq -\delta$. Moreover, we immediately get the following corollary.

COROLLARY 4.2. (a) *$\mathcal{L}_\delta^\sigma$ contains $\lambda \leq m$ elements $\mathbf{P}_1, \ldots, \mathbf{P}_\lambda$ being linearly independent over $\mathbb{K}[z]$ if and only if there are distinct $l_1, \ldots, l_\lambda \in \{1, \ldots, m\}$ with $d_{l_j,\sigma} \geq -\delta$.*

(b) *$\mathcal{L}_\delta^\sigma$ contains $\lambda \leq m - s$ elements $\mathbf{P}_1, \ldots, \mathbf{P}_\lambda$ such that $\mathbf{P}_1(0), \ldots, \mathbf{P}_\lambda(0)$ are linearly independent over $\mathbb{K}$ if and only if there are distinct $l_1, \ldots, l_\lambda \in L_\sigma$ with $d_{l_j,\sigma} \geq -\delta$.*

(c) *In both cases linearly independent approximants from $\mathcal{L}_\delta^\sigma$ are given by $\mathbf{P}_j = \mathbf{P}_{l_j,\sigma}, j = 1, \ldots, \lambda$.*

In most applications, the first s components of $\mathbf{F}$ take the simple form $f_j(z) = z^{j-1}$. Here we consider the first s components $\mathbf{p}$ and the last $(m - s)$ components $\mathbf{q}$ of a PHPA $\mathbf{P} = (\mathbf{p}, \mathbf{q})$ separately and ask for approximants $\mathbf{P}_1, \ldots, \mathbf{P}_\lambda \in \mathcal{L}_\delta^\sigma$ with $\mathbf{q}_1, \ldots, \mathbf{q}_\lambda$ (or $\mathbf{q}_1(0), \ldots, \mathbf{q}_\lambda(0)$) being linearly independent. Here also the criteria given in Corollary 4.2.(a) and (b) can be applied as long as we can guarantee there is no $\mathbf{P} = (\mathbf{p}, \mathbf{q}) \in \mathcal{L}_\delta^\sigma$ with $\mathbf{p} \neq 0$ and $\mathbf{q} = 0$ ($\mathbf{p}(0) \neq 0$ and $\mathbf{q}(0) = 0$, respectively). But due to the simple form of $\mathbf{F}$ it can be easily verified that $\mathbf{P} = (\mathbf{p}, \mathbf{q}) \in \mathcal{L}_\delta^\sigma$ with $\mathbf{q}(0) = 0$ and $\sigma \geq s$ also implies that $\mathbf{p}(0) = 0$. Similarly, if $s \cdot (n_j + \delta) + j \leq \sigma$ for $j = 1, \ldots, s$ (which for the most interesting PHPA cases of §2 is true) and $\mathbf{P} = (P_1, \ldots, P_m) = (\mathbf{p}, 0) \in \mathcal{L}_\delta^\sigma$, then $\mathbf{p}$ must also be identical zero since ord $\mathbf{P} \leq \max\{s \cdot \deg P_j + j - 1 : j = 1, \ldots, s\}$.

If the solution set is multidimensional, we are interested in classifying particular solutions that have certain uniqueness properties. The concept of approximants with correct degree satisfying "best possible" order conditions is discussed in Corollary 4.3.

COROLLARY 4.3. *Let each $\mathbf{P} \in \mathbb{K}^m[z]$ have finite order and let $\delta + \min\{n_1, \ldots, n_m\} \geq 0$. Consider the problem of finding "optimal" PHPAs $\mathbf{P}_1, \ldots, \mathbf{P}_\lambda, \lambda \leq m - s$ with*

(i) $\mathbf{P}_1(0), \ldots, \mathbf{P}_\lambda(0)$ *are linearly independent,*
(ii) $\operatorname{dct} \mathbf{P}_1 > -\delta, \ldots, \operatorname{dct} \mathbf{P}_\lambda > -\delta$,
(iii) *the number* $(\operatorname{ord} \mathbf{P}_1 + \cdots + \operatorname{ord} \mathbf{P}_\lambda)$ *is maximal,*
(iv) $\operatorname{ord} \mathbf{P}_1 =: \sigma(1) > \operatorname{ord} \mathbf{P}_2 =: \sigma(2) > \cdots > \operatorname{ord} \mathbf{P}_\lambda =: \sigma(\lambda)$

(*it is easy to see that condition* (iv) *only implies a particular ordering for the* PHPAs *determined by* (i), (ii), (iii)). *A solution for this problem is given by*

$$\sigma(j) := \max\{\sigma : \operatorname{card}\ \{l \in L_\sigma\ :\ d_{l,\sigma} \geq -\delta\} \geq j\}$$

and $\mathbf{P}_j = \mathbf{P}_{\pi_{\sigma(j)},\sigma(j)}, j = 1, \ldots, \lambda$.

Corollary 4.3 is a canonical generalization of the optimal Hermite Padé form of type $\mathbf{n}(\delta)$ of [22] ($s = \lambda = 1$). Paszkowski [22] speaks of nonexistent optimal Hermite Padé forms if $\mathbf{P}_1$ is not unique, i.e., if there is a further (necessarily reducible) PHPA $\mathbf{P}_0$ with $\operatorname{dct} \mathbf{P}_0 > -\delta$ and $\operatorname{ord} \mathbf{P}_0 > \operatorname{ord} \mathbf{P}_1$. For $\lambda = m - s = 1$, for example, scalar simultaneous (partial) Padé approximation, our approach is closely connected to a concept proposed by de Bruin [8] for nonnormal solution tables. Note that although in view of Corollary 4.2.(b), the numbers $\sigma(1), \ldots, \sigma(\lambda)$ are unique, we might get several tuples of optimal PHPAs being essentially different. The significance of the integer $\sigma(\lambda)$ for matrix Padé approximation is discussed at the end of §5.

Following Corollary 4.3, we always find irreducible approximants with correct degree, but the order condition might be weakened. In contrast, Van Barel and Bultheel [24], [26] look for irreducible approximants with correct order and a type of minimal degree. More precisely, instead of (ii)–(iv), the conditions

(v) $\operatorname{ord} \mathbf{P}_1 \geq \sigma, \ldots, \operatorname{ord} \mathbf{P}_\lambda \geq \sigma$,
(vi) the number $(\operatorname{dct} \mathbf{P}_1 + \cdots + \operatorname{dct} \mathbf{P}_\lambda)$ is maximal

are imposed. As above, this problem will not generally have a unique solution. However, the method FPHPS also gives a solution for this problem: due to Corollary 4.2(b) we can take those λ approximants $\mathbf{P}_{l,\sigma}, l \in L_\sigma$ with maximal defect.

The problem of uniqueness for both concepts is illustrated in Example 4.4.

Example 4.4. Let

$$m = 4, \qquad s = 2, \qquad \mathbf{n} = (2,2,2,2), \qquad \delta = 0,$$
$$\mathbf{F}(z) = \left(1, z, \frac{z}{1-z^4} + z^{10}, \frac{z}{1+z^4} + z^{12}\right)^T + \mathcal{O}\left(z^{16}\right).$$

An application of FPHPS gives the values $\pi_0, \pi_1, \ldots, \pi_{13} = 1,2,1,2,1,3,1,3,1,4,2,4,3,4$. In particular, we obtain a σ-basis for $\sigma = 10$ (output in matrix form with the rows $\mathbf{P}_{1,10}, \mathbf{P}_{2,10}, \mathbf{P}_{3,10}$, and $\mathbf{P}_{4,10}$ as the basis elements) as

$$\mathbf{P}_{10}(z) = \begin{bmatrix} z^5 & 0 & 0 & 0 \\ 0 & z^2 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 1-z^2 & -\frac{1}{2}+z^2 & -\frac{1}{2} \\ 0 & -2z & z & z \end{bmatrix},$$

with s-residuals

$$\mathbf{P}_{10}(z^2) \cdot \mathbf{F}(z) = \begin{bmatrix} z^{10} + \mathcal{O}\left(z^{26}\right) \\ -\frac{z^{10}}{2} + \frac{z^{12}}{2} - z^{13} + \mathcal{O}\left(z^{16}\right) \\ -\frac{z^{10}}{2} - \frac{z^{12}}{2} + z^{13} + z^{14} + \mathcal{O}\left(z^{16}\right) \\ 2z^{11} + z^{12} + z^{14} + \mathcal{O}\left(z^{18}\right) \end{bmatrix}.$$

The defects for this basis are $-2, 1, 1$ and 2, respectively. Hence, $\mathcal{L}_0^{10}$ does not have dimension 2 (as expected from comparing the number of equations and unknowns) but 4. For $\lambda = 1$, a particular solution with "minimal degree" (satisfying conditions (i), (v), and (vi) above) is given by $a \cdot \mathbf{P}_{2,10} + b \cdot \mathbf{P}_{3,10} + (cz + d) \cdot \mathbf{P}_{4,10}$ with arbitrary constants $a, b, c, d, |a| + |b| \neq 0$. A particular solution with "maximal order" $\sigma(1) = 12$ (satisfying conditions (i), (ii), (iii), and (iv) above) is given by $a \cdot \mathbf{P}_{3,12} + b \cdot \mathbf{P}_{4,12} = a \cdot (\mathbf{P}_{3,10} - \mathbf{P}_{2,10}) + b \cdot z \cdot \mathbf{P}_{4,10}$ with arbitrary constants $a, b, a \neq 0$ (the solution proposed in Corollary 4.3 equals $\mathbf{P}_{3,12}$).

Consider now the problem of determining the complexity of the FPHPS algorithm. For simplicity, we still impose the conditions before Theorem 4.1 (otherwise, the complexity will be still smaller). As seen in §2, in most applications one must determine σ-bases of PHPAs for $\sigma \approx \|\mathbf{n}\|$. To determine the number of arithmetic operations (AO) required for the computation of a $\|\mathbf{n}\|$-basis, we essentially only have to take into account the computation of $c_{1,\sigma}, \ldots, c_{m,\sigma}$ and of $\mathbf{P}_{1,\sigma+1}, \ldots, \mathbf{P}_{m,\sigma+1}, 0 \leq \sigma < \|\mathbf{n}\|$. Here the complexity strongly depends on the parameters $\mathbf{F}$ and s.

THEOREM 4.5 (Complexity). *The* FPHPS *algorithm for computing* PHPAs *of order* $\sigma = 0, 1, \ldots, \|\mathbf{n}\|$ *has a complexity of at most*

$$4(m - s) \cdot \|\mathbf{n}\|^2 + \mathcal{O}\left(m^2 \cdot \|\mathbf{n}\|\right) \text{ AO}, \tag{13}$$

roughly half additions and half multiplications plus $\mathcal{O}(m \cdot \|\mathbf{n}\|)$ *divisions. At least for the case* $\mathbf{n} = (n, \ldots, n)$, *we obtain the sharper bound*

$$\left(1 - \frac{s}{m}\right) \cdot (2m - \text{card } L) \cdot \|\mathbf{n}\|^2 + \mathcal{O}\left(m^2 \cdot \|\mathbf{n}\|\right) \text{ AO}, \tag{14}$$

where $L = \{l : f_l(z) = z^j \text{ with a } j \in \mathbb{N}_0\}$.

Proof. Since $c_{\pi_{\sigma-s},\sigma} = c_{\pi_{\sigma-s},\sigma-s}$ and $P_{\pi_\sigma,\sigma+1}$ can be easily determined by shifting some coefficients, for the complexity it remains to consider the computation of at most $c_{l,\sigma}$ and $\mathbf{P}_{l,\sigma+1}$ for $l \in L_{\sigma+1}$. In addition, we are not interested in PHPAs with dct $\mathbf{P}_{l,\sigma} \leq 0$, since they do not occur in the solution sets $\mathcal{L}_\delta^\sigma, \delta \leq 0$ (cf. (6)). Therefore, the degree of the λth component of $\mathbf{P}_{l,\sigma}$ is bounded by $n_\lambda - d_{l,\sigma} \leq n_\lambda$ and we require for loop number σ the number of at most $2 \cdot \sum_{l \in L_{\sigma+1}} \sum_{\lambda=1}^{m} (n_\lambda + 1) + \mathcal{O}(m^2) = 2(m-s) \cdot \|\mathbf{n}\| + \mathcal{O}(m^2)$ additions/subtractions and the same number of multiplications that totally gives a complexity as stated in (13). For the case $\mathbf{n} = (n, \ldots, n)$, we can apply the relation

$$2 \cdot \sum_{\sigma=0}^{\|\mathbf{n}\|-1} \sum_{l \in L_{\sigma+1}} (d_{l,\sigma} + 1) \geq \cdots \geq \|\mathbf{n}\|^2 - 2 \cdot s \cdot \sum_{l=1}^{m} \sum_{j=0}^{n+1} j,$$

which by using similar arguments leads to (14). □

TABLE 2
Complexity for solving matrix-type Padé approximation problems.

Example	Complexity via (13)	Via (14), special case
Classical Hermite Padé	$4(m-1)\cdot\|\mathbf{n}\|^2+\mathcal{O}(m^2\cdot\|\mathbf{n}\|)$	for $n_1=\cdots=n_m$: $2(m-1)\cdot\|\mathbf{n}\|^2+\mathcal{O}(m^2\cdot\|\mathbf{n}\|)$
2.1	$4q[p(M+1)+q(N+1)]^2$ $+\mathcal{O}((p+q)^2\cdot(M+N))$	for $M=N$: $q(2q+p)(p+q)(M+1)^2$ $+\mathcal{O}((p+q)^2\cdot M)$
2.2	$4p[q(M+1)+p(N+1)]^2$ $+\mathcal{O}((p+q)^2\cdot(M+N))$	for $M=N$: $p(2p+q)(p+q)(M+1)^2$ $+\mathcal{O}((p+q)^2\cdot M)$
2.3	$4\mu p^3\rho^2+\mathcal{O}(\mu^2p^4\rho)$	for $\rho_0=\cdots=\rho_\mu$: $2\mu p^3\rho^2+\mathcal{O}(\mu^2p^4\rho)$
2.4 with $p=1, A_0=1$	$4\mu^2\rho^2+\mathcal{O}(\mu^4\rho)$	for $\rho_0=\cdots=\rho_\mu$: $\frac{\mu+2}{\mu+1}\mu^2\rho^2+\mathcal{O}(\mu^4\rho)$

It should be mentioned that our algorithm can be implemented very efficiently on a vector or on a parallel processor (with, e.g., m or $\|\mathbf{n}\|$ processors). The complexity of our algorithm for the examples of §2 is given in Table 2, whereas in Table 3 some solved subproblems and their corresponding PHPA solution space are listed.

5. An example of matrix Padé approximation. In this section, we give an example of a matrix Padé approximation problem computed using the FPHPS algorithm. Let

$$A(z)=\begin{bmatrix}1+z^2+2z^4-z^5+z^6+\mathcal{O}\left(z^8\right) & z^7+\mathcal{O}\left(z^8\right)\\ -z^5+\mathcal{O}\left(z^8\right) & 1+z^2+z^4+z^7+\mathcal{O}\left(z^8\right)\end{bmatrix}$$

and consider the problem of determining a (2, 3) right-hand matrix Padé form for $A(z)$. Thus, we are looking for 2×2 matrix polynomials P and Q of degree at most 2 and 3, respectively, such that

$$A(z)\cdot Q(z)-P(z)=z^6\cdot R(z)$$

for some matrix power series R. The suitable choice of the parameters is stated in Table 1, row 2. Note that, for any PHPA (P_1,P_2,P_3,P_4) of type $(M,M,N,N),2(M+N+1),2)$, the components P_1 and P_2 correspond to a column of an (M,N) right-hand matrix Padé numerator, whereas P_3,P_4 correspond to a column of the denominator (cf. Table 3, for left-hand matrix Padé approximation, P_1,P_2 and P_3,P_4 correspond to rows of numerator and denominator, respectively).

Setting $s=2,\mathbf{n}=(2,2,3,3)$ and

$$\begin{aligned}\mathbf{F}^T(z)&=[1,z]\cdot\left[\mathbf{I},\,-A\left(z^2\right)\right]\\&=\left[1,z,-1-z^4-2z^8+z^{10}+z^{11}-z^{12}+\mathcal{O}\left(z^{16}\right),\right.\\&\qquad\left.-z-z^5-z^9-z^{14}-z^{15}+\mathcal{O}\left(z^{16}\right)\right]\end{aligned}$$

and using the FPHPS algorithm gives a σ-basis for $\sigma=12$ (output in matrix form with the rows as the basis elements) as

$$\begin{bmatrix}-z-z^2+z^3 & 0 & -z-z^2+2z^3+z^4 & 0\\ 1+z-z^2 & z^3 & 1+z-2z^2-z^3 & z^3\\ -z^2 & 0 & -z^2+z^4 & 0\\ 0 & -1 & 0 & -1+z^2\end{bmatrix}.$$

TABLE 3
Some matrix-type Padé subproblems solved by FPHPS *and their corresponding* PHPA *solution spaces, parameters* $m, \mathbf{n}, s, \sigma, \mathbf{F}$ *as in Table* 1.

Example	Type of subproblem	PHPA solution space
Classical Hermite Padé	$(n_1 - j, \ldots, n_m - j), j \leq \min\{n_l + 1\}$	$\mathcal{L}_{-j}^{\sigma - j \cdot m}$
2.1	$(M - j, N - j), j \leq \min\{M + 1, N + 1\}$	$\mathcal{L}_{-j}^{\sigma - 2 \cdot j \cdot s}$ contains rows of (P^T, Q^T)
2.2	$(M - j, N - j), j \leq \min\{M + 1, N + 1\}$	$\mathcal{L}_{-j}^{\sigma - 2 \cdot j \cdot s}$ contains rows of (P, Q)
2.3	$(\rho_0 - j, \ldots, \rho_\mu - j), j \leq \min\{\rho_l\}$	$\mathcal{L}_{-j}^{\sigma - j \cdot m}$ contains rows of $(P_0, \ldots, P_\mu)$
2.4 with $p = 1, A_0 = 1$	$(\rho_0 - j, \ldots, \rho_\mu - j), j \leq \min\{\rho_l\}$	$\mathcal{L}_{-j \cdot s}^{\sigma - j \cdot s \cdot m}$

The defects for this basis are 0, 0, 0, and 2, respectively. Therefore, a basis for the solution space $\mathcal{L}_0^{12}$, as a finite-dimensional space over $\mathbb{K}$, is given by $(a + b \cdot z) \cdot \mathbf{P}_{4,12} = (a + b \cdot z) \cdot [0, -1, 0, -1 + z^2]$, with a and b being arbitrary constants. Translating the solution space basis into matrix form implies that the columns of P and Q are generated by

$$(a + bz) \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \text{and} \quad (a + bz) \cdot \begin{bmatrix} 0 \\ -1 + z^2 \end{bmatrix},$$

respectively. This gives a right matrix Padé form of type (2, 3) for $A(z)$ as

$$P(z) = \begin{bmatrix} 0 & 0 \\ -1 & -z \end{bmatrix} \quad \text{and} \quad Q(z) = \begin{bmatrix} 0 & 0 \\ -1 + z^2 & -z + z^3 \end{bmatrix}.$$

In this case, such a matrix Padé form is unique up to multiplication on the right by a nonsingular 2×2 matrix. In particular, note that it is not possible to construct a right matrix Padé fraction of type (2, 3) in this instance.

The left matrix Padé forms of type (2, 3) for $A(z)$ can also be computed by the FPHPS procedure. Setting $s = 2, \mathbf{n} = (2, 2, 3, 3)$ and

$$\mathbf{F}(z) = \begin{bmatrix} \mathbf{I} \\ -A\left(z^2\right) \end{bmatrix} \cdot \begin{bmatrix} 1 \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ z \\ -1 - z^4 - 2z^8 + z^{10} + \mathcal{O}\left(z^{12}\right) \\ -z - z^5 - z^9 + z^{10} + \mathcal{O}\left(z^{12}\right) \end{bmatrix},$$

and computing the σ-basis for $\sigma = 12$ gives

$$\begin{bmatrix} -1 + z^2 & 1 & -1 + 2z^2 & 1 - z^2 \\ 0 & z^4 & 0 & z^4 \\ -z & -1 & -z + z^3 & -1 + z^2 \\ 0 & -z & 0 & -z + z^3 \end{bmatrix}.$$

In this case the defects are $1, -1, 1$, and 1, respectively, so the solution space $\mathcal{L}_0^{12}$ is of the form $a \cdot \mathbf{P}_{1,12} + b \cdot \mathbf{P}_{3,12} + c \cdot \mathbf{P}_{4,12}$ with a, b, and c arbitrary constants. Again translating the basis information to matrix form implies that the rows of P and Q are generated by

$$a \cdot \left[-1 + z^2, 1\right] + b \cdot [-z, -1] + c \cdot [0, -z]$$

and

$$a \cdot \left[-1+2z^2, 1-z^2\right] + b \cdot \left[-z+z^3, -1+z^2\right] + c \cdot \left[0, -z+z^3\right],$$

respectively. Unlike the previous example, there is not one Padé form that is unique up to left multiplication by a nonsingular matrix of scalars. One possibility for a left matrix Padé form in this case is

$$P(z) = \begin{bmatrix} -z & -1 \\ -1+z^2 & 1 \end{bmatrix} \quad \text{and} \quad Q(z) = \begin{bmatrix} -z+z^3 & -1+z^2 \\ -1+2z^2 & 1-z^2 \end{bmatrix}.$$

Note that the denominator has a nonzero determinant, indeed that $Q(0)$ is nonsingular. Therefore, unlike the case for approximants on the right, one can always form the rational expression $Q(z)^{-1} \cdot P(z)$.

Using the FPHPS algorithm in the above example also determines, at no added cost, the σ-basis for $\mathcal{L}^4_{-2}$ and $\mathcal{L}^8_{-1}$. Hence, the right matrix Padé forms of type (0, 1)

$$P(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and (1, 2)

$$P(z) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad Q(z) = \begin{bmatrix} -1+z^2 & 0 \\ 0 & -1+z^2 \end{bmatrix}$$

(determined uniquely up to matrix multiplication on the right in both cases) are byproducts of the previous computation. In addition, one can continue the computation to determine the matrix Padé form of type (3, 4) since the σ-basis for $\mathcal{L}^{12}_0$ can be used to determine the σ-basis for $\mathcal{L}^{16}_1$. In the case of the right matrix Padé form of type (3, 4) this gives (again unique up to matrix multiplication on the right)

$$P(z) = \begin{bmatrix} 0 & 1+z/5+11/5z^2+4/5z^3 \\ -z^2 & 1/5-2/5z+z^3 \end{bmatrix},$$

$$Q(z) = \begin{bmatrix} 0 & 1+z/5+6/5z^2+3/5z^3-16/5z^4 \\ -z^2+z^4 & -1/5-2/5z+1/5z^2+7/5z^3 \end{bmatrix},$$

an example where the denominator matrix polynomial Q is nonsingular but has a singular leading term $Q(0)$.

Our example shows that, in general, the matrix Padé approximation problem does not have a unique rational solution as in the scalar case. Moreover, there are three distinct and possible forms of a denominator matrix polynomial Q. First, the case occurs when $Q(z)$ is singular for all z and hence no matrix rational form exists; this type of degeneracy is not found in the scalar case. Second, it is possible that $Q(0)$ is nonsingular (cf. Corollary 4.2(a) and (b) and the following remarks). Here we can form $P(z) \cdot Q(z)^{-1}$ and its matrix power series agrees with $A(z)$ to the full order condition. Finally, if $Q(z)$ is nonsingular for some z, but $Q(0)$ is singular, we can cancel P and Q by a common matrix polynomial factor on the right. Here, similarly to the degenerate case found in scalar Padé approximation, the resulting matrix rational form $P(z) \cdot Q(z)^{-1}$ does not agree anymore with $A(z)$ to the full order condition.

Note that the concept as proposed in Corollary 4.3 ($2\lambda = 2s = m$) always leads to a matrix Padé-like form with correct degree and maximal order $[\sigma(s)/s]$ (perhaps less than $(M+N+1)$ as required for matrix Padé approximants) where by forming the rational function $P(z) \cdot Q(z)^{-1}$ we do not obtain an additional order deflation. In fact one can show that there is no other rational function of the form $P(z) \cdot Q(z)^{-1}$ satisfying the degree constraints and having an order greater than $[\sigma(s)/s]$.

6. A superfast PHPA solver. In §4 we have shown that the FPHPS algorithm computes a σ-basis with quadratic complexity. This is better than using methods such as Gaussian elimination and is optimal in special cases for arbitrary fields $\mathbb{K}$. However, when the field $\mathbb{K}$ allows for fast polynomial multiplication via the use of the FFT (cf. [15]), then there are faster methods in special cases. For example, when $s = 1$ and $m = 2$ (i.e., the case of Padé approximation) the algorithms of Brent, Gustavson, and Yun [9] and Cabay and Choi [10] compute these approximants with the superfast complexity $\mathcal{O}(\sigma \log^2 \sigma)$. Similarly, a recent algorithm of Cabay and Labahn [12] also solves the Hermite Padé and simultaneous Padé problems with superfast complexity. In this section we describe a second algorithm that takes advantage of fast polynomial multiplication when solving the PHPA problem. The new algorithm has the advantage of always being superfast—the algorithm of [12] sometimes slows down to quadratic or even cubic complexity (if most of the subproblems of type $\mathbf{n}(\delta), \delta < 0$ do not have a unique solution), although in practical problems this is rare.

The FPHPS algorithm of §3 provides a σ-basis $\mathbf{P}_1, \ldots, \mathbf{P}_m$ with respect to given $\mathbf{F}$, $\mathbf{n}$, and σ (and a fixed parameter s). For convenience, we arrange the $\mathbf{P}_l = (P_{l,1}, \ldots, P_{l,m})$ in a matrix

$$\mathbf{P} = (P_{l,\lambda})_{l=1,\ldots,m}^{\lambda=1,\ldots,m}.$$

Then with $\mathbf{d} := (d_1, \ldots, d_m), d_l := \operatorname{dct} \mathbf{P}_l - 1$, we can symbolize the procedure as follows

$$(\mathbf{P}, \mathbf{d}) \leftarrow \text{FPHPS } (\mathbf{F}, \sigma, \mathbf{n}).$$

Note that, in general, the choice of π_σ and therefore the output of FPHPS is not unique, but uniqueness could be easily obtained, for instance, by the additional restriction that π_σ must be as small as possible.

The basic step of a divide-and-conquer version is described in Theorem 6.1.

THEOREM 6.1. *Let ρ, σ be integers with $0 \le \rho \le \sigma$. Suppose that we have iterated $\rho \le \sigma$ times the recursive step of* FPHPS

$$(\mathbf{P}^{(1)}, \mathbf{d}^{(1)}) \leftarrow \text{FPHPS } (\mathbf{F}, \rho, \mathbf{n}),$$

and then continue iterating

$$(\mathbf{P}^{(3)}, \mathbf{d}^{(3)}) \leftarrow \text{FPHPS } (\mathbf{F}, \sigma, \mathbf{n}).$$

Suppose further that we restart the procedure with new initializations

$$\left(\mathbf{P}^{(2)}, \mathbf{d}^{(2)}\right) \leftarrow \text{FPHPS } \left(\mathbf{F}^{(1)}, \sigma - \rho, \mathbf{d}^{(1)}\right), \quad \textit{where } \mathbf{F}^{(1)}(z) := z^{-\rho} \cdot \mathbf{P}^{(1)}(z^s) \cdot \mathbf{F}(z),$$

where we always use the above uniqueness condition for the values π_σ. Then

$$\mathbf{P}^{(3)} = \mathbf{P}^{(2)} \cdot \mathbf{P}^{(1)} \quad \textit{and} \quad \mathbf{d}^{(3)} = \mathbf{d}^{(2)}.$$

Proof. We show Theorem 6.1 by induction on $(\sigma - \rho)$. Extending our notation slightly, set

$$\left(\mathbf{P}_{\rho}^{(1)}, \mathbf{d}_{\rho}^{(1)}\right) = \left(\mathbf{P}^{(1)}, \mathbf{d}^{(1)}\right), \qquad \left(\mathbf{P}_{\sigma}^{(3)}, \mathbf{d}_{\sigma}^{(3)}\right) = \left(\mathbf{P}^{(3)}, \mathbf{d}^{(3)}\right),$$
$$\left(\mathbf{P}_{\sigma-\rho}^{(2)}, \mathbf{d}_{\sigma-\rho}^{(2)}\right) = \left(\mathbf{P}^{(2)}, \mathbf{d}^{(2)}\right).$$

Note that Theorem 6.1 is trivially true for $\sigma - \rho = 0$. Assume now that the result is true for $\sigma - \rho \geq 0$. Then

$$\mathbf{P}_{\sigma}^{(3)} = \mathbf{P}_{\sigma-\rho}^{(2)} \cdot \mathbf{P}_{\rho}^{(1)} \quad \text{and} \quad \mathbf{d}_{\sigma}^{(3)} = \mathbf{d}_{\sigma-\rho}^{(2)}. \tag{15}$$

Consequently, the corresponding s-residuals

$$\mathbf{R}_{\sigma}^{(3)}(z) = z^{-\sigma} \cdot \mathbf{P}_{\sigma}^{(3)}(z^s) \cdot \mathbf{F}(z) \quad \text{and} \quad \mathbf{R}_{\sigma-\rho}^{(2)}(z) = z^{-\sigma+\rho} \cdot \mathbf{P}_{\sigma-\rho}^{(2)}(z^s) \cdot \mathbf{F}^{(1)}(z)$$

are equal. Hence, in both cases we must take the same value π and the assertion (15) with σ replaced by $(\sigma + 1)$ follows. □

The basic step of a divide-and-conquer version (15) yields the *superfast power Hermite Padé solver* (SPHPS), a reliable algorithm for computing a σ-basis of PHPAs with complexity $\mathcal{O}(\sigma \cdot \log^2 \sigma)$. The reason for the improvement in complexity results from the use of fast Fourier transform (FFT) techniques for fast polynomial multiplication. Such techniques consist of converting to a new coordinate representation via polynomial evaluation at roots of unity, computing the arithmetic operations in these new coordinates and transfering the results back to the original computation domain via polynomial interpolation. For purposes of efficiency we describe our superfast algorithm in both coordinate representations. Hence, we require some FFT details needed for our implementation. Additional details of the FFT procedure can be found in many texts (cf. [15]).

Let ω_κ be the principal κth root of unity (e.g., if $\mathbb{K}$ is the complex numbers, then $\omega_\kappa := \cos(2\pi/\kappa) + i \cdot \sin(2\pi/\kappa)$) and let

$$(\xi_j)_{j=0,\ldots,2\kappa-1} \leftarrow \mathrm{DFT}_{2\kappa}(p(z))$$

denote the evaluation of $\xi_j := p(\omega_{2\kappa}^j), j = 0, \ldots, 2\kappa - 1$. Then for the classical discrete FFT algorithm, we split p into its even and odd part $p(z) = p_e(z^2) + z \cdot p_o(z^2)$ and use the fact that for $j = 0, \ldots, \kappa - 1$ we have

$$\xi_j = \xi_j^{(e)} + \omega_{2\kappa}^j \cdot \xi_j^{(o)} \quad \text{and} \quad \xi_{\kappa+j} = \xi_j^{(e)} - \omega_{2\kappa}^j \cdot \xi_j^{(o)},$$

where $(\xi_j^{(e)})_{j=0,\ldots,\kappa-1} \leftarrow \mathrm{DFT}_\kappa(p_e(z))$ and $(\xi_j^{(o)})_{j=0,\ldots,\kappa-1} \leftarrow \mathrm{DFT}_\kappa(p_o(z))$. The "inverse" polynomial interpolation computation of

$$p(z) \leftarrow \mathrm{IDFT}_\kappa\left((\xi_j)_{j=0,\ldots,\kappa-1}\right),$$

i.e. of the uniquely defined polynomial p of degree less than κ with $\xi_j := p(\omega_\kappa^j), j = 0, \ldots, \kappa - 1$, is done by

$$\hat{p}(z) := \sum_{j=0}^{\kappa-1} \xi_j z^{n-j}, (\hat{\xi}_j)_{j=0,\ldots,\kappa-1} \leftarrow \mathrm{DFT}_\kappa(\hat{p}(z)), \qquad \text{then } p(z) = \frac{1}{\kappa} \cdot \sum_{j=0}^{\kappa-1} \hat{\xi}_j z^j.$$

Polynomials are multiplied componentwise in the new coordinates, that is, if

$$\left(\xi_j^{(1)}\right)_{j=0,\ldots,\kappa-1} \leftarrow \mathrm{DFT}_\kappa\left(p_1(z)\right),$$
$$\left(\xi_j^{(2)}\right)_{j=0,\ldots,\kappa-1} \leftarrow \mathrm{DFT}_\kappa\left(p_2(z)\right), \quad \text{and}$$
$$p^{(3)}(z) \leftarrow \mathrm{IDFT}_\kappa\left(\left(\xi_j^{(1)} \cdot \xi_j^{(2)}\right)_{j=0,\ldots,\kappa-1}\right),$$

and if c denotes the leading coefficient of $p_1(z) \cdot p_2(z)$, then

$$\begin{aligned} p^{(3)}(z) &= p^{(1)}(z) \cdot p^{(2)}(z) \bmod z^\kappa, \\ &= p^{(1)}(z) \cdot p^{(2)}(z) + \begin{cases} 0 & \text{if } \deg\left(p^{(1)} \cdot p^{(2)}\right) < \kappa, \\ -c \cdot z^\kappa + c & \text{if } \deg\left(p^{(1)} \cdot p^{(2)}\right) = \kappa. \end{cases} \end{aligned}$$

For κ a power of two, the complexity of converting to the new coordinate representation and back again (via either DFT_κ or IDFT_κ) is at most $\frac{1}{2} \cdot \kappa \cdot \log \kappa + \mathcal{O}(\kappa)$ multiplications and $\kappa \cdot \log \kappa + \mathcal{O}(\kappa)$ additions (the logarithm taken with respect to the basis 2). Therefore, the polynomial multiplication is of complexity $\mathcal{O}(\kappa \cdot \log \kappa)$.

In the SPHPS algorithm, we use the notations $\mathbf{e}_l$ is the lth unit vector, $\mathbf{I}$ the unit matrix of size $(m \times m)$, and $\{\sum_{j=0}^{\infty} c_j z^j\}_\kappa := \sum_{j=0}^{\kappa-1} c_j z^j$ denotes a truncated power series.

SPHPS Algorithm $(\mathbf{F}, \sigma, \kappa, \mathbf{n})$
Input: $\sigma, \kappa \in \mathbb{N}_0$, with $\sigma \le \kappa = 2^k$ for a $k \in \mathbb{N}_0$,
$\mathbf{n} = (n_1, \ldots, n_m)$, vector of integers,
$\mathbf{F} = (f_1, \ldots, f_m)^T$ vector of truncated power series,
i.e., of polynomials of degree less than κ,
Let $\mathbf{G} \in \mathbb{K}^{m \times s}[z]$ be defined by $\mathbf{F}(z) = \mathbf{G}(z^s) \cdot (1, z, \ldots, z^{s-1})^T$

Output: $\mathbf{P}, \xi$, and $\mathbf{d}$ where:
$\mathbf{d} = (d_1, \ldots, d_m)$, vector of integers,
$\mathbf{P} = (P_{l,\lambda})_{l=1,\ldots,m}^{\lambda=1,\ldots,m}$, consisting of rows
$\mathbf{P}_l = (P_{l,1}, \ldots, P_{l,m})$ with $\mathrm{dct}\ \mathbf{P}_l = d_l + 1$,
$\deg P_{l,l} \le \kappa$ and for $l \ne \lambda : \deg P_{l,\lambda} < \kappa$,
for all $\delta \in \mathbb{Z} : \mathcal{L}_\delta^\sigma = \{\alpha_1 \mathbf{P}_1 + \cdots + \alpha_m \mathbf{P}_m : \deg \alpha_l \le d_l + \delta\}$
$\xi = (\xi_j)_{j=0,\ldots,2\kappa-1}$, each ξ_j an m by m matrix,
$(\xi_j)_{j=0,\ldots,2\kappa-1} \longleftarrow \mathrm{DFT}_{2\kappa}(\mathbf{P}(z))$

The Recursion
Case ($\sigma = 0$ and $\kappa \ge 1$) or ($\sigma = \kappa = 1$ and $f_1(0) = \cdots = f_m(0) = 0$):
Return $(\mathbf{P}, \xi, \mathbf{d}) = (\mathbf{I}, \underbrace{(\mathbf{I}, \ldots, \mathbf{I})}_{2\kappa}, \mathbf{n})$

Case $\sigma = \kappa = 1, f_\pi(0) \ne 0$ and for all l with $n_l > n_\pi$: $f_l(0) = 0$:

$$P \leftarrow \begin{bmatrix} 1 & & & -f_1(0)/f_\pi(0) & & \\ & \ddots & & \vdots & & \\ & & 1 & -f_{\pi-1}(0)/f_\pi(0) & & \\ & & & z & & \\ & & & -f_{\pi+1}(0)/f_\pi(0) & 1 & \\ & & & \vdots & & \ddots \\ & & & -f_m(0)/f_\pi(0) & & 1 \end{bmatrix}$$

RETURN $(\mathbf{P}, \xi, \mathbf{d}) = (\mathbf{P}, (\mathbf{P}(1), \mathbf{P}(-1)), \mathbf{n} - \mathbf{e}_\pi)$

CASE $\sigma \geq 1$ and $\kappa > 1$: (Divide-and-conquer step)

Compute basis to order $\kappa/2$:

$\overline{\kappa} \longleftarrow \kappa/2, \overline{\sigma} \longleftarrow \min\{\sigma, \overline{\kappa}\}; \mathbf{F}^{(1)}(z) \longleftarrow \{\mathbf{F}(z)\}_{\overline{\kappa}}$

$(\mathbf{P}^{(1)}, \xi^{(1)}, \mathbf{d}^{(1)}) \longleftarrow$ SPHPS $(\mathbf{F}^{(1)}, \overline{\sigma}, \overline{\kappa}, \mathbf{n})$

Compute basis to order κ:

$(\eta_j)_{j=0,\ldots,\kappa-1} \longleftarrow \mathrm{DFT}_\kappa(\mathbf{G}(z))$

$\mathbf{G}^{(2)}(z) \longleftarrow \mathrm{IDFT}_\kappa((\xi_j^{(1)} \cdot \eta_j)_{j=0,\ldots,\kappa-1})$

$\mathbf{F}^{(2)}(z) \longleftarrow \{z^{-\overline{\kappa}} \cdot \mathbf{G}^{(2)}(z^s) \cdot (1, z, \ldots, z^{s-1})^T\}_{\overline{\kappa}}$

$(\mathbf{P}^{(2)}, \xi^{(2)}, \mathbf{d}^{(2)}) \longleftarrow$ SPHPS $(\mathbf{F}^{(2)}, \sigma - \overline{\sigma}, \overline{\kappa}, \mathbf{d}^{(1)})$

Combine both parts:

$\xi_{2j}^{(3)} \longleftarrow \xi_j^{(2)} \cdot \xi_j^{(1)}$ for $j = 0, 1, \ldots, \kappa - 1$

$\mathbf{P}^{(3)}(z) \longleftarrow \mathrm{IDFT}_\kappa((\xi_{2j}^{(3)})_{j=0,\ldots,\kappa-1})$

If $\deg P_{l,l}^{(1)} = \deg P_{l,l}^{(2)} = \overline{\kappa}$, then $P_{l,l}^{(3)}(z) \longleftarrow P_{l,l}^{(3)}(z) - 1 + z^\kappa$

$(\xi_{2j+1}^{(3)})_{j=0,1,\ldots,\kappa-1} \longleftarrow \mathrm{DFT}_\kappa(\mathbf{P}^{(3)}(\omega_{2\kappa} \cdot z))$

RETURN $(\mathbf{P}, \xi, \mathbf{d}) = (\mathbf{P}^{(3)}, \xi^{(3)}, \mathbf{d}^{(2)})$

Consider now the problem of determining the complexity of the SPHPS algorithm. For simplicity, we still impose the conditions before Theorem 4.1 (otherwise, the complexity will be still smaller). As in §2, in most applications one must determine σ-bases of PHPAs for $\sigma \approx \|\mathbf{n}\|$.

THEOREM 6.2 (Complexity). *The* SPHPS *algorithm for computing* PHPAs *of order σ has a complexity of at most*

$$\frac{3}{2} \cdot (m+s) \cdot m \cdot \sigma \cdot \log^2 \sigma + \mathcal{O}(\sigma \cdot \log \sigma)\,\mathrm{AO}, \tag{16}$$

roughly half multiplications as additions.

Proof. Let $\Phi_A(\kappa)$ and $\Phi_M(\kappa)$ denote the number of additions/subtractions and multiplications/divisions required for the SPHPS algorithm with parameter κ, respectively. We easily obtain $\Phi_A(1) \leq 1$ and $\Phi_M(1) \leq m - 1$. Moreover, in the last case we call the subroutines DFT_κ or IDFT_κ at most $2(m+s)m$ times; hence,

$$\Phi_A(\kappa) \leq 2 \cdot \Phi_A(\kappa/2) + 2 \cdot (m+s) \cdot m \cdot \kappa \cdot \log \kappa + \mathcal{O}(\kappa)$$

and

$$\Phi_M(\kappa) \leq 2 \cdot \Phi_M(\kappa/2) + (m+s) \cdot m \cdot \kappa \cdot \log \kappa + \mathcal{O}(\kappa).$$

This gives the complexity result. □

We remark that, as was the case with method FPHPS, the complexity will be even less for some special cases. For example, for simultaneous Padé approximation, this number will be smaller if one carefully checks whether some entries of the matrix $\mathbf{G}$ always equal zero or 1.

7. Conclusions. In this paper we have studied the concept of a power Hermite Padé approximant. These approximants are shown to generalize a number of Padé approximation problems, including, for example, the classical Hermite Padé and simultaneous Padé approximation problems as well as matrix-type generalizations of common Padé approximation problems. A fast (and also a superfast), reliable algorithm to compute these approximants is given. In this way our work provides a

uniform method of both describing and computing a wide variety of Padé and matrix-Padé approximation problems. As an immediate application, our work results in new and faster algorithms for a number of problems that rely on matrix-type Padé computation. For example, our algorithms, used in conjunction with the results of [17], gives faster algorithms for the inversion of striped or layered block Hankel (or Toeplitz) matrices. Similarly, the same algorithms combined with the results of [18] give similar improvements for the inversion of rectangular-block Hankel (or Toeplitz) matrices.

There are a number of directions for new research in this area. Our algorithm follows an m-dimensional "diagonal" path. In special cases, however, fast, reliable algorithms are given (cf. [4]) that can succeed on arbitrary staircase paths in m-dimensional space. The methods of [4] could also be extended to compute the more general PHPAs on arbitrary staircase paths, leading to a method with smaller complexity (cf. [26]).

Our algorithm does not consider the problem of stability when the computations are to be done with floating point numbers. Recently, Cabay and Meleshko [13] presented a (weakly) stable algorithm for the case $s = 1$ and $m = 2$. We conjecture that such an algorithm is also possible for the PHPA problem with arbitrary s and m, though not necessarily using the same approach as used in this paper. Our algorithm assumes exact arithmetic and has been implemented in the Maple computer algebra system. However, it does not consider the problem of exponential growth of the coefficients resulting in our computations. It would be interesting to extend our algorithm to this case. This would be done by restricting $\mathbb{K}$ to be an integral domain rather than a field and perhaps using fraction-free methods similar to those used for solving polynomial greatest common divisor (gcd) problems (cf. [15]).

Finally, the concept of a PHPA is a scalar generalization of a Hermite Padé approximant used to solve matrix-like Padé approximation problems. For example, as shown in [5] this concept also allows for a description of the structures in a singular PHPA solution table by adapting the scalar techniques of [3]. For matrix-like rational interpolation problems (with arbitrary knots), a common framework is given by the vector M-Padé approximation as a canonical extension of Example 2.5 (see [26]). In contrast, we are interested in a scalar generalization of the M-Padé approximant that can be used for simple, fast, and efficient algorithms and that, following [3], [5], might also be helpful for obtaining results about the structure of the singular matrix rational interpolation table.

REFERENCES

[1] G. A. Baker and P. R. Graves-Morris, *Padé Approximants, Part* II, Addison-Wesley, Reading, MA, 1981.

[2] B. Beckermann, *Zur Interpolation mit polynomialen Linearkombinationen beliebiger Funktionen*, Ph.D. thesis, Dept. of Mathematics, University of Hannover, Germany, 1990.

[3] ———, *The structure of the singular solution table of the M-Padé approximation problem*, J. Comput. Appl. Math., 32 (1990), pp. 3–15.

[4] ———, *A reliable method for computing M-Padé approximants on arbitrary staircases*, J. Comput. Appl. Math., 40 (1992), pp. 19–42.

[5] B. Beckermann and G. Labahn, *A uniform approach for Hermite Padé and simultaneous Padé approximants and their matrix generalizations*, Numerical Algorithms, 3 (1992), pp. 45–54.

[6] M. G. de Bruin, *The interruption phenomenon for generalized continued fractions*, Bull. Austral. Math. Soc., 19 (1978), pp. 245–272.

[7] M. G. DE BRUIN, *Some aspects of simultaneous rational approximation*, in Numerical Analysis and Mathematical Modelling, Banach center publications, Vol 24, PWN-Polish Scientific Publishers, Warsaw, 1990, pp. 51–84.

[8] M. G. DE BRUIN, *Simultaneous partial Padé approximants*, J. Comput. Appl. Math., 21 (1988), pp. 343–355.

[9] R. BRENT, F. G. GUSTAVSON, AND D. Y. Y. YUN, *Fast solution of Toeplitz systems of equations and computation of Padé approximants*, J. Algorithms, 1 (1980), pp. 259–295.

[10] S. CABAY AND D. K. CHOI, *Algebraic computations of scaled Padé fractions*, SIAM J. Comput., 15 (1986), pp. 243–270.

[11] S. CABAY, G. LABAHN, AND B. BECKERMANN, *On the theory and computation of non-perfect Padé-Hermite approximants*, J. Comput. Appl. Math., 39 (1992), pp. 295–313.

[12] S. CABAY AND G. LABAHN, *A superfast algorithm for multidimensional Padé systems*, Numerical Algorithms, 2 (1992), pp. 201–224.

[13] S. CABAY AND R. MELESHKO, *A weakly stable algorithm for the Padé approximants and the inversion of Hankel matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 735–765.

[14] J. COATES, *On the algebraic approximation of functions*, Indag. Math., 28 (1966), pp. 421–461.

[15] K. O. GEDDES, S. R. CZAPOR, AND G. LABAHN, *Algorithms for Computer Algebra*, Kluwer, Boston, MA, 1992.

[16] H. JAGER, *A multidimensional generalization of the Padé table*, Indag. Math., 26 (1964), pp. 193–249.

[17] G. LABAHN, *Inversion components of block Hankel-like matrices*, Linear Algebra Appl., 177 (1992), pp. 7–48.

[18] ———, *Inversion Algorithms for Rectangular-block Hankel Matrices*, Research report CS-90-52, University of Waterloo, 1990.

[19] G. LABAHN AND S. CABAY, *Matrix Padé fractions and their computation*, SIAM J. Comput., 18 (1989), pp. 639–657.

[20] W. LÜBBE, *Über ein allgemeines Interpolationsproblem—Lineare Identitäten zwischen benachbarten Lösungssystemen*, Ph.D. thesis, University of Hannover, Germany 1983.

[21] K. MAHLER, *Perfect systems*, Compos. Math., 19 (1968), pp. 95–166.

[22] S. PASZKOWSKI, *Hermite Padé approximation: Basic notions and theorems*, J. Comput. Appl. Math., 32 (1990), pp. 229–236.

[23] R. E. SHAFER, *On quadratic approximation*, SIAM J. Numer. Anal., 11 (1974), pp. 447–460.

[24] M. VAN BAREL AND A. BULTHEEL, *A new approach to the rational interpolation problem*, J. Comput. Appl. Math., 32 (1990), pp. 281–289.

[25] ———, *The computation of nonperfect Padé–Hermite approximants*, Numerical Algorithms, 1 (1991), pp. 285–304.

[26] ———, *A general module theoretic framework for vector M-Padé and matrix rational interpolation*, Numerical Algorithms, 3 (1992), pp. 451–462.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 824-844, July 1994

008

A BLOCK-PARALLEL NEWTON METHOD VIA OVERLAPPING EPSILON DECOMPOSITIONS *

A. I. ZEČEVIĆ† AND D. D. ŠILJAK‡

Abstract. The purpose of this paper is to present a block-parallel Newton method for solving large nonlinear systems. A graph-theoretic decomposition algorithm is first used to partition the Jacobian into weakly coupled, possibly overlapping blocks. It is then shown that it suffices to invert only the diagonal blocks to carry out the Newton iterates. A rigorous justification of this practice is provided by using a convergence result of Kantorovich in the expanded space of the iterates, where overlapping blocks appear as disjoint. The individual blocks, or a group of blocks, can be inverted by a dedicated processor, making the new block-diagonal Newton method ideally suited for parallel processing. Applications to the power flow problem are presented and parallelization issues are discussed.

Key words. nonlinear equations, block-iterative solutions, weak coupling, overlapping decompositions, bigraphs, power systems, load-flow problem

AMS subject classification. 65

1. Introduction.

Due to steadily increasing demands for speedups and reliability in solving large systems of nonlinear equations, there has been a concerted effort to develop new methods for parallel computation via multiprocessor architectures. A common obstacle in this context has been excessive requirements for communication between the processors that can slow down the convergence of the solution process, if not destroy it altogether. For this reason, a good deal of research has been devoted to formulations of effective partitioning algorithms for mapping of large problems onto multiprocessor architectures, which result in minimal communication requirements [8].

The objective of this paper is to propose a block-parallel Newton method for solving large systems of nonlinear equations. The principal part of the method is the partitioning algorithm based on overlapping epsilon decompositions [21] with which we decompose the corresponding Jacobian matrix into weakly coupled diagonal blocks. By assigning a block per processor and ignoring the coupling between the blocks, we can significantly reduce the communication between the processors during the solution process. Most importantly, we extend the results of Kantorovich [12] to rigorously establish the convergence of the corresponding block-Newton iterates containing overlapping blocks.

The proposed block-parallel method is ideally suited for solving the load-flow problem in electric power systems, e.g., [25]. A whole range of epsilon decompositions is available that allows for tradeoffs in balancing the size of the blocks and the strength of coupling. This translates into a desired balance between the load across processors and the amount of communication between processors. Furthermore, the graph-theoretic algorithm underlying the decompositions is linear in complexity, thus making the proposed solution procedure more attractive with increasing size of the system.

---

*Received by the editors April 9, 1992; accepted for publication (in revised form) March 4, 1993. This research was supported by National Science Foundation grant ECS-9114872.

†School of Engineering, Santa Clara University, Santa Clara, California 95053 (azecevic@scu.bitnet)

‡B & M Swig Professor, Santa Clara University, Santa Clara, California 95053 (dsiljak@scu.bitnet)

Finally, since epsilon decompositions are inherently nested, there is a real potential in building hierarchical multiprocessor-multirate schemes with rate and granularity at each level being determined by the level of coupling between the corresponding parts of the system.

2. Epsilon decomposition. Our objective is to compute a solution x^* of the system

$$\mathcal{S} : f(x) = 0 \tag{2.1}$$

using the simplified Newton method

$$\mathcal{N} : x_{k+1} = x_k - A(x_0)^{-1} f(x_k), \qquad k = 0, 1, 2, \ldots, \tag{2.2}$$

where $A(x_0) = f'(x_0)$ is the Jacobian of nonlinear mapping $f : \Omega \subset \mathbf{R}^n \to \mathbf{R}^n$, which is computed at initial point x_0. A nonstandard feature of our solution procedure is a partitioning of Jacobian $A(x_0)$ into weakly coupled blocks, so that the inversion $A(x_0)^{-1}$ can be reduced to an inversion of the diagonal blocks of $A(x_0)$. A justification of the block-diagonal inversion in the context of Newton's iterative process $\mathcal{N}$ is provided in the next section. Our immediate interest, however, is to illustrate the decomposition solution procedure by a simple example.

Example (2.3). Let us consider a system

$$\begin{aligned} \mathcal{S} : f_1(x) &\equiv e^{x_1} e^{x_2/5} \cos \frac{x_3}{5} - 1.85 = 0, \\ f_2(x) &\equiv x_2 e^{x_3/3} \cos \frac{x_1}{3} - 2.9 = 0, \\ f_3(x) &\equiv e^{x_2/2} \cos \frac{1}{10}(x_3 + 1) - 4.3 = 0, \end{aligned} \tag{2.4}$$

with the Jacobian computed as

(2.5)

$$A(x) = \begin{bmatrix} e^{x_1} e^{x_2/5} \cos \frac{x_3}{5} & \frac{1}{5} e^{x_1} e^{x_2/5} \cos \frac{x_3}{5} & -\frac{1}{5} e^{x_1} e^{x_2/5} \sin \frac{x_3}{5} \\ -\frac{1}{3} x_2 e^{x_3/3} \sin \frac{x_1}{3} & e^{x_3/3} \cos \frac{x_1}{3} & \frac{1}{3} x_2 e^{x_3/3} \cos \frac{x_1}{3} \\ 0 & \frac{1}{2} e^{x_2/2} \cos \frac{1}{10}(x_3 + 1) & -\frac{1}{10} e^{x_2/2} \sin \frac{1}{10}(x_3 + 1) \end{bmatrix}.$$

Choosing initial approximation $x_0 = (0.1, 3, 0)^T$, we get

$$A(x_0) = \begin{bmatrix} 2.014 & 0.403 & 0 \\ -0.333 & 0.999 & 0.999 \\ 0 & 2.23 & -0.045 \end{bmatrix}. \tag{2.6}$$

To decompose this matrix into weakly coupled blocks, we denote by ε all nonzero elements of the matrix with absolute value less than or equal to 0.05 and by $*$ all other nonzero elements. Then, a symbolic representation of $A = A(x_0)$ is

$$A = \begin{bmatrix} * & * & 0 \\ \varepsilon & * & * \\ 0 & * & \varepsilon \end{bmatrix}. \tag{2.7}$$

Since there is a column of stars in A, no permutation of rows and columns can produce an epsilon decomposition of A; that is, a decomposition such that all off-diagonal blocks of A consist of epsilon elements only. However, matrix A has an *overlapping decomposition* [21]. By expanding the underlying space $\mathbf{R}^3$ using the transformation matrices,

$$V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \tilde{V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{2.8}$$

we get the expanded Jacobian

$$\tilde{A} = \left[\begin{array}{cc:cc} * & * & 0 & 0 \\ 0 & * & \varepsilon & 0 \\ \hdashline \varepsilon & 0 & * & * \\ 0 & 0 & \varepsilon & * \end{array}\right], \tag{2.9}$$

which is defined by

$$VA = \tilde{A}\tilde{V}. \tag{2.10}$$

Matrix $\tilde{A}$ now has an epsilon decomposition indicated by dashed lines in (2.9).

We can take advantage of the weak coupling in (2.9) and consider Newton iterates

$$\tilde{\mathcal{N}} : \tilde{x}_{k+1} = \tilde{x}_k - \tilde{A}^{-1}\tilde{f}(\tilde{x}_k), \qquad k = 0, 1, 2, \ldots \tag{2.11}$$

in the expanded space $\mathbf{R}^4$ to solve the system of equations

$$\begin{aligned} \tilde{\mathcal{S}} : \tilde{f}_1(\tilde{x}) &\equiv e^{\tilde{x}_1} e^{\tilde{x}_2/5} \cos \frac{\tilde{x}_3}{5} - 1.85 = 0, \\ \tilde{f}_2(\tilde{x}) &\equiv e^{\tilde{x}_2/2} \cos \frac{1}{10}(\tilde{x}_3 + 1) - 4.3 = 0, \\ \tilde{f}_3(\tilde{x}) &\equiv \tilde{x}_4 e^{\tilde{x}_3/3} \cos \frac{\tilde{x}_1}{3} - 2.9 = 0, \\ \tilde{f}_4(\tilde{x}) &\equiv e^{\tilde{x}_4/2} \cos \frac{1}{10}(\tilde{x}_3 + 1) - 4.3 = 0 \end{aligned} \tag{2.12}$$

with initial approximation $\tilde{x}_0 = \tilde{V}x_0 = (0.1, 3, 0, 3)^T$. A description of the method used to construct such an expanded function $\tilde{f}$ is presented in §4.

The matrix $\tilde{A}(\tilde{x}_0)$ is Jacobian $\tilde{f}'(\tilde{x}_0)$ of mapping $\tilde{f}(\tilde{x})$, which is defined by $\tilde{f}(\tilde{V}x) = Vf(x)$. The significance of expansion is that Jacobian $\tilde{A}$ has a weak coupling structure indicated in (2.9), which provides an epsilon decomposition

$$\tilde{A} = \tilde{A}_D + \varepsilon \tilde{A}_C, \tag{2.13}$$

where

$$\tilde{A}_D = \left[\begin{array}{cc:cc} 2.014 & 0.403 & & \\ 0 & 2.23 & \multicolumn{2}{c}{\odot} \\ \hdashline & & 0.999 & 0.999 \\ \multicolumn{2}{c:}{\odot} & -0.045 & 2.23 \end{array}\right],$$

(2.14)

$$\tilde{A}_C = \left[\begin{array}{cc:cc} & & 0 & 0 \\ \multicolumn{2}{c:}{\odot} & -0.9 & 0 \\ \hdashline -0.66 & 0 & \multicolumn{2}{c}{\odot} \\ 0 & 0 & & \end{array}\right].$$

A crucial step in the solution process is to take advantage of the weak coupling term $\varepsilon\tilde{A}_c$ and replace $\tilde{A}^{-1}$ in Newton iterates (2.11) by $\tilde{A}_D^{-1}$ computed as

$$\tilde{A}_D^{-1} = \left[\begin{array}{cc:cc} 0.4965 & -0.09 & & \\ 0 & 0.4485 & \multicolumn{2}{c}{\odot} \\ \hdashline & & 0.981 & -0.4395 \\ \multicolumn{2}{c:}{\odot} & 0.0198 & 0.4395 \end{array}\right], \tag{2.15}$$

which amounts to inverting only the diagonal 2×2 blocks of $\tilde{A}$. This fact allows for the construction of parallel schemes that can provide significant computational speedups in large systems.

After several iterations, the *block-diagonal Newton iterates*

$$\tilde{\mathcal{N}}_D : \tilde{x}_{k+1} = \tilde{x}_k - \tilde{A}_D^{-1}\tilde{f}(\tilde{x}_k), \qquad k = 0, 1, 2, \ldots \tag{2.16}$$

produce a solution of $\tilde{\mathcal{S}}$

$$\tilde{x}^* = (0.029859,\ 2.926705,\ -0.027351,\ 2.926705)^T. \tag{2.17}$$

A straightforward elimination of the repeated components in $\tilde{x}^*$ using transformation

$$\tilde{x}^* = \tilde{V}x^* \tag{2.18}$$

provides the solution

$$x^* = (0.029859,\ 2.926705,\ -0.027351)^T \tag{2.19}$$

of the original system $\mathcal{S}$.

Remark (2.20). Although in Example (2.3) the overlapping epsilon decomposition was necessary, most of the time a disjoint decomposition would suffice [20], [23]. Disjoint decompositions result in less computation and they are easier to obtain. As in Example (2.3), however, a disjoint decomposition may not exist for a given epsilon and more general overlapping decompositions are required.

3. The main result. We now provide a justification of the procedure proposed in Example (2.3). The full generality of our development is based on a result of Kantorovich, which was established for a weak perturbation of the Jacobian matrix. What makes our result interesting is that it concerns weak perturbations of a *block-diagonal* Jacobian and that the blocks are *overlapping*. The latter feature requires use of the inclusion principle [23] in the context of Newton iterates, as well as an extension of Kantorovich's result to the expansion-contraction process involving the underlying space of the iterates.

We consider the system of nonlinear equations

$$\mathcal{S} : f(x) = 0, \tag{3.1}$$

where $f : \Omega \subset \mathbf{R}^n \to \mathbf{R}^n$ is a twice differentiable mapping on a domain Ω, and x_0 is an initial approximation of the solution x^* of $\mathcal{S}$. Our crucial assumption is that there exist two $\tilde{n} \times n$ matrices V and $\tilde{V}$ with full column rank and $\tilde{n} \geq n$, such that Jacobian $A = f'(x_0)$, satisfying the condition

$$VA = \tilde{A}\tilde{V} \tag{3.2}$$

has an epsilon decomposition in the expanded space,

$$\tilde{A} = \tilde{A}_D + \varepsilon \tilde{A}_C, \tag{3.3}$$

where $\varepsilon > 0$ is a sufficiently small number, and $\tilde{A}_D$ is a block-diagonal matrix

$$\tilde{A}_D = \operatorname{diag}\left\{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_N\right\} \tag{3.4}$$

with invertible blocks.

The expanded matrix $\tilde{A}(\tilde{x}_0)$ is the Jacobian $\tilde{f}'(\tilde{x}_0)$ of the mapping $\tilde{f} : \tilde{\Omega} \subset \mathbf{R}^{\tilde{n}} \to \mathbf{R}^{\tilde{n}}$ defined by

$$V f(x) = \tilde{f}\left(\tilde{V}x\right) \quad \forall x \in \mathbf{R}^n. \tag{3.5}$$

How epsilon decompositions (3.3) of $\tilde{A}$ are generated from a matrix A by choosing different values for ε and how each decomposition produces (automatically) the expanded mapping $\tilde{f}(\tilde{x})$ is explained in the next section. Here we want to establish rigorously that when we use linear transformation

$$\tilde{x} = \tilde{V}x, \tag{3.6}$$

which is obtained as a by-product of epsilon decomposition, and get the expanded system

$$\tilde{\mathcal{S}} : \tilde{f}(\tilde{x}) = 0, \tag{3.7}$$

we can then solve $\tilde{\mathcal{S}}$ instead of $\mathcal{S}$, as illustrated by Example (2.3) in §2.

We start by applying the inclusion principle to iterative processes $\mathcal{N}$ and $\tilde{\mathcal{N}}$ of (2.2) and (2.11), which are discrete dynamic systems that generate sequences $x(k; x_0)$ and $\tilde{x}(k; \tilde{x}_0)$ starting at x_0 and $\tilde{x}_0 = \tilde{V}x_0$.

DEFINITION (3.8). *An iterative process $\tilde{\mathcal{N}}$ is said to include process $\mathcal{N}$ if there exists an $\tilde{n} \times n$ matrix $\tilde{V}$ with $\tilde{n} \geq n$ and full column rank, such that*

$$\tilde{x}\left(k; \tilde{V}x_0\right) = \tilde{V}x\left(k; x_0\right) \quad \forall x_0 \in \mathbf{R}^n. \tag{3.9}$$

The reason for this definition is obvious: if sequence $\tilde{x}(k; \tilde{x}_0)$ converges to a solution $\tilde{x}^*$ of $\tilde{\mathcal{S}}$, then the corresponding solution x^* of $\mathcal{S}$ can be extracted from $\tilde{x}^*$ by using matrix $\tilde{V}$. We recall [23] that when $\tilde{\mathcal{N}}$ includes $\mathcal{N}$, we say that $\tilde{\mathcal{N}}$ is an expansion of $\mathcal{N}$, or that $\mathcal{N}$ is a contraction of $\tilde{\mathcal{N}}$.

To provide conditions for inclusion (3.9), we prove the following.

LEMMA (3.10). *Let us assume that $\tilde{\mathcal{N}}$ is obtained by expanding $A(\underline{x}_0)$ at a fixed $\underline{x}_0$, and that* (3.2) *and* (3.5) *hold. Then, $\tilde{\mathcal{N}}$ includes $\mathcal{N}$.*

Proof. Let x_0 be an arbitrary initial approximation for $\mathcal{N}$ and consider $\tilde{\mathcal{N}}$ with $\tilde{x}_0 = \tilde{V}x_0$. We note that (3.2) implies

$$\tilde{A}^{-1}V = \tilde{V}A^{-1} \tag{3.11}$$

and the proof follows by induction. Denoting $x(k) = x(k; x_0)$ and $\tilde{x}(k) = \tilde{x}(k; \tilde{V}x_0)$, and assuming $\tilde{x}(k) = \tilde{V}x(k)$, we conclude, using (3.5) and (3.11), that

$$\begin{aligned} \tilde{x}\left(k+1\right) &= \tilde{x}\left(k\right) - \tilde{A}^{-1}\tilde{f}\left[\tilde{x}\left(k\right)\right], \\ &= \tilde{V}x\left(k\right) - \tilde{A}^{-1}Vf\left[x\left(k\right)\right], \\ &= \tilde{V}\left\{x\left(k\right) - A^{-1}f\left[x\left(k\right)\right]\right\}, \\ &= \tilde{V}x\left(k+1\right) \quad \forall k > 0. \end{aligned} \tag{3.12}$$

□

Remark (3.13). It is interesting to note that inclusion (3.9) holds for all x_0 and $\tilde{x}_0 = \tilde{V}x_0$, even though Jacobian $A(x) = f'(x)$ is computed at a fixed $\underline{x}_0 \neq x_0$, as long as $\tilde{A}(\tilde{V}\underline{x}_0)$ is used in $\tilde{\mathcal{N}}$. This is clear from (3.12). We should note, however, that even though inclusion (3.9) holds for all $x_0 \in \mathbf{R}^n$, process $\tilde{\mathcal{N}}$, and thus $\mathcal{N}$, may not converge.

To establish convergence of $\tilde{\mathcal{N}}_D$, when we use $\tilde{A}_D$ instead of $\tilde{A}$, we first define a region

$$\tilde{\Omega}_0 = \{\tilde{x} \in \mathbf{R}^{\tilde{n}} : \|\tilde{x} - \tilde{x}_0\| \leq \rho\} \tag{3.14}$$

for some $\rho > 0$. Here, and throughout the paper, we use l_∞ norm $\|x\|_\infty = \max_{i \in \mathbf{N}} \{|x_i|\}$ in $\mathbf{R}^n$, and the corresponding operator norm $L(\mathbf{R}^n), \|A\|_\infty = \max_{i \in \mathbf{N}} \{\sum_{j=1}^{n} |a_{ij}|\}$, where $\mathbf{N} = \{1, 2, \ldots, n\}$. We state our main result.

THEOREM (3.15). *Let us assume that mapping $f(x)$ and its expansion $\tilde{f}(\tilde{x})$ satisfy* (3.2) *through* (3.5) *and that the following bounds hold:*

$$\left\|\tilde{A}_D^{-1}\tilde{f}\left(\tilde{x}_0\right)\right\| \leq \alpha, \tag{3.16}$$

$$\left\|\tilde{A}_D^{-1}\tilde{A}_C\right\| \leq \beta, \tag{3.17}$$

$$\left\|\tilde{A}_D^{-1}\tilde{f}''\left(\tilde{x}\right)\right\| \leq \gamma, \tag{3.18}$$

for some positive numbers α, β, γ, and for all $\tilde{x} \in \tilde{\Omega}_0$. Furthermore, we assume that

$$\varepsilon\beta < 1, \tag{3.19}$$

$$\delta = \frac{\alpha\gamma}{(1-\varepsilon\beta)^2} < \frac{1}{2}, \tag{3.20}$$

$$\underline{\rho} \leq \rho < \bar{\rho}, \tag{3.21}$$

where

$$\underline{\rho} = \frac{1-\sqrt{1-2\delta}}{\delta}\frac{\alpha}{1-\varepsilon\beta}, \qquad \bar{\rho} = \frac{1+\sqrt{1-2\delta}}{\delta}\frac{\alpha}{1-\varepsilon\beta}. \tag{3.22}$$

Then,

(i) The system $\tilde{S}$ of (3.7) has a solution $\tilde{x}^* \in \tilde{\Omega}_0$, and block-diagonal iterative process $\tilde{\mathcal{N}}_D$ converges to this solution.

(ii) Solution $\tilde{x}^*$ is unique in $\tilde{\Omega}_0$.

(iii) Iterative process $\tilde{\mathcal{N}}_D$ converges to $\tilde{x}^*$ as

$$\|\tilde{x}^* - \tilde{x}_k\| \leq \frac{\alpha}{\delta(1-\varepsilon\beta)^2}\left[1-(1-\varepsilon\beta)\sqrt{1-2\delta}\right]^{k+1}. \tag{3.23}$$

(iv) Define a manifold

$$\tilde{M} = \left\{\tilde{x} \in \mathbf{R}^{\tilde{n}} : \tilde{x} = \tilde{V}x, x \in \mathbf{R}^n\right\}. \tag{3.24}$$

Then, $\tilde{x}^* \in \tilde{M}$, and the system $\mathcal{S}$ has a unique solution x^* in the region

$$\Omega_0 = \{x \in \mathbf{R}^n : \|x - x_0\| \leq \rho\}, \tag{3.25}$$

which is related to $\tilde{x}^*$ by

$$\tilde{V}x^* = \tilde{x}^*. \tag{3.26}$$

The proof of Theorem (3.15) is provided in the Appendix.

4. The algorithm. Efficiency of the proposed block-diagonal Newton scheme hinges on our ability to systematically generate overlapping decompositions of the Jacobian matrix $A(x_0)$ and produce the transformation matrices V and $\tilde{V}$. With these matrices at hand, we can proceed to construct the function $\tilde{f}(\tilde{x})$ and use it to solve the expanded Newton iterates $\tilde{\mathcal{N}}_D$ of (2.16). Since our main task is to describe the construction of $\tilde{f}(\tilde{x})$, the epsilon decomposition algorithm will be outlined only to the extent needed for understanding the construction of $\tilde{f}(\tilde{x})$. A detailed presentation of the algorithm is provided in [21].

Given an $n \times n$ matrix $A = (a_{ij})$ and a number $\varepsilon > 0$, we say that A has an epsilon decomposition if there exist $\tilde{n} \times n$ ($\tilde{n} \geq n$) permutation matrices V and $\tilde{V}$ having full rank, such that

$$\tilde{A} = \tilde{A}_D + \varepsilon\tilde{A}_C, \tag{4.1}$$

where

$$VA = \tilde{A}\tilde{V} \tag{4.2}$$

and $\tilde{A}_D$ is a nonsingular block diagonal matrix,

$$\tilde{A}_D = \operatorname{diag}\left\{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_N\right\}. \tag{4.3}$$

In the case of singular blocks, another decomposition with a different ε should be attempted. When $\tilde{n} > n$ and $N > 1$, (4.1) represents an overlapping decomposition of A.

To generate an overlapping decomposition of a nonsingular matrix A, we associate with A a bigraph $\mathbf{B}(\mathcal{X}, \mathcal{Y}; \mathcal{E})$ such that $|\mathcal{X}| = |\mathcal{Y}| = n$ and $(x_j, y_i) \in \mathcal{E}$ if and only if $a_{ij} \neq 0, i, j, = 1, 2, \ldots, n$. We assume that bigraph $\mathbf{B}$ has a *perfect matching* $\mathbf{M}^*$, which is consistent with generic nonsingularity of A. This is equivalent to assuming that A can be permuted into a matrix with nonzero diagonal elements, e.g., [21].

For example, with a matrix

$$A = \begin{bmatrix} \otimes & * & 0 \\ \varepsilon & * & \otimes \\ 0 & \otimes & \varepsilon \end{bmatrix}, \tag{4.4}$$

which is that of (2.7) in Example (2.3), we associate bigraph $\mathbf{B}$ of Fig. 1(a). A perfect matching, which is denoted by $\otimes$ in A, is shown in Fig. 1(a) by heavy lines. By removing the edges of $\mathbf{B}$ that correspond to epsilon elements of A, we get subgraph $\mathbf{B}^\varepsilon = (\mathcal{X}, \mathcal{Y}; \mathcal{E}^\varepsilon)$ shown in Fig. 1(b). It is $\mathbf{B}^\varepsilon$ that we decompose into two overlapping components. Before doing that, however, we note that $\mathbf{B}^\varepsilon$ has retained the perfect matching of $\mathbf{B}$. If this were not true, the decomposition algorithm would terminate prematurely. In this case ε should be decreased to $\varepsilon_1 < \varepsilon$ and the algorithm restarted with a new $\mathbf{B}^{\varepsilon_1}$, which has more edges than $\mathbf{B}^\varepsilon$. The algorithm recursively determines (component by component) a perfect matching of a given bigraph and, thus, the *generic rank* of the corresponding matrix. This is performed *simultaneously* with the partitioning of bigraph $\mathbf{B}^\varepsilon$ into overlapping components.

The basic idea of the algorithm is to rearrange and often split vertices of $\mathbf{B}^\varepsilon$ to obtain a new bigraph $\tilde{\mathbf{B}}^\varepsilon = (\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}; \tilde{\mathcal{E}}^\varepsilon)$ consisting of several (disjoint) components. These components define the blocks $\tilde{A}_i$ of the diagonal matrix $\tilde{A}_D$ in (4.3). By reconnecting the epsilon edges of the original bigraph $\mathbf{B}$, we subsequently obtain an expanded bigraph $\tilde{\mathbf{B}}$ that identifies the matrix $\tilde{A}$ of (4.1) having the desired epsilon decomposition [21]. We provide the code in C for a simplified algorithm of the epsilon decomposition, which is sufficient for our objective in this paper. A new version of the complete algorithm is in a refinement stage and will appear elsewhere.

ALGORITHM (4.5).

```
main( )
{
  int tst1, tst2;
  initialization( );
/* Set column node x1 ∈ X as the current column xc */
  A: horiz_edges( );
/* Link xc with its matching. This matching becomes the current row yc ∈ Y */
  other_edges( );
```

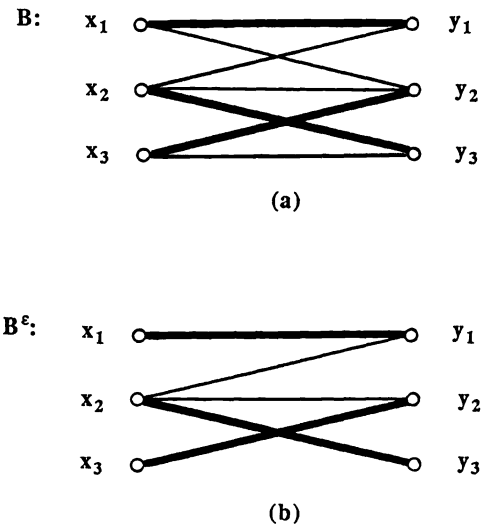


FIG. 1. *System bigraphs.*

```
/* Link y_c with all columns other than x_c that have an element > ε in row y_c. Add any
   such vertices of X to the current component if they are not there already */
     tst1 = next_x( );
/* Check if there are some unexamined x-vertices in the current component. If yes, set
   next one as x_c */
     if (tst1)
/* There exist unexamined x-vertices in current component */
        goto A;
     else{
/* All x-vertices in current component have been examined */
        tst2 = end_of_block( );
/* Check if there are any unused x-vertices left */
        if(tst2){
/* Unused x-vertices exist */
           new_block ( );
/* Initiate new block and set next unused x-vertex as x_c */
           goto A;
        }
        else
/* All x-vertices have been used—the graph is complete */
           final( );
     }
  }
```

This algorithm produces an expansion $\tilde{\mathbf{B}}$ of the original bigraph $\mathbf{B}$, which defines the expanded matrix $\tilde{A}$, as well as the transformation matrices V and $\tilde{V}$ relating $\tilde{A}$ to the original matrix A. Once the matrices V and $\tilde{V}$ are identified, the corresponding

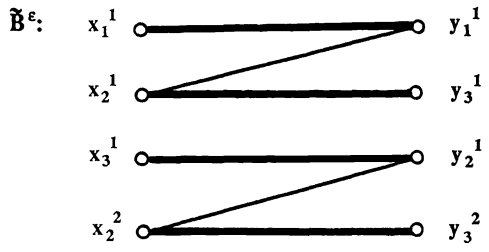


FIG. 2. *Expanded bigraph.*

function $\tilde{f}(\tilde{x})$ defined as

$$\tilde{f}'\left(\tilde{V}x_0\right) = \tilde{A}, \qquad \tilde{f}\left(\tilde{V}x\right) = Vf(x) \quad \forall x \in \mathbf{R}^n \tag{4.6}$$

is obtained by a straightforward bookkeeping procedure. The procedure is best described by reconsidering Example (2.3).

We start with the expanded bigraph $\tilde{\mathbf{B}}^{\epsilon}$ in Fig. 2, which was obtained from $\mathbf{B}^{\epsilon}$ of Fig. 1(b) by using Algorithm (4.5). In $\tilde{\mathbf{B}}^{\epsilon}$, x_j^r and y_j^r represent the rth appearance of the vertices x_j and y_j. This notation will help us track the vertex ordering in $\tilde{\mathbf{B}}^{\epsilon}$ in forming V and $\tilde{V}$.

The ordering of y-vertices in $\tilde{\mathbf{B}}^{\epsilon}$ uniquely determines the matrix V together with a correspondence relating the components $f_i : \mathbf{R}^n \to \mathbf{R}$ of $f(x)$ with the components $\tilde{f}_i : \mathbf{R}^{\tilde{n}} \to \mathbf{R}$ of the expanded function $\tilde{f}(\tilde{x})$ as follows:

$$V = \begin{array}{c} \\ y_1^1 \\ y_3^1 \\ y_2^1 \\ y_3^2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{l} \\ f_1 \to \tilde{f}_1, \\ f_3 \to \tilde{f}_2, \\ f_2 \to \tilde{f}_3, \\ f_3 \to \tilde{f}_4. \end{array} \tag{4.7}$$

The arrows indicate the defining implications and the dashed line indicates the two components induced by the overlapping decomposition.

The ordering of x-vertices in $\tilde{\mathbf{B}}^{\epsilon}$ uniquely determines matrix $\tilde{V}$ as well as the correspondence between the original n variables x_i and the $\tilde{n}$ variables $\tilde{x}_i$ of the expansion:

$$\tilde{V} = \begin{array}{c} \\ x_1^1 \\ x_2^1 \\ x_3^1 \\ x_2^2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right] \end{array} \begin{array}{l} \\ x_1 \to \tilde{x}_1, \\ x_2 \to \tilde{x}_2, \\ x_3 \to \tilde{x}_3, \\ x_2 \to \tilde{x}_4. \end{array} \tag{4.8}$$

It should be pointed out that, in general, any original variable x_i can have multiple representations x_i^r in $\mathbf{R}^{\tilde{n}}$. In (4.8), only x_2 exhibits this property with $r = 1, 2$.

Based on the established correspondences, each component $\tilde{f}_i(\tilde{x})$ is now constructed by identifying function $f_j \to \tilde{f}_i$ and then by replacing each variable $x_k, k = 1, 2, \ldots, n$, in $f_j(x_1, \ldots, x_k, \ldots, x_n)$ by one of its representations in the expanded space. We note that once the components $\tilde{f}_i$ of $\tilde{f}$ are identified with the kth block of V as in (4.7), the variables $\tilde{x}_i$ associated with the same kth block of $\tilde{V}$, as in (4.8), are the *endogenous* (subsystem) variables. They will replace their original variables in all

components $\tilde{f}_i$ comprising the kth block. On the other hand, an original variable, which is represented only by *exogenous* (interconnection) variables to the kth block, can be replaced by any one of its representations. Referring to (4.7) and (4.8), we have

Block 1:

$$\begin{aligned} &x_1 \to \tilde{x}_1 \quad \text{endogenous}, \\ &x_2 \to \tilde{x}_2 \quad \text{endogenous}, \\ &x_3 \to \tilde{x}_3 \quad \text{exogenous}, \\ &\tilde{f}_1(\tilde{x}) = f_1(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3), \\ &\tilde{f}_2(\tilde{x}) = f_3(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3). \end{aligned} \tag{4.9}$$

Block 2:

$$\begin{aligned} &x_1 \to \tilde{x}_1 \quad \text{exogenous}, \\ &x_2 \to \tilde{x}_4 \quad \text{endogenous}, \\ &x_3 \to \tilde{x}_3 \quad \text{endogenous}, \\ &\tilde{f}_3(\tilde{x}) = f_2(\tilde{x}_1, \tilde{x}_4, \tilde{x}_3), \\ &\tilde{f}_4(\tilde{x}) = f_3(\tilde{x}_1, \tilde{x}_4, \tilde{x}_3). \end{aligned} \tag{4.10}$$

It is obvious that functions (4.9) and (4.10) are those of (2.12).

5. Power systems: the load-flow problem. To determine the static operating condition of an electric power transmission system, commonly known as the load-flow problem [24], [25], one must solve a large system of nonlinear algebraic equations

$$f(x; P, Q) = 0. \tag{5.1}$$

Typically, there are several thousand equations that need to be solved for a variety of input vectors P and Q. Since it is crucial to obtain solutions as efficiently as possible, preferably on-line, parallel computations via multiprocessor architectures have become a central research topic.

Since the initial studies of parallel solutions to load-flow problems [9], it has been recognized that the key step in mapping a problem on a multiprocessor system is an efficient partitioning algorithm. Schemes for partitioning power systems have been based on a wide variety of principles, such as sparsity [7]; coherency [16], [19]; diakoptics [10], [13]; decoupling and time scales [17], [3], [15]; and overlapping subsystems [22], [11], [26], [2]. Recent studies [4], [6], [1] have indicated that the most efficient parallel configurations are those that happen to group the tightly coupled variables together. These are precisely the configurations that the overlapping epsilon decomposition [21] is designed to produce. Most importantly, the epsilon decomposition algorithm is linear in complexity; it is binary, recursive and exhaustive, and does not rely on heuristic or intuitive reasoning. That epsilon partitions are conducive to convergence of the corresponding Newton's iterates is confirmed by (3.19), (3.20), and (3.23) of our main result, Theorem (3.15), where the effect of ε is apparent.

An n-bus electric power system consists of n *nodes,* each representing either a generator or a load. If any two nodes i and k are connected by a *line,* the corresponding complex line admittance is denoted as

$$Y_{ik} = G_{ik} + jB_{ik}. \tag{5.2}$$

In addition, with each node we associate its self-admittance

$$Y_{ii} = G_{ii} + jB_{ii}, \tag{5.3}$$

where

$$G_{ii} \cong -\sum_{k \neq i} G_{ik}, \qquad B_{ii} \cong -\sum_{k \neq i} B_{ik}. \tag{5.4}$$

The load-flow problem now amounts to computing the complex voltage $E_i = V_i e^{j\theta_i}$ for all nodes, given the injected powers

$$S_i = P_i + jQ_i. \tag{5.5}$$

Since all quantities are complex, an n-bus power system generally results in a $2n \times 2n$ system of nonlinear equations

$$f_i(x_1, x_2, \ldots, x_n; P_i, Q_i) = 0, \quad i = 1, 2, \ldots, n, \tag{5.6}$$

where

$$f_i = (F_{P_i}, F_{Q_i})^T, \qquad x_i = (\theta_i, V_i)^T \tag{5.7}$$

and

$$\begin{aligned} F_{P_i} &= \bar{P}_i - V_i \sum_k \left[\bar{G}_{ik} \cos(\theta_i - \theta_k) + \bar{B}_{ik} \sin(\theta_i - \theta_k)\right] V_k, \\ F_{Q_i} &= \bar{Q}_i - V_i \sum_k \left[\bar{G}_{ik} \sin(\theta_i - \theta_k) - \bar{B}_{ik} \cos(\theta_i - \theta_k)\right] V_k, \end{aligned} \tag{5.8}$$

The overbar denotes scaling by B_{ii}.

At some nodes, the voltage magnitude V_i may be regulated at a preassigned value V_i^R, and only P_i is known. Such nodes are referred to as PV nodes. The nodes at which both P_i and Q_i are available are termed PQ nodes. With this distinction, we reformulate (5.7) as

$$f_i = \begin{cases} (F_{P_i}, F_{Q_i})^T, & i \in PQ, \\ F_{P_i}, & i \in PV. \end{cases} \tag{5.9}$$

To solve (5.6) by the Newton method, we assume the initial approximation x_0 to be

$$\theta_i^0 = 0, \qquad V_i^0 = \begin{cases} 1, & i \in PQ, \\ V_i^R & i \in PV, \end{cases} \tag{5.10}$$

which is known as "flat voltage start." In practice, this approximation is usually improved by one or two iterations on a linearization of (5.6).

To apply overlapping epsilon decompositions to the load-flow problem, we should first note that the Jacobian computed at x_0 involves both $\bar{G}_{ik}$ and $\bar{B}_{ik}$. However, since in power systems typically $\bar{B}_{ik} \gg \bar{G}_{ik}$, it suffices to perform an epsilon decomposition on the $n \times n$ matrix $\bar{B} = (\bar{B}_{ik})$, which automatically induces the corresponding decomposition of the $2n \times 2n$ Jacobian. This reduces the dimension of the decomposition problem a priori by a factor of two.

$$B = \begin{bmatrix}
-1 & .158 & .169 & .172 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.487 & -1 & .517 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.134 & .132 & -1 & .563 & 0 & .125 & 0 & .147 & 0 & 0 & 0 & 0 & 0 \\
.149 & 0 & .618 & -1 & .114 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & .229 & -1 & 0 & 0 & 0 & 0 & .236 & .183 & .352 & 0 \\
0 & 0 & .245 & 0 & 0 & -1 & .290 & .465 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & .074 & 0 & 0 & .374 & 0 & -1 & .427 & 0 & 0 & 0 & .125 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .702 & -1 & .298 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & .482 & 0 & 0 & 0 & .518 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & .585 & 0 & 0 & 0 & 0 & 0 & -1 & .415 & 0 \\
0 & 0 & 0 & 0 & .572 & 0 & 0 & 0 & 0 & 0 & .211 & -1 & .217 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .567 & 0 & 0 & 0 & .433 & -1
\end{bmatrix}$$

FIG. 3. *B matrix.*

$$B = \left[\begin{array}{cccc|cccc|cccc|cccc|cccc}
-1 & .517 & 0 & .487 & 0 & 0 & 0 & 0 & 0 & 0 & .0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.132 & -1 & .563 & .134 & 0 & .125 & .047 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & .618 & -1 & .149 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .114 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.158 & .169 & .172 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & .245 & 0 & 0 & .290 & -1 & .465 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & .074 & 0 & 0 & 0 & .374 & -1 & .427 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .125 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .702 & -1 & .298 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & .518 & 0 & .482 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .298 & -1 & .702 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & .074 & 0 & 0 & 0 & .374 & 0 & 0 & 0 & .427 & -1 & 0 & 0 & 0 & 0 & .125 & 0 & 0 & 0 \\
0 & 0 & .229 & 0 & 0 & 0 & 0 & 0 & .236 & 0 & 0 & -1 & .183 & .352 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & .415 & .585 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .211 & -1 & .572 & .217 & 0 & 0 & 0 \\
0 & 0 & .229 & 0 & 0 & 0 & 0 & 0 & .236 & 0 & 0 & 0 & .183 & .352 & -1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & .433 & 0 & .567 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .211 & 0 & 0 & .217 & -1 & .572 & 0 & 0 \\
0 & 0 & .229 & 0 & 0 & 0 & 0 & 0 & .236 & 0 & 0 & 0 & .183 & .352 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & .074 & 0 & 0 & 0 & .374 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .125 & 0 & 0 & -1 & .427 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .298 & 0 & .0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .702 & -1
\end{array}\right]$$

FIG. 4. *Expanded matrix* $\tilde{B}$.

As an illustration of the proposed solution procedure, we consider the IEEE 14-bus system [25] with matrix $\bar{B}$ shown in Fig. 3. This matrix has a variety of epsilon decompositions, one of which is presented in Fig. 4 for $\varepsilon = 0.4$. The expanded and original solutions $\tilde{x}^*$ and x^* that are obtained by the block-diagonal Newton method, are shown in Tables 1 and 2, respectively.

For a complete assessment of the proposed Newton method, we also discuss a number of parallelization issues that came up during experimentation with the IEEE

118-bus system using the Intel iPSC/860 parallel processing system. We begin by recalling that, in general, execution of the original Newton method for an $n \times n$ system requires inversion of the Jacobian that amounts to solving an $n \times n$ system of linear equations in each iteration. Although typical power systems do not require many iterations, for large n the computational effort may still be prohibitively large.

Consider now a partition of the Jacobian into N diagonal blocks of dimension n_i with $n_m = \max_i n_i$. In a parallelization based on inverting only the diagonal blocks, the execution would effectively be reduced to solving an $n_m \times n_m$ system in each iteration. Nevertheless, this advantage in itself is not enough because the number of iterations I is seen to vary significantly with different choices of partitions and termination criteria; for some choices the iterations may not converge at all. To resolve this problem, we first point out that given N processors, only *balanced* partitions (i.e., those satisfying $n_m \approx n/N$) should be considered for parallel implementation. This is due to the fact that such partitions provide adequate load balancing and at the same time require minimal execution time T_i per iteration. Our experiments have confirmed that with a fixed number of processors, time T_i is indeed approximately equal for a variety of balanced partitions. Consequently, in the class of such partitions I becomes the dominant measure for the efficiency of the iterative algorithm.

In view of the above discussion, given N processors and a preassigned termination criterion, optimal parallelization of the solution process amounts to choosing a balanced partition that provides the minimal I. Expectedly, experimental studies have indicated that the partition resulting in the optimal I is indeed an epsilon decomposition. To demonstrate this optimality, in Table 3 we present a comparison of three different partitionings that map the 118-bus system onto eight processors, with a mismatch at 0.001 per unit as the termination criterion in all cases. Partitions denoted Heuristic1 and Heuristic2 represent two common ways of partitioning power systems based on the geography and node numbering of the network, respectively. The choice of ε is induced by the preassigned number of processors and I_0 represents the number of iterations corresponding to the epsilon decomposition.

It should be pointed out that the block-parallel algorithm normally requires more iterations than the original Newton method. However, by virtue of the reduction of dimensionality, the parallel algorithm based on epsilon decompositions can result in significant computational savings. Our experiments on the IEEE 118-bus system confirmed our expectations even for this relatively small system, in which a parallel solution was seen to be up to 40% faster than when using a single processor. This is explicitly demonstrated in Table 4, where we present a comparison of computation times needed for solving the load flow equations for the IEEE 118-bus system with one, two, four, and eight processors. An improved initial approximation and the usual termination criterion (0.001 p.u. mismatch) were used.

In computations with a *single* processor, the *standard modified Newton method* of (2.2) was used and a suboptimal ordering scheme for minimizing fill in was applied in the LU factorization of the complete (unpartitioned) Jacobian. The same ordering scheme was also utilized for factorizing the individual blocks in parallel computations. When interpreting the results of Table 4 it is also important to note that such an ordering algorithm shows full efficiency only when the blocks are sufficiently large, so an upper limit should be imposed on the number of processors to be used on a given system. In particular, results presented in Table 4 indicate that the 118-bus system is

not sufficiently large to be efficiently mapped onto more than four processors (although results using eight processors are still better than those using a single processor).

6. Conclusions. We have shown how overlapping epsilon decompositions can help in solving large systems of nonlinear algebraic equations on parallel computers. By partitioning the Jacobian into overlapping weakly coupled blocks, we can take advantage of the special structural features of the system and invert only diagonal blocks of the Jacobian in a Newton interative scheme. Each block can be assigned to a dedicated processor to speed up the solution process without excessive communication among individual processors.

TABLE 1
Expanded solution $\tilde{x}^*$.

	θ	V
$\tilde{x}_0$	−.218859	.915801
$\tilde{x}_1$	−.167806	.931219
$\tilde{x}_2$	−.141123	.947302
$\tilde{x}_3$	−.077127	1.000000
$\tilde{x}_4$	−.239707	.887231
$\tilde{x}_5$	−.239707	.887231
$\tilde{x}_6$	−.280452	.866018
$\tilde{x}_7$	−.286466	.859102
$\tilde{x}_8$	−.280058	.865181
$\tilde{x}_9$	−.286498	.859108
$\tilde{x}_{10}$	−.280424	.866051
$\tilde{x}_{11}$	−.266840	.879896
$\tilde{x}_{12}$	−.288677	.861690
$\tilde{x}_{13}$	−.290361	.855968
$\tilde{x}_{14}$	−.266894	.879882
$\tilde{x}_{15}$	−.310500	.838280
$\tilde{x}_{16}$	−.290323	.855961
$\tilde{x}_{17}$	−.266849	.879887
$\tilde{x}_{18}$	−.280425	.866053
$\tilde{x}_{19}$	−.286447	.859127

TABLE 2
Original solution x^*.

	θ	V
$x_0 = \tilde{x}_3$	−.077127	1.0
$x_1 = \tilde{x}_0$	−.218859	.915801
$x_2 = \tilde{x}_1$	−.167806	.931219
$x_3 = \tilde{x}_2$	−.141123	.947302
$x_4 = \tilde{x}_{11} = \tilde{x}_{14} = \tilde{x}_{17}$	−.266861	.879888
$x_5 = \tilde{x}_5$	−.239707	.887231
$x_6 = \tilde{x}_4$	−.239707	.887231
$x_7 = \tilde{x}_6 = \tilde{x}_{10} = \tilde{x}_{18}$	−.280434	.866041
$x_8 = \tilde{x}_7 = \tilde{x}_9 = \tilde{x}_{19}$	−.286470	.859112
$x_9 = \tilde{x}_8$	−.280058	.865181
$x_{10} = \tilde{x}_{12}$	−.288677	.861690
$x_{11} = \tilde{x}_{13} = \tilde{x}_{16}$	−.290342	.855965
$x_{12} = \tilde{x}_{15}$	−.310500	.838280

The proposed method has been applied to the load-flow problem of an 118-bus power system. A pleasing fact in this context is the ability of the method to exploit sparsity of the system, thus allowing a whole host of powerful sparse techniques to be developed in the new framework. Other applications are considered in transient stability analysis of power systems [4], VLSI circuit simulations along the lines suggested in [21], as well as parallel computations in control system design [5], [20], [23].

Appendix.

Proof of Theorem (3.15). Parts (i)–(iii) follow from various results of Kantorovich [12], and will be established with a minimum of derivation. Define a mapping $\tilde{g} : \tilde{\Omega}_0 \subset \mathbf{R}^{\tilde{n}} \to \mathbf{R}^{\tilde{n}}$ as

$$\tilde{g}(\tilde{x}) = \tilde{x} - \tilde{A}_D^{-1} \tilde{f}(\tilde{x}) \tag{A.1}$$

and a scalar function

$$\varphi(t) = \frac{1}{2}\gamma t^2 + \varepsilon\beta t + \alpha. \tag{A.2}$$

(i) First, from (3.16), we conclude that

$$\|\tilde{g}(\tilde{x}_0) - \tilde{x}_0\| \leq \varphi(0). \tag{A.3}$$

Second, we set $\Delta\tilde{x} = \tilde{x} - \tilde{x}_0$ and assume $\|\Delta\tilde{x}\| \leq t, t \in [0, \rho]$. Then,

$$\begin{aligned} \|\tilde{g}'(\tilde{x})\| &= \left\| I - \tilde{A}_D^{-1} \tilde{f}'(\tilde{x}) \right\| \\ &\leq \left\| I - \tilde{A}_D^{-1} \tilde{f}'(\tilde{x}_0) \right\| + \left\| \tilde{A}_D^{-1} \left[\tilde{f}'(\tilde{x}_0) - \tilde{f}'(\tilde{x}) \right] \right\|. \end{aligned} \tag{A.4}$$

TABLE 3
Decompositions vs iterations.

Decomposition method	Block size	I/I_0
Epsilon ($\varepsilon = 0.4$)	17, 17, 17, 16, 16, 13, 12, 9	1
Heuristic1	15, 15, 15, 15, 15, 14, 14, 14	1.45
Heuristic2	15, 15, 15, 15, 15, 14, 14, 14	2.26

TABLE 4
Computation time.

Number of processors	Number of iterations	Computation time (s)	Communication time(s)	Total execution time (s)
Single processor	2	0.103	0	0.103
2	10	0.062	0.001	0.063
4	19	0.053	0.008	0.061
8	39	0.072	0.013	0.085

From (3.17) and (3.18), we can provide a bound

$$\|\tilde{g}'(\tilde{x})\| \leq \gamma t + \varepsilon\beta \equiv \varphi'(t) \tag{A.5}$$

for all $\|\tilde{x} - \tilde{x}_0\| \leq t, t \in [0, \rho]$.

Now, let us introduce a modification $\psi(t) = \varphi(t) - t$ of $\varphi(t)$,

$$\psi(t) = \frac{1}{2}\gamma t^2 - (1 - \varepsilon\beta)\, t + \alpha, \tag{A.6}$$

which has the zeros $\underline{t} = \underline{\rho}$ and $\bar{t} = \bar{\rho}$ defined in (3.22). Since $\varphi(t)$ is a nondecreasing function on $[0, \rho]$, we can show that the sequence

$$t_{k+1} = \varphi(t_k), \qquad k = 0, 1, 2, \ldots, \tag{A.7}$$

for $t_0 = 0$, converges to $\underline{t}$.

Next, by using inequalities (A.4) and (A.5), we can prove by induction on k that

$$\|\tilde{x}_{k+1} - \tilde{x}_k\| \leq t_{k+1} - t_k, \tag{A.8}$$

for all $k \geq 0$, and that $\tilde{x}_k \in \tilde{\Omega}_0$. Finally, from convergence of (A.7), compactness of $\tilde{\Omega}_0$, continuity of $\tilde{g}(\tilde{x})$, and inequality (A.8), it follows that, in fact, the sequence $\{\tilde{x}_k\}$ converges to $\tilde{x}^* \in \tilde{\Omega}_0$, which is a stationary solution of the iterative process $\tilde{\mathcal{N}}_D$, that is, $\tilde{x}^* = \tilde{g}(\tilde{x}^*)$.

(ii) To establish uniqueness of $\tilde{x}^*$ in $\tilde{\Omega}_0$, we rely on the properties of function $\varphi(t)$. We consider a sequence

$$\tau_{k+1} = \varphi(\tau_k), \qquad k = 0, 1, 2, \ldots, \tag{A.9}$$

for $\tau_0 = \rho$, and show that the sequence $\{\tau_k\}$ converges to some $\tau^* \in [\underline{t}, \rho]$. From condition (3.21) of the theorem, we have $\underline{t} \leq \rho < \bar{t}$, and $\varphi(\rho) \leq \rho$. Continuity of $\varphi(t)$ implies $\tau^* = \varphi(\tau^*)$, and because $\bar{t} > \rho$, $\underline{t}$ is the only solution of (A.9) in $[\underline{t}, \rho]$.

Now, to show uniqueness of $\tilde{x}^*$, we take an arbitrary initial element $\tilde{y}_0 \in \tilde{\Omega}_0$ of the sequence

$$\tilde{y}_{k+1} = g(\tilde{y}_k), \qquad k = 0, 1, 2, \ldots, \tag{A.10}$$

and prove, similarly as in (i) above, that

$$\|\tilde{y}_k - \tilde{x}_k\| \leq \tau_k - t_k, \qquad k = 1, 2, \ldots, \tag{A.11}$$

and $\tilde{y}_k \in \tilde{\Omega}_0$. Finally, since $\lim_{k\to\infty} \tau_k = \underline{t}$, from (A.11) it follows that $\lim_{k\to\infty} \tilde{y}_k = \lim_{k\to\infty} \tilde{x}_k = \tilde{x}^* \in \tilde{\Omega}_0$.

(iii) To estimate the convergence rate, we note that from part (i) of the proof, we have

$$\|\tilde{x}^* - \tilde{x}_k\| \leq \underline{t} - t_k \tag{A.12}$$

for all $k \geq 0$. Furthermore,

$$\underline{t} - t_k = \varphi(\underline{t}) - \varphi(t_{k-1}) = \varphi'(t')(\underline{t} - t_{k-1}), \tag{A.13}$$

where $t' \in [t_{k-1}, \underline{t}]$. Since $\varphi'(t') = \gamma t' + \varepsilon\beta \leq \gamma\underline{t} + \varepsilon\beta = 1 - (1 - \varepsilon\beta)\sqrt{1 - 2\delta}$, we have

$$\underline{t} - t_k \leq \left[1 - (1 - \varepsilon\beta)\sqrt{1 - 2\delta}\right](\underline{t} - t_{k-1}). \tag{A.15}$$

Proceeding recursively, we obtain

$$\underline{t} - t_k \leq \left[1 - (1 - \varepsilon\beta)\sqrt{1 - 2\delta}\right]^k (\underline{t} - t_0), \tag{A.16}$$

where $t_0 = 0$, and the estimate (3.23) follows.

(iv) We first establish that conditions (3.16) through (3.22) also guarantee that the Newton iterative process

$$\tilde{\mathcal{N}} : \tilde{x}_{k+1} = \tilde{x}_k - \tilde{A}^{-1}\tilde{f}(\tilde{x}_k), \qquad \tilde{x}_0 = \tilde{V}x_0 \tag{A.17}$$

converges to $\tilde{x}^*$. Let us define operator U as

$$U = \left(\tilde{A}_D^{-1}\tilde{A}\right)^{-1} = \left(I + \varepsilon\tilde{A}_D^{-1}\tilde{A}_C\right)^{-1}. \tag{A.18}$$

By (3.17) and (3.19) it follows that U exists and

$$\|U\| \leq \frac{1}{1 - \varepsilon\beta}. \tag{A.19}$$

Definition (A.18) now implies

$$U\tilde{A}_D^{-1} = \tilde{A}^{-1}, \tag{A.20}$$

and, consequently, by (3.16) and (3.18) we get

$$\left\|\tilde{A}^{-1}\tilde{f}(\tilde{x}_0)\right\| \leq \|U\| \left\|\tilde{A}_D^{-1}\tilde{f}(\tilde{x}_0)\right\| \leq \frac{\alpha}{1 - \varepsilon\beta}, \tag{A.21}$$

$$\left\|\tilde{A}^{-1}\tilde{f}''(\tilde{x})\right\| \leq \|U\| \left\|\tilde{A}_D^{-1}\tilde{f}''(\tilde{x})\right\| \leq \frac{\gamma}{1 - \varepsilon\beta} \tag{A.22}$$

for all $\tilde{x} \in \tilde{\Omega}_0$.

We now define operator

$$\tilde{h}(\tilde{x}) = \tilde{x} - \tilde{A}^{-1}\tilde{f}(\tilde{x}) \tag{A.23}$$

and auxiliary function

$$\varphi_1(t) = \frac{\gamma}{1-\varepsilon\beta} \cdot \frac{t^2}{2} + \frac{\alpha}{1-\varepsilon\beta}. \tag{A.24}$$

It should be pointed out that the modification $\psi_1(t) \equiv \varphi_1(t) - t$, that is,

$$\psi_1(t) = \frac{\gamma}{1-\varepsilon\beta}\frac{t^2}{2} - t + \frac{\alpha}{1-\varepsilon\beta} \tag{A.25}$$

has the same roots $\underline{t} = \underline{\rho}$ and $\bar{t} = \bar{\rho}$ defined in (3.22). Furthermore,

$$\left\|\tilde{h}(\tilde{x}_0) - \tilde{x}_0\right\| \leq \varphi_1(0) \tag{A.26}$$

and

$$\begin{aligned} \left\|\tilde{h}'(\tilde{x})\right\| &\leq \left\|\tilde{A}^{-1}\left(\tilde{f}'(\tilde{x}_0) - \tilde{f}'(\tilde{x})\right)\right\| \\ &\leq \frac{\gamma}{1-\varepsilon\beta}\, t = \varphi_1'(t) \end{aligned} \tag{A.27}$$

for all $\|\tilde{x} - \tilde{x}_0\| \leq t, t \in [0, \rho]$. Proceeding as in part (i), we readily conclude that $\tilde{\mathcal{N}}$ converges to the same $\tilde{x}^*$ as $\tilde{\mathcal{N}}_D$.

Since (3.2) and (3.5) hold by assumption from Lemma (3.10), it follows that $\tilde{\mathcal{N}}$ is an expansion of $\mathcal{N}$. Consequently, since $\tilde{x}_0 = \tilde{V}x_0$,

$$\tilde{x}_k = \tilde{V}x_k \in \tilde{M}, \qquad k = 0, 1, 2, \ldots. \tag{A.28}$$

Furthermore, it is easily established that manifold $\tilde{M}$ is closed, implying $\tilde{x}^* = \tilde{V}x^*$ for some $x^* \in \mathbf{R}^n$. It now only remains to show that x^* is the unique solution of $\mathcal{S}$ in Ω_0.

From (3.1), it follows that

$$0 = \tilde{f}(\tilde{x}^*) = \tilde{f}\left(\tilde{V}x^*\right) = Vf(x^*). \tag{A.29}$$

Since V and $\tilde{V}$ are permutation matrices of full rank, we have

$$f(x^*) = 0 \tag{A.30}$$

and, in addition,

$$\|x^* - x_0\| = \left\|\tilde{V}x^* - \tilde{V}x_0\right\| = \|\tilde{x}^* - \tilde{x}_0\| \leq \rho \tag{A.31}$$

since $\tilde{x}^* \in \tilde{\Omega}_0$. Consequently, $x^* \in \Omega_0$, where Ω_0 is defined in (3.25).

Assume now that there exists some $y^* \in \Omega_0, y^* \neq x^*$, which is another solution of $\mathcal{S}$. Then,

$$0 = Vf(y^*) = \tilde{f}\left(\tilde{V}y^*\right) \tag{A.32}$$

is a solution of $\tilde{\mathcal{S}}$. Furthermore, since $y^* \in \Omega_0$,

$$\|\tilde{y}^* - \tilde{x}_0\| = \left\|\tilde{V}y^* - \tilde{V}x_0\right\| = \|y^* - x_0\| \leq \rho \tag{A.33}$$

and, therefore, $\tilde{y}^* \in \tilde{\Omega}_0$. From the fact that $\tilde{V}$ has full rank, it follows that $y^* \neq x^*$ implies $\tilde{y}^* \neq \tilde{x}^*$. This contradicts the uniqueness of $\tilde{x}^*$ in $\tilde{\Omega}_0$. □

Acknowledgments. The authors are grateful to M. Amano, Hitachi Research Laboratory, Ibaraki-ken, Japan, for his useful comments on the paper and experimental results obtained by applying the epsilon decompositions to parallelization of load-flow equations.

REFERENCES

[1] A. Bhaya, E. Kaszkurewicz, and F. Mota, *Asynchronous block-iterative methods for almost linear equations*, Linear Algebra Appl., 154 (1991), pp. 487–508.

[2] M. Brucoli, M. La Scala, F. Torelli, and M. Trovato, *Overlapping decomposition for small disturbance stability analysis of interconnected power networks*, Large Scale Systems, 13 (1987), pp. 115–129.

[3] J. H. Chow, *Time Scale Modeling of Dynamic Networks with Applications to Power Systems*, Springer-Verlag, New York, 1982.

[4] M. L. Crow and M. Ilić, *The parallel implementation of the waveform relaxation method for transient stability simulations*, IEEE Trans. Power Syst., 5 (1990), pp. 922–929 (discussion of, pp. 929–932).

[5] B. N. Datta, *Parallel and large-scale computations in control: Some ideas*, Linear Algebra Appl., 121 (1989), pp. 243–264.

[6] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, *Parallel implementation of a power system dynamic simulation methodology using conjugate gradient method*, IEEE Power Industry Computer Application Conference, 1991, Baltimore, MD, pp. 245–252.

[7] I. S. Duff, *Sparse Matrices and Their Uses*, Academic Press, New York, 1981.

[8] K. A. Gallivan, R. J. Plemmons, and A. H. Sameh, *Parallel algorithms for dense linear algebra computations*, SIAM Rev., 32 (1990), pp. 54–135.

[9] H. H. Happ, *Multicontroller configurations and diakoptics: Dispatch of real power in power pools*, IEEE Trans. Power Apparatus Syst., 88 (1969), pp. 764–772.

[10] ———, *Diakoptics—the solution of system problems by tearing*, Proc. IEEE, 62 (1974), pp. 930–940.

[11] M. Ikeda, and D. D. Šiljak, *Overlapping decompositions, expansions, and contractions of dynamic systems*, Large Scale Systems, 1 (1980), pp. 29–38.

[12] L. V. Kantorovich and G. P. Akilov, *Functional Analysis*, Nauka, Moscow, Russia, 1984.

[13] R. Kasturi and M. S. N. Potti, *Piecewise Newton-Raphson load flow—an exact method using ordered elimination*, IEEE Trans. Power Apparatus Syst., 95 (1976), pp. 1244–1253.

[14] E. Kaszkurewicz, A. Bhaya, and D. D. Šiljak, *On the convergence of parallel asynchronous block-iterative computations*, Linear Algebra Appl., 131 (1990), pp. 139–160.

[15] K. Khorasani and M. A. Pai, *Two time scale decomposition and stability analysis of power systems*, IEE Proc., 135 (1988), pp. 205–212.

[16] T. Y. Lee and F. C. Schweppe, *Distance measures and coherency recognition for transient stability equivalents*, IEEE Trans. Power Apparatus Syst., 92 (1973), pp. 1550–1557.

[17] J. Medanić and B. Avramović, *Solution of load-flow problems in power systems by ε-coupling method*, IEE Proc., 122 (1975), pp. 801–805.

[18] J. M. Ortega and R. G. Voigt, *Solutions of Partial Differential Equations on Vector and Parallel Computers*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1985.

[19] M. A. Pai and R. P. Adgaonkar, *Electromechanical distance measure for decomposition of power systems*, Electrical Power & Energy Systems, 6 (1982), pp. 249–254.

[20] M. E. Sezer and D. D. Šiljak, *Nested ε-decompositions and clustering of complex systems*, Automatica, 22 (1986), pp. 321–331.

[21] ———, *Nested epsilon decompositions of linear systems: Weakly coupled and overlapping blocks*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 521–533.

[22] D. D. Šiljak, *Large-Scale Dynamic Systems: Stability and Structure*, North-Holland, New York, 1978.

[23] ———, *Decentralized Control of Complex Systems*, Academic Press, Cambridge, MA, 1991.

[24] B. Stott, *Review of load-flow calculation methods*, Proc. IEEE, 62 (1974), pp. 916–929.

[25] Y. Wallach, *Calculations and Programs for Power Systems Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[26] J. Zaborszky, G. Huang, and K. W. Lu, *A textured model for computationally efficient reactive power control and management*, IEEE Trans. Power Apparatus Syst., 104 (1985), pp. 1718–1727.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 845–864, July 1994

009

FACTORIZATION OF MATRIX POLYNOMIALS WITH SYMMETRIES *

A. C. M. RAN† AND L. RODMAN‡

Abstract. An $n \times n$ matrix polynomial $L(\lambda)$ (with real or complex coefficients) is called selfadjoint if $L(\lambda) = (L(\bar{\lambda}))^*$ and symmetric if $L(\lambda) = (L(\pm\lambda))^T$. Factorizations of selfadjoint and symmetric matrix polynomials of the form $L(\lambda) = (M(\bar{\lambda}))^* DM(\lambda)$ or $L(\lambda) = (M(\pm\lambda))^T DM(\lambda)$ are studied, where D is a constant matrix and $M(\lambda)$ is a matrix polynomial. In particular, the minimal possible size of D is described in terms of the elementary divisors of $L(\lambda)$ and (sometimes) signature of the Hermitian values of $L(\lambda)$.

Key words. matrix polynomials, symmetries, factorization

AMS subject classifications. 15A22, 15A54, 15A23

1. Introduction.

Let $L(\lambda) = \sum_{j=0}^{l} \lambda^j A_j$ be a matrix polynomial, where $A_j (j = 0, \ldots, l)$ are complex $n \times n$ matrices and λ is a complex parameter. The polynomial $L(\lambda)$ is called *selfadjoint* if $L(\lambda) = (L(\bar{\lambda}))^*$ for all $\lambda \in \mathbb{C}$.

Factorizations of the form

$$L(\lambda) = \left(M\left(\bar{\lambda}\right)\right)^* DM(\lambda), \tag{1.1}$$

where $D = D^*$ is a constant matrix (not necessarily of the same size as $L(\lambda)$) and $M(\lambda)$ is a matrix polynomial, have been studied in the literature under various additional hypotheses (see [Ja], [Co], [GLR1], [GLR2]). The study of factorizations (1.1) is motivated by several applied problems, such as filtering [AM, Chap. 9]. Factorizations of a matrix polynomial $L(\lambda)$ having other types of symmetries, such as $L(\lambda) = (L(-\lambda))^T$ or $L(\lambda) = (L(\lambda))^T$, have been studied in the literature as well (see, e.g., [Lyu1], [Lyu2]). For such polynomials, it is natural to seek factorizations of type

$$L(\lambda) = (M(\varepsilon\lambda))^T DM(\lambda), \tag{1.2}$$

where $D = D^T$ is a constant matrix (not necessarily of the same size as $L(\lambda)$), $M(\lambda)$ is a matrix polynomial, and $\varepsilon = 1$ or $\varepsilon = -1$, as appropriate.

In this paper we identify the minimal possible size of the matrix D in factorizations of types (1.1) and (1.2), where $L(\lambda)$ has the appropriate symmetry. The cases when $L(\lambda)$ has complex coefficients or real coefficients are studied (if $L(\lambda)$ is assumed to be real, then in (1.1) and (1.2) $M(\lambda)$ and D are assumed to be real as well). Our result concerning the factorization (1.1) is a generalization of the main result of [GLR2], where only the case of constant signature was considered under the additional hypothesis that $\det L(\lambda) \not\equiv 0$.

---

* Received by the editors July 8, 1992; accepted for publication (in revised form) March 31, 1993. This research was supported in part by the Institute for Mathematics and its Applications with funds provided by National Science Foundation.

† Faculteit Wiskunde en Informatica, Vrije Universiteit, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands.

‡ Department of Mathematics, College of William and Mary, Williamsburg, Virginia 23187-8795 (cs.wm.edu). This research was partially supported by National Science Foundation grant DMS-9000839 and by National Science Foundation International Cooperation grant with The Netherlands.

In §2, we also present general factorization results in an abstract framework for matrix polynomials over a field having suitable symmetries. These results, although independently interesting, play an auxiliary role in this paper, serving as essential ingredients in the proofs of the main results given in §§3–6.

The following notation is used throughout the paper. $\mathbb{R}(\mathbb{C})$ denotes the real (complex) field, and I_k is the $k \times k$ unit matrix. A^T (respectively, A^*) stands for the transpose (respectively, conjugate transpose) of a matrix A, and $(A^T)^{-1}$ (respectively, $(A^*)^{-1}$) is abbreviated as A^{-T} (respectively, A^{-*}). A block diagonal matrix with blocks $Z_1, \ldots, Z_m$ on the main diagonal will be denoted $Z_1 \oplus \cdots \oplus Z_m$ or diag $(Z_1, \ldots, Z_m)$. For a Hermitian $n \times n$ matrix X, let $\nu_+(X)$ (respectively, $\nu_-(X)$, or $\nu_0(X)$) be the number of positive (respectively, negative, or zero) eigenvalues of X counted with multiplicities. Thus,

$$\nu_+(X) + \nu_-(X) + \nu_0(X) = n.$$

Given a matrix polynomial $L(\lambda)$ over $\mathbb{C}$, its *general rank* $r(L)$ is defined by

$$r(L) = \max_{\lambda_0 \in \mathbb{C}} \{\text{rank } L(\lambda_0)\}.$$

This coincides (when $F = \mathbb{C}$) with the notion of general rank introduced and used in §2 for a matrix polynomial over a field F. The points $\lambda_0 \in \mathbb{C}$ for which rank $L(\lambda_0) = r(L)$ are called *regular points* of $L(\lambda)$; all other points $\lambda_0 \in \mathbb{C}$ are called *singular points* of $L(\lambda)$. Clearly, the set of singular points is finite (or possibly empty). An $n \times n$ matrix polynomial $L(\lambda)$ is called *regular* if $r(L) = n$, or, equivalently, if $\det L(\lambda) \not\equiv 0$.

2. Symmetric matrix polynomials over a general field. Let F be a (commutative) field, and let $F[\lambda]$ be the ring of polynomials over F in one variable λ. Matrices $L(\lambda)$ with entries in $F[\lambda]$ are called *matrix polynomials* (over F). It is well known (see, e.g., [M]) that every $m \times n$ matrix polynomial $L(\lambda)$ admits a representation (called the *Smith form*)

$$L(\lambda) = E(\lambda) \text{ diag } (d_1(\lambda), d_2(\lambda), \ldots, d_r(\lambda), 0, \ldots, 0) F(\lambda), \tag{2.1}$$

where $E(\lambda)$ and $F(\lambda)$ are matrix polynomials with sizes $m \times m$ and $n \times n$, respectively, and having constant nonzero determinants, and $d_1(\lambda), \ldots, d_r(\lambda)$ are monic scalar polynomials (over F) such that $d_i(\lambda)$ divides $d_{i+1}(\lambda)$ $(i = 1, \ldots, r-1)$. The polynomials $d_i(\lambda)$ are called *invariant polynomials* of $L(\lambda)$; these polynomials, as well as their number r, are uniquely determined by $L(\lambda)$: $r \times r$ is the maximal size of a square submatrix in $L(\lambda)$ with determinant not identically zero and, for $i = 1, \ldots, r$ the product $d_1(\lambda) \cdots d_i(\lambda)$ is the greatest common divisor of the determinants of all $i \times i$ submatrices in $L(\lambda)$.

The number r coincides with the *general rank* of $L(\lambda)$.

In this section we study factorizations of symmetric matrix polynomials using the Smith form as our main tool.

From now on we assume that the characteristic of F is different from two. For a given automorphism σ of F such that $\sigma^2 = \text{identity}$, and for fixed $\varepsilon = \pm 1$ consider the following transformation: for $a(\lambda) = \sum a_j \lambda^j \in F[\lambda]$, let

$$a_*(\lambda) = \sum \sigma(a_j)(\varepsilon\lambda)^j = \sum \sigma(a_j)\varepsilon^j \lambda^j \in F[\lambda]. \tag{2.2}$$

For an $m \times n$ matrix polynomial

$$X(\lambda) = [x_{ij}(\lambda)]_{i=1,j=1}^{m,n} \text{ over } F, \quad \text{define } X_*(\lambda) = [\tilde{x}_{ij}(\lambda)]_{i=1,j=1}^{n,m},$$

where $\tilde{x}_{ij}(\lambda) = [x_{ji}(\lambda)]_*$. We have

(i) $[X(\lambda)Y(\lambda)]_* = Y(\lambda)_* X(\lambda)_*$,

(ii) $[X(\lambda)]_{**} = X(\lambda)$,

(iii) $[x(\lambda)X(\lambda) + y(\lambda)Y(\lambda)]_* = x_* X_* + y_* Y_*$ for scalar polynomials $x(\lambda)$ and $y(\lambda)$.

(iv) If $\det X(\lambda) \equiv \text{const.} \neq 0$, then $[X_*(\lambda)]^{-1} = ([X(\lambda)]^{-1})_*$.

These rules are used often in the sequel.

An $m \times n$ matrix polynomial $L(\lambda)$ is called *generally invertible* if all its invariant polynomials are constant one. The terminology is justified by the fact that $L(\lambda)$ is generally invertible if and only if $L(\lambda)$ has a generalized inverse, i.e., a matrix polynomial $N(\lambda)$ such that $N(\lambda)L(\lambda)N(\lambda) = N(\lambda)$ and $L(\lambda)N(\lambda)L(\lambda) = L(\lambda)$ (this fact is easily proved using the Smith form). A matrix polynomial $L(\lambda)$ is called *right* (respectively, *left*) *invertible* if there exists a matrix polynomial $N(\lambda)$ such that $L(\lambda)N(\lambda) \equiv I$ (respectively, $N(\lambda)L(\lambda) \equiv I$).

We now state one of the main factorization results of this section.

THEOREM 2.1. *Let $L(\lambda)$ be an $n \times n$ generally invertible matrix polynomial such that*

$$L(\lambda) = L_*(\lambda), \tag{2.3}$$

and let r be the general rank of $L(\lambda)$. Then $L(\lambda)$ can be factorized in the form

$$L(\lambda) = M_*(\lambda) D M(\lambda), \tag{2.4}$$

where $M(\lambda)$ is an $r \times n$ right invertible matrix polynomial and D is an $r \times r$ constant matrix such that $D = D_*$.

Conversely, if (2.4) *holds for an $r \times n$ right invertible matrix polynomial $M(\lambda)$ and a constant matrix $D = D_*$, then $L(\lambda)$ satisfies* (2.3), *is generally invertible, and has general rank r.*

Proof. The converse statement is easy. Indeed, if

$$M(\lambda) = \tilde{E}(\lambda)[I \quad 0]\tilde{F}(\lambda)$$

is the Smith form for $M(\lambda)$, then

$$\begin{aligned} L(\lambda) &= \tilde{F}_*(\lambda)\begin{bmatrix} I \\ 0 \end{bmatrix}\tilde{E}_*(\lambda) D \tilde{E}(\lambda)[I \quad 0]\tilde{F}(\lambda) \\ &= \tilde{F}_*(\lambda)\begin{bmatrix} \tilde{E}_*(\lambda) & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \tilde{E}(\lambda) & 0 \\ 0 & I \end{bmatrix}\tilde{F}(\lambda). \end{aligned}$$

So by uniqueness of the Smith form, $L(\lambda)$ is generally invertible and has general rank r. The verification of (2.3) is trivial.

We now prove the direct statement.

Observe that the proof is easily reduced to the case when $r = n$, i.e., $\det L(\lambda) \equiv \text{const.} \neq 0$. Indeed, let $L(\lambda) = E(\lambda)D(\lambda)F(\lambda)$ be the Smith form of $L(\lambda)$, and let $\tilde{L}(\lambda) = (F_*(\lambda))^{-1}L(\lambda)F(\lambda)^{-1}$. Clearly, $\tilde{L}(\lambda)$ is a matrix polynomial $\tilde{L}(\lambda) = \tilde{L}_*(\lambda)$, and because of the equality $\tilde{L}(\lambda) = (F_*(\lambda))^{-1}E(\lambda)D(\lambda)$ the last $n - r$ rows and columns of $\tilde{L}(\lambda)$ are zeros. Obviously, it suffices to prove the direct statement for the

$r \times r$ matrix polynomial $N(\lambda)$ formed by the first r rows and columns of $\tilde{L}(\lambda)$. As $\det N(\lambda) \equiv \text{const.} \neq 0$, the required reduction is accomplished.

We assume from now on that $\det L(\lambda) \equiv \text{const.} \neq 0$. In this case the direct statement follows from Theorem 3 in [Lyu1] (see also [Lyu2]). We outline an alternative procedure developed in [Co]. As in [Co] or [GLR2] (§4), we prove that there exists an $n \times n$ matrix polynomial $X(\lambda)$ with $\det X(\lambda) \equiv \text{const.} \neq 0$ such that for the matrix polynomial

$$A(\lambda) := X_*(\lambda) L(\lambda) X(\lambda) = [\alpha_{ij}(\lambda)]_{i,j=1}^{n}$$

either $\alpha_{11} \equiv 0$ or $A(\lambda)$ is diagonally dominant (i.e., for $j = 1, \dots, n$, the degree of $\alpha_{jj}(\lambda)$ is bigger than the degrees of all nonzero entries in the jth row and the jth column in $A(\lambda)$). Because of this fact, without loss of generality we can assume that either $L(\lambda)$ is diagonally dominant or the (1, 1) entry in $L(\lambda)$ is identically zero. If $L(\lambda)$ is diagonally dominant, then it must be constant, and we are done. So let

$$L = \begin{bmatrix} 0 & a_* \\ a & A_1 \end{bmatrix}, \qquad L^{-1} = \begin{bmatrix} \gamma & c_* \\ c & C_1 \end{bmatrix},$$

where $A_1 = A_{1*}$ and $C_1 = C_{1*}$ are $(n-1) \times (n-1)$ matrix polynomials. Now put

$$y = \frac{1}{2}(1 - c_* A_1 c), \quad x = -A_1 c - a y, \quad \text{and} \quad Y = \begin{bmatrix} y & x_* \\ c & I \end{bmatrix}.$$

A calculation shows that

$$\begin{bmatrix} 1 & -x_* \\ 0 & I \end{bmatrix} Y \begin{bmatrix} 1 & 0 \\ -c & I \end{bmatrix} \equiv I,$$

and so $\det Y = 1$. Another straightforward calculation shows that

$$Y_* L Y = \begin{bmatrix} 1 & 0 \\ 0 & L_0 \end{bmatrix},$$

where $L_0(\lambda)$ is an $(n-1) \times (n-1)$ matrix polynomial. Thus, we have reduced the size of L by one and can complete the proof by induction on n. □

As the proof of Theorem 2.1 shows, the constant matrix D can be taken to be diagonal.

If $L(\lambda)$ is not generally invertible, then easy examples show that the representation (2.4) (with D having the size equal to the general rank of $L(\lambda)$) is not always possible, even if we omit the requirement that $M(\lambda)$ is right invertible. We can, however, obtain a factorization result for not generally invertible $L(\lambda)$ if we allow D to be a polynomial (with special properties). To state and prove this result we need the concept of elementary divisors. Let $L(\lambda)$ be an $m \times n$ matrix polynomial with invariant polynomials $d_1(\lambda), \dots, d_r(\lambda)$, where $d_s(\lambda), d_{s+1}(\lambda), \dots, d_r(\lambda)$ are nonconstant (if $L(\lambda)$ is generally invertible, then we say that $L(\lambda)$ has no elementary divisors). Factor

$$d_i(\lambda) = (f_{i1}(\lambda))^{\alpha_{i1}} (f_{i2}(\lambda))^{\alpha_{i2}} \cdots (f_{ik_i}(\lambda))^{\alpha_{ik_i}}; \qquad i = s, s+1, \dots, r \tag{2.5}$$

where $f_{i1}(\lambda), \dots, f_{i,k_i}(\lambda)$ are irreducible and pairwise relatively prime nonconstant monic scalar polynomials (over F). The collection of factors $(f_{ij}(\lambda))^{\alpha_{ij}} (j = 1, \dots, k_i; i = s, s+1, \dots, r)$, where each factor is repeated as many times as it occurs in (2.5), is called the *elementary divisors* of $L(\lambda)$, and the positive integer α_{ij} is called the

order of the elementary divisor $(f_{ij}(\lambda))^{\alpha_{ij}}$. Because of the divisibility relations among invariant polynomials, the collection of elementary divisors of $L(\lambda)$ determines the invariant polynomials uniquely, and therefore is invariant under the transformations $L(\lambda) \to E(\lambda)L(\lambda)F(\lambda)$, where

$$\det E(\lambda) \equiv \text{const.} \neq 0, \qquad \det F(\lambda) \equiv \text{const.} \neq 0.$$

THEOREM 2.2. *Let $L(\lambda)$ be an $n \times n$ matrix polynomial such that*

$$L(\lambda) = L_*(\lambda),$$

and let r be the general rank of $L(\lambda)$. Furthermore, let $\{f_1(\lambda)^{\alpha_1}, \dots, f_q(\lambda)^{\alpha_q}\}$ be the collection of elementary divisors of $L(\lambda)$. Then $L(\lambda)$ admits a factorization

$$L(\lambda) = M_*(\lambda)\, D(\lambda)\, M(\lambda),$$

where $M(\lambda)$ is an $r \times n$ matrix polynomial and $D(\lambda) = D_*(\lambda)$ is an $r \times r$ matrix polynomial. Moreover, the collection of elementary divisors of $D(\lambda)$ is $\{f_j(\lambda) : j \in J\}$, where the subset J of $\{1, 2, \dots, q\}$ consists precisely of those indices j for which $f_j = \varepsilon^{\text{degree } f_j} f_{j^*}$ and α_j is odd.

Recall that $\varepsilon = \pm 1$ is taken from (2.2).

Proof. As in the proof of Theorem 2.1, we can assume that $r = n$. Let $L(\lambda) = E(\lambda)D_1(\lambda)F(\lambda)$ be the Smith form $L(\lambda)$, where the invariant polynomials $d_1(\lambda), \dots, d_r(\lambda)$ are on the main diagonal of $D_1(\lambda)$. Because $L = L_*$, and by the uniqueness of invariant polynomials, we have in fact $d_{i^*} = \varepsilon^{\text{degree } d_i} d_i$ $(i = 1, \dots, r)$. The factor $\varepsilon^{\text{degree } d_i}$ appears because d_i is monic (this is part of the definition of invariant polynomials) while the leading coefficient of d_{i^*} is $\varepsilon^{\text{degree } d_i}$. In the sequel it is convenient to denote $f_+ = \varepsilon^{\text{degree } f} f_*$ for a scalar polynomial f. Thus, $d_{i+} = d_i$. Observe that f_+ is monic if f is monic and that $(f_1 f_2)_+ = f_{1+} f_{2+}$ for all pairs of polynomials f_1, f_2.

Replacing L by $F_*^{-1} L F^{-1}$, we can further assume without loss of generality that the ith column of L is divisible by $d_i(\lambda)$ $(i = 1, \dots, n)$. By symmetry, the ith row of L is divisible by $d_{i^*}(\lambda)$. Let the nonconstant invariant polynomials of $L(\lambda)$ be $d_s(\lambda), \dots, d_r(\lambda)$ and factor them as in (2.5). Then

$$d_i = \prod_{j=1}^{k_i} (f_{ij})^{\alpha_{ij}} = \prod_{j=1}^{k_i} (f_{ij+})^{\alpha_{ij}} \qquad (i = s, s+1, \dots, r),$$

and by the uniqueness of the decomposition (2.5), we see that the set $\{f_{i1}, \dots, f_{ik_i}\}$ must consist of selfsymmetric polynomials $(f_{ij} = f_{ij+})$ and/or of pairs of mutually symmetric polynomials $f_{ij_1} = f_{ij_2+}$; in this case necessarily $\alpha_{ij_1} = \alpha_{ij_2}$. Say, $f_{i1}, \dots, f_{i,p_i}$ are selfsymmetric and

$$f_{i,p_i+1} = (f_{i,p_i+2})_+, \dots, f_{i,p_i+2q_i-1} = (f_{i,p_i+2q_i})_+;$$

here $p_i + 2q_i = k_i$. Let i_0 be the smallest index such that $\alpha_{i_0 j} > 1$ for some $j \in \{1, \dots, p_i\}$; say, $\alpha_{i_0 1} > 1$ (if no such i_0 exists, we put

$$h_i = \prod_{j=1}^{q_i} (f_{i,p_i+2j})^{\alpha_{i,p_i+2j}} \qquad (i = s, s+1, \dots, r).$$

Define

$$h_i = \prod_{j=1}^{q_i} (f_{i,p_i+2j})^{\alpha_{i,p_i+2j}} \qquad i = s, s+1, \ldots, i_0 - 1,$$

$$h_i = f_{i1} \prod_{j=1}^{q_i} (f_{i,p_i+2j})^{\alpha_{i,p_i+2j}} \qquad i = i_0, i_0+1, \ldots, r,$$

where we assume that the elementary divisors are numbered so that $f_{i1} = f_{i_0 1}$ for $i = i_0 + 1, \ldots, r$. To make the subsequent formulas more uniform, we define also $h_i \equiv 1$ for $i = 1, \ldots, s-1$. The divisibility relations among the d_i's imply that whenever $f_{i_1 j_1} = f_{i_2 j_2}$, where $i_1 < i_2$, then $\alpha_{i_1 j_1} \leq \alpha_{i_2 j_2}$. It follows that h_i divides h_{i+1} $(i = 1, \ldots, r-1)$.

The formulas (2.5) lead to the factorization

$$d_i = h_{i^*} g_i h_i \qquad (i = 1, \ldots, r), \tag{2.6}$$

where $g_i \equiv 1$ for $i = 1, \ldots, s-1$;

$$g_i = \pm \prod_{j=1}^{p_i} (f_{ij})^{\alpha_{ij}} \quad \text{for } i = s, \ldots, i_0 - 1,$$

$$g_i = \pm f_{i1}^{\alpha_{ij}-2} \prod_{j=2}^{p_i} (f_{ij})^{\alpha_{ij}} \quad \text{for } i = i_0, i_0+1, \ldots, r,$$

(the sign $+$ or $-$ in g_i is chosen so that g_i is monic). Clearly, g_i divides g_{i+1} for $i = 1, \ldots, r-1$.

In view of (2.6) we now have a factorization

$$L(\lambda) = \operatorname{diag}\ (1, \ldots, 1, h_{s^*}(\lambda), \ldots, h_{r^*}(\lambda))\, \tilde{L}(\lambda)\ \operatorname{diag}\ (1, \ldots, 1, h_s(\lambda), \ldots, h_r(\lambda)) \tag{2.7}$$

for some matrix polynomial $\tilde{L} = \tilde{L}_*$. Denote by $\tilde{d}_1(\lambda), \ldots, \tilde{d}_r(\lambda)$ the invariant polynomials of $\tilde{L}(\lambda)$. Equality (2.7), together with the Binet–Cauchy formula for determinants of submatrices in the product of several matrices implies the following. The determinant of every $j \times j$ submatrix in $L(\lambda)$ is a linear combination (with polynomial coefficients) of determinants of $j \times j$ submatrices in $\tilde{L}(\lambda)$ when the determinants are multiplied by $\prod_{i=1}^{j} (h_{i^*}(\lambda) h_i(\lambda))$. It follows that $d_1(\lambda) \cdots d_j(\lambda)$ divides $\tilde{d}_1(\lambda) \cdots \tilde{d}_j(\lambda) \prod_{i=1}^{j} (h_{i^*}(\lambda) h_i(\lambda))$ for $j = 1, \ldots, r$. The equality (2.6) now shows that $g_1(\lambda) \cdots g_j(\lambda)$ divides $\tilde{d}_1(\lambda) \cdots \tilde{d}_j(\lambda)$. On the other hand, for the (i,j)th entries $\tilde{p}_{ij}$ of $\tilde{L}$ and p_{ij} of L, respectively, we obtain

$$\tilde{p}_{ij} = h_{i^*}^{-1} p_{ij} h_j^{-1} = h_{i^*}^{-1} p_{ij} d_i^{-1} h_{j^*} g_j,$$

and since (assuming $i \leq j$) both $p_{ij} d_i^{-1}$ and $h_{i^*}^{-1} h_{j^*}$ are polynomials, $\tilde{p}_{ij}$ is divisible by g_j. Also, $\tilde{p}_{ij}$ is divisible by g_i (because g_i divides g_j if $i \leq j$). By the symmetry of $\tilde{L}$, we see that $\tilde{p}_{ij}$ is divisible by $g_{\max(i,j)}$. Therefore, the determinant of every $j \times j$ submatrix of $\tilde{L}$ is divisible by $g_1 \cdots g_j$. Consequently, $\tilde{d}_1 \cdots \tilde{d}_j$ divides $g_1 \cdots g_j$. Comparing this result with the previously obtained opposite divisibility relation, we conclude that $g_1, \ldots, g_r$ are the invariant polynomials of $\tilde{L}$.

We repeat the procedure given above with L replaced by $\tilde{L}$, and so on, until (after a finite number of steps) we obtain a matrix polynomial $D(\lambda)$ with the properties required in Theorem 2.2. □

3. Factorization of selfadjoint matrix polynomials on the real axis. In this section we consider matrix polynomials $L(\lambda)$ over $\mathbb{C}$ with the following property:

$$L(\lambda) = \left(L\left(\bar{\lambda}\right)\right)^*, \qquad \lambda \in \mathbb{C}.$$

Such polynomials are called *selfadjoint.*

THEOREM 3.1. *Let $L(\lambda)$ be a selfadjoint $n \times n$ matrix polynomial. Then $L(\lambda)$ admits a factorization*

$$L(\lambda) = \left(M\left(\bar{\lambda}\right)\right)^* D\, M(\lambda), \tag{3.1}$$

where D is an $m \times m$ constant Hermitian matrix and $M(\lambda)$ an $m \times n$ matrix polynomial, if and only if

$$m \geq m_0, \tag{3.2}$$

where

$$m_0 = \max_{\lambda \in \mathbb{R}} \nu_+(L(\lambda)) + \max_{\lambda \in \mathbb{R}} \nu_-(L(\lambda)). \tag{3.3}$$

Moreover, in all factorizations (3.1) *having the minimal size $m_0 \times m_0$ of D, the matrix D is uniquely determined up to congruence: D has* $\max \nu_+(L(\lambda))$ *positive eigenvalues and* $\max \nu_-(L(\lambda))$ *negative eigenvalues (multiplicities counted).*

We can say more about the spectral properties of the factor $M(\lambda)$ in (3.1). A set Λ of nonreal numbers is called a *c-set* (with respect to a selfadjoint matrix polynomial $L(\lambda)$) if Λ is a maximal (by inclusion) set of nonreal singular points of $L(\lambda)$ with the property that $\lambda_0 \in \Lambda \Longrightarrow \bar{\lambda}_0 \notin \Lambda$. (The case when a c-set is empty is not excluded.) The concept of c-set was introduced and used in [GLR1] and [GLR3]. It turns out that, given $L(\lambda)$ as in Theorem 3.1, and given a c-set Λ, there exists a factorization (3.1), where D is $m_0 \times m_0$ and where the set of nonreal singular points of $M(\lambda)$ coincides with Λ. This statement follows as a by-product of the proof of Theorem 3.1.

Theorem 3.1 admits an alternative formulation. An $n \times n$ matrix polynomial $M(\lambda)$ will be called *elementary* if $r(M) = 1$ and $M(\lambda)$ is positive semidefinite for all real λ. It is not difficult to see (this fact is actually a particular case of Theorem 3.1) that $M(\lambda)$ is elementary if and only if $M(\lambda)$ is of the form $M(\lambda) = x(\lambda)(x(\bar{\lambda}))^*$, where $x(\lambda) \not\equiv 0$ is an $n \times 1$ column polynomial. One can consider elementary matrix polynomials as building blocks for selfadjoint matrix polynomials in the same spirit in which the constant rank-1 positive semidefinite matrices are building blocks for constant Hermitian matrices.

THEOREM 3.2. *Any selfadjoint $n \times n$ matrix polynomial $L(\lambda)$ admits a representation*

$$L(\lambda) = \sum_{j=1}^{m} \varepsilon_j M_j(\lambda), \tag{3.4}$$

where $\varepsilon_j = \pm 1$ and $M_j(\lambda)$ are elementary matrix polynomials. The number m of terms in (3.4) *is greater than or equal to m_0, where m_0 is given by* (3.3); *if $m = m_0$,*

then exactly $\max \nu_+(L(\lambda))$ *of the* ε_j*'s are equal to* $+1$ *and exactly* $\max \nu_-(L(\lambda))$ *of the* ε_j*'s are equal to* -1.

To obtain Theorem 3.2 from Theorem 3.1, assume (without loss of generality) that D in (3.1) is a diagonal matrix with ± 1's on the main diagonal. Then let $M_j(\lambda) = (x_j(\bar{\lambda}))^* x_j(\lambda)$, where $x_j(\lambda)$ is the jth row of $M(\lambda)$, to produce the formula (3.4).

COROLLARY 3.3. *Any selfadjoint* $n \times n$ *matrix polynomial admits a factorization* (3.1), *or a representation* (3.4), *where* $m \leq 2n$.

There are selfadjoint matrix polynomials, for example, $L(\lambda) = \lambda I$, for which there do not exist representations (3.1) or (3.4) with $m < 2n$.

The rest of this section is devoted to the proof of Theorem 3.1.

We start with the easy direction. Let a factorization (3.1) be given, and let λ_0 be a real point for which

$$\nu_+(L(\lambda_0)) = \max_{\lambda \in \mathbb{R}} \nu_+(L(\lambda)).$$

As $L(\lambda_0) = Y^* D Y$, where $Y = M(\lambda_0)$, the Hermitian matrix D must have at least $\nu_+(L(\lambda_0))$ positive eigenvalues. Analogously, D must have at least $\nu_-(L(\lambda_1))$ negative eigenvalues, where $\lambda_1 \in \mathbb{R}$ is chosen so that

$$\nu_-(L(\lambda_1)) = \max_{\lambda \in \mathbb{R}} \nu_-(L(\lambda)).$$

We obtain therefore the inequality (3.2). It is also clear that in any factorization (3.1), where D is $m_0 \times m_0$, the Hermitian matrix D is unique up to congruence.

It remains to show that a given selfadjoint matrix polynomial $L(\lambda)$ admits a factorization (3.1) with $m_0 \times m_0$ the size of D. This is the difficult part and we need some preliminaries. Note that $L(\lambda)$ is selfadjoint if and only if $L = L_*$, where the transformation $a \to a_*$ is defined as in §2, with $F = \mathbb{C}, \sigma(x) = \bar{x}$ $(x \in \mathbb{C})$, and $\varepsilon = 1$. Nevertheless, here the general results of §2 are not used because the preliminary results we need (such as Proposition 3.4 below) are already available in the literature. It should be noted, however, that the result of Theorem 2.2 plays an essential role in the proof of Proposition 3.4.

First observe that there exists an $n \times n$ matrix polynomial $N(\lambda)$ with constant nonzero determinant such that

$$L(\lambda) = \left(N(\bar{\lambda})\right)^* \begin{bmatrix} L_0(\lambda) & 0 \\ 0 & 0 \end{bmatrix} N(\lambda), \tag{3.5}$$

where $L_0(\lambda)$ is a selfadjoint $k \times k$ matrix polynomial, $k = r(L)$. For example, see Theorem 32.4 in [M], where (3.5) is proved for symmetric matrices over principal ideal rings with $(N(\bar{\lambda}))^*$ replaced by $N(\lambda)^T$; the same proof works to produce (3.5). Also, (3.5) can be obtained without difficulties from the Smith form of $L(\lambda)$ (see §2). Because of (3.5) we can (and do) assume from the very beginning, that the general rank of L is equal to n, i.e., $\det L(\lambda) \not\equiv 0$.

Our next observation is that the result of Theorem 3.1 is known in the case where $L(\lambda)$ has *constant signature*, i.e., $\nu_+(L(\lambda))$, and therefore also $\nu_-(L(\lambda))$ and $\nu_0(L(\lambda))$, is constant for all real regular points λ.

PROPOSITION 3.4 ([GLR2]). *Let* $L(\lambda)$ *be a selfadjoint* $n \times n$ *matrix polynomial such that*

$$\max_{\lambda \in \mathbb{R}} \nu_+(L(\lambda)) + \max_{\lambda \in \mathbb{R}} \nu_-(L(\lambda)) = n$$

(*necessarily* $\det L(\lambda) \not\equiv 0$). *Then* $L(\lambda)$ *admits a factorization* (3.1) *in which* D *is* $n \times n$.

We now prove the following lemma.

LEMMA 3.5. *Let* $L(\lambda)$ *be a selfadjoint* $n \times n$ *matrix polynomial with* $\det L(\lambda) \not\equiv 0$, *and let* $m_0 \geq n$ *be defined by* (2.3). *Then there exists an* $m_0 \times m_0$ *selfadjoint matrix polynomial* $\tilde{L}(\lambda)$ *such that*

$$\tilde{L}(\lambda) = \begin{bmatrix} L(\lambda) & 0 \\ 0 & * \end{bmatrix}$$

and such that

$$\max_{\lambda \in \mathbb{R}} \nu_+(\tilde{L}(\lambda)) + \max_{\lambda \in \mathbb{R}} \nu_-(\tilde{L}(\lambda)) = m_0, \tag{3.6}$$

or, equivalently, $\tilde{L}$ *is regular and has constant signature.*

Proof. By Rellich's theorem [R] (see also [GLR3]) the eigenvalues $\mu_1(\lambda), \dots, \mu_n(\lambda)$ of $L(\lambda)$ for λ real can be enumerated so that $\mu_1(\lambda), \dots, \mu_n(\lambda)$ are real analytic functions of the real variable λ. Clearly, $\lambda_0 \in \mathbb{R}$ is a singular point of $L(\lambda)$ if and only if λ_0 is a zero of at least one of the analytic functions $\mu_1(\lambda), \dots, \mu_n(\lambda)$. Let $\lambda_0 \in \mathbb{R}$ be a singular point of $L(\lambda)$, and let

$$\Omega(\lambda_0) = \{1 \leq j \leq n | \mu_j(\lambda_0) = 0\}.$$

For every $j \in \Omega(\lambda_0)$, let m_j be the multiplicity of λ_0 as a zero of $\mu_j(\lambda)$, and let ε_j be the sign of the nonzero real number $[\mu_j^{(m_j)}(\lambda)]|_{\lambda=\lambda_0}$. (We suppress the dependence of m_j and ε_j on λ_0 in the notation.) Define the integer $q(\lambda_0)$ by

$$\begin{aligned} q(\lambda_0) = &\{\text{number of indices } j \in \Omega(\lambda_0) \text{ such that } m_j \text{ is odd and } \varepsilon_j = 1\} \\ &- \{\text{number of indices } j \in \Omega(\lambda_0) \text{ such that } m_j \text{ is odd and } \varepsilon_j = -1\}. \end{aligned}$$

From the definition of $q(\lambda_0)$ it is clear that

$$\nu_+(L(\lambda_0 + \varepsilon)) - \nu_+(L(\lambda_0 - \varepsilon)) = q(\lambda_0) \tag{3.7}$$

for all sufficiently small $\varepsilon > 0$. It is easy to see that

$$m_0 := \max_{\lambda \in \mathbb{R}} \nu_+(L(\lambda)) + \max_{\lambda \in \mathbb{R}} \nu_-(L(\lambda)) = n + \max |\nu_+(L(\lambda_1)) - \nu_+(L(\lambda_2))|, \tag{3.8}$$

where the maximum is taken over all regular real points λ_1 and λ_2. Also, it follows from (3.7) that

$$\max |\nu_+(L(\lambda_1)) - \nu_+(L(\lambda_2))| = \max_{\lambda_1 < \lambda_2} \left| \sum_{\lambda_1 < \lambda_0 < \lambda_2} q(\lambda_0) \right|, \tag{3.9}$$

where the summation in the right-hand side of (3.9) is over all singular points λ_0 of $L(\lambda)$ in the interval $\lambda_1 < \lambda_0 < \lambda_2$.

Denote the right-hand side of (3.9) by p. We now construct p scalar real polynomials $r_1(\lambda), \dots, r_p(\lambda)$ with the following properties:

Property 1. all zeros of $r_j(\lambda)$ $(j = 1, \dots, p)$ are real and simple and belong to the set S of real singular points λ_0 of $L(\lambda)$ for which $q(\lambda_0) \neq 0$;

Property 2. for every $\lambda \in S$ exactly $|q(\lambda_0)|$ polynomials among $r_1(\lambda), \dots, r_p(\lambda)$ have λ_0 as their zeros, and for each $r_j(\lambda)$ such that $r_j(\lambda_0) = 0$ we have $q(\lambda_0) r_j'(\lambda_0) < 0$.

The definition of p ensures that such polynomials $r_1(\lambda), \ldots, r_p(\lambda)$ can indeed be constructed. Let

$$\tilde{L}(\lambda) = \operatorname{diag}\left(L(\lambda), r_1(\lambda), \ldots, r_p(\lambda)\right).$$

By Property 1, and in view of (3.8) and (3.9), it is easy to see that the number of positive eigenvalues of $\tilde{L}(\lambda)$ is constant for every real λ that is a regular point for $L(\lambda)$. The assertion in (3.6) therefore follows. □

Now we can easily finish the proof of Theorem 3.1. Indeed, given a selfadjoint matrix polynomial $L(\lambda)$ with $\det L(\lambda) \not\equiv 0$, construct $\tilde{L}(\lambda)$ as in Lemma 3.5 and apply Proposition 3.4 to $\tilde{L}(\lambda)$:

$$\tilde{L}(\lambda) = \left(N(\bar{\lambda})\right)^* D N(\lambda),$$

where D is a constant $m_0 \times m_0$ Hermitian matrix. Then (3.1) holds for $M(\lambda)$ formed by first n columns of $N(\lambda)$.

4. Factorization of real symmetric matrix polynomials. Let $L(\lambda) = \sum_{j=0}^{l} \lambda^j A_j$ be a *real symmetric* matrix polynomial, i.e., A_j $(j = 0, \ldots, l)$ are real symmetric $n \times n$ matrices. For such polynomials $L(\lambda)$ we consider factorizations

$$L(\lambda) = (M(\lambda))^T D M(\lambda), \tag{4.1}$$

where D is a constant real symmetric $m \times m$ matrix and $M(\lambda)$ is a matrix polynomial with real coefficients.

It is convenient to state the next theorem in terms of elementary divisors (see §2 for definitions of the concepts related to elementary divisors).

THEOREM 4.1. *Let $L(\lambda)$ be a real symmetric $n \times n$ matrix polynomial and assume that the elementary divisors of $L(\lambda)$ that are powers of irreducible quadratic polynomials (over $\mathbb{R}$) all have even orders. Then $L(\lambda)$ admits a factorization* (4.1) *if and only if $m \geq m_0$, where m_0 is defined by* (3.3). *Moreover, in factorization* (4.1) *with the minimal possible size of D, the matrix D is uniquely determined up to congruence and has exactly $\max_{\lambda \in \mathbb{R}} \nu_+(L(\lambda))$ positive eigenvalues and exactly $\max_{\lambda \in \mathbb{R}} \nu_-(L(\lambda))$ negative eigenvalues, multiplicities counted. Alternatively, $L(\lambda)$ admits a representation*

$$L(\lambda) = \sum_{j=1}^{m} \varepsilon_j M_j(\lambda), \tag{4.2}$$

where $M_j(\lambda)$ are real elementary matrices and $\varepsilon_j = \pm 1$, if and only if $m \geq m_0$. In case $m = m_0$, the number of $+1$'s (respectively, -1's) among the $\varepsilon_1, \ldots, \varepsilon_m$ is exactly $\max_{\lambda \in \mathbb{R}} \nu_+(L(\lambda))$ (respectively, $\max_{\lambda \in \mathbb{R}} \nu_-(L(\lambda))$).

In particular, Theorem 4.1 applies if all singular points of $L(\lambda)$ are real.

Proof. The only if part (easy direction) is proved as in the proof of Theorem 3.1. Also, we can easily reduce the proof to the case in which $\det L(\lambda) \not\equiv 0$. Using Theorem 2.2 (with $F = \mathbb{R}, \sigma = \text{identity}, \varepsilon = 1$), we can further assume that all elementary divisors of $L(\lambda)$ are first degree polynomials (necessarily with real roots). From now on the proof proceeds in the same way as that of Theorem 3.1. The role of Proposition 3.4 is played by Proposition 4.2 below. □

PROPOSITION 4.2. *Let $L(\lambda)$ be a real symmetric $n \times n$ matrix polynomial, all of whose elementary divisors are first degree polynomials. Assume further that*

$$\max_{\lambda \in \mathbb{R}} \nu_+(L(\lambda)) + \max_{\lambda \in \mathbb{R}} \nu_-(L(\lambda)) = n.$$

Then $L(\lambda)$ admits a factorization (4.1) *in which D is $n \times n$.*

Proposition 4.2 can be proved by repeating the arguments leading to the proof of Theorem 1 in [GLR2]. We omit the details.

If the hypothesis on the orders of elementary divisors of $L(\lambda)$ is omitted in Theorem 4.1, easy scalar examples (for example, $L(\lambda) = \lambda^2 + 1$) show that the result of Theorem 4.1 is not generally valid. Scalar examples show also that, in this case, the matrices D of minimal size in factorizations (4.1) are not necessarily congruent to each other:

$$\lambda^2 + 1 = [\lambda \quad 1] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda \\ 1 \end{bmatrix} = \left[\tfrac{1}{2}(\lambda^2+2), \tfrac{1}{2}\lambda^2\right] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}(\lambda^2+2) \\ \frac{1}{2}\lambda^2 \end{bmatrix}.$$

We have, however, an upper bound on the minimal size of D.

THEOREM 4.3. *Let $L(\lambda)$ be a real symmetric $n \times n$ matrix polynomial, and let m_0 be defined by* (3.3). *Then for every $m \geq 2 \min(m_0, n)$ $L(\lambda)$ admits a factorization* (4.1).

Proof. Assume first that $m_0 \leq n$. By Theorem 3.1, we have

$$L(\lambda) = \left(M(\bar{\lambda})\right)^* D M(\lambda), \tag{4.3}$$

where D is an $m_0 \times m_0$ constant Hermitian matrix (which can be chosen to be real without loss of generality), and $M(\lambda)$ is a complex matrix polynomial. Write $M(\lambda) = M_1(\lambda) + iM_2(\lambda)$, where $M_1(\lambda)$ and $M_2(\lambda)$ are real matrix polynomials. Then, separating the real part in (4.3), we obtain

$$L(\lambda) = \left[M_1(\lambda)^T \quad M_2(\lambda)^T\right] \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} M_1(\lambda) \\ M_2(\lambda) \end{bmatrix},$$

which is the desired factorization (with $m = 2m_0$).

If $m_0 > n$, use the simple identity:

$$4L(\lambda) = [L(\lambda) + I \quad L(\lambda) - I] \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} L(\lambda) + I \\ L(\lambda) - I \end{bmatrix}. \qquad \square$$

5. Factorization of symmetric real polynomials on the imaginary axis. In this section we consider the case of $n \times n$ matrix polynomials $L(\lambda)$ such that $L(\lambda) = (L(-\lambda))^T$ and $L(\lambda)$ is real for real λ. Note that such a polynomial is selfadjoint on the imaginary axis, i.e., $L(\lambda) = (L(\lambda))^*$ for $\lambda \in i\mathbb{R}$. An immediate consequence of Theorem 3.1 (applied to $L(i\lambda)$) is that such a matrix polynomial admits a factorization

$$L(\lambda) = (M(-\lambda))^T D M(\lambda) \tag{5.1}$$

for a complex $m \times m$ Hermitian matrix D and a complex $m \times n$ matrix polynomial $M(\lambda)$ if and only if

$$m \geq \max_{\lambda \in i\mathbb{R}} \nu_+(L(\lambda)) + \max_{\lambda \in i\mathbb{R}} \nu_-(L(\lambda)).$$

We show in this section that D and $M(\lambda)$ can be taken to be real. Note that there is a contrast here with the situation of §4, where an analogous factorization of a real symmetric matrix polynomial having real factors is not always possible.

First we deal with the case in which $L(\lambda)$ is regular and has constant signature (on the imaginary axis), after which the general case is reduced to the case of constant signature.

In view of Theorem 2.2 (with $F = \mathbb{R}, \sigma = \text{identity}, \varepsilon = -1$), we can restrict our attention to matrix polynomials having only elementary divisors of the form λ or $\lambda^2 + \lambda_0^2$ where λ_0 is real and nonzero. Next, we deal with the case in which L is regular and has constant signature.

THEOREM 5.1. *Suppose that $L(\lambda)$ is a real regular $n \times n$ matrix polynomial satisfying $L(\lambda) = L(-\lambda)^T$ and having constant signature on the imaginary axis: $\nu_+(L(\lambda))$ is constant for all regular points $\lambda \in i\mathbb{R}$. Then L admits a factorization*

$$L(\lambda) = (M(-\lambda))^T D M(\lambda),$$

where M is an $n \times n$ matrix polynomial with real coefficients and D is an $n \times n$ constant real matrix.

Proof. Again by Theorem 2.2 we may assume that $L(\lambda)$ has only pure imaginary eigenvalues and all elementary divisors are linear (in the sense of $\mathbb{C}$). First we deal with the case in which $\lambda^2 + \lambda_0^2, \lambda_0 \in \mathbb{R} \setminus \{0\}$ is an elementary divisor of $L(\lambda)$. Using the Smith form of $L(\lambda)$, write

$$L(\lambda) = E(\lambda) \text{ diag } ((\lambda^2 + \lambda_0^2) p_1(\lambda), \ldots, (\lambda^2 + \lambda_0^2) p_q(\lambda), p_{q+1}(\lambda), \ldots, p_n(\lambda)) F(\lambda),$$

where $p_j(\lambda)$ $(j = 1, \ldots, n)$ are real monic scalar polynomials and $E(\lambda), F(\lambda)$ are real $n \times n$ matrix polynomials with $\det E(\lambda) \equiv \text{const.} \neq 0, \det F(\lambda) \equiv \text{const.} \neq 0$. Then we have for

$$\begin{aligned} &\hat{L}(\lambda) := F(-\lambda)^{-T} L(\lambda) F(\lambda)^{-1}: \\ &\hat{L}(\lambda) = \begin{bmatrix} (\lambda^2 + \lambda_0^2) \hat{B}_{11}(\lambda) & (\lambda^2 + \lambda_0^2) \hat{B}_{12}(\lambda) \\ (\lambda^2 + \lambda_0^2) \hat{B}_{21}(\lambda) & \hat{A}_{22}(\lambda) \end{bmatrix}, \end{aligned} \tag{5.2}$$

where $\hat{B}_{11}$ is a $q \times q$ matrix polynomial. Moreover, $\hat{B}_{11}(\pm i\lambda_0)$ must be invertible, as otherwise $\det \hat{L}(\lambda)$ and hence also $\det L(\lambda)$ would be divisible by $(\lambda^2 + \lambda_0^2)^{q+1}$. Note: If L has constant signature, so has $\hat{L}$, which means $\hat{L}(i\lambda)$ for $\lambda \in \mathbb{R}$ is a Hermitian matrix having constant signature for all real λ except for a finite number of points. Using Rellich's theorem [R], we can write

$$\hat{L}(i\lambda) = (U(\lambda))^* \text{ diag } (\mu_1(\lambda), \ldots, \mu_n(\lambda)) U(\lambda), \qquad \lambda \in \mathbb{R},$$

where $U(\lambda)$ is unitary-valued and analytic and $\mu_j(\lambda)$ is analytic and real. The functions $\mu_j(\lambda)$ have only simple zeros as $\hat{L}$ has only linear elementary divisors (over $\mathbb{C}$). Without loss of generality we may assume that $\mu_1(\lambda_0) = \cdots = \mu_q(\lambda_0) = 0$. By Lemma 6 in [GLR2] we see that q is even, and exactly half of the numbers $\mu_j'(\lambda_0)$ $(j = 1, \ldots, q)$ are positive; the other half are negative. Let u_j be the jth column of $(U(\lambda_0))^*$. Then one calculates

$$\langle \hat{L}'(i\lambda_0) u_j, u_i \rangle = \langle D'(\lambda_0) e_j, e_i \rangle,$$

where $D(\lambda) = \text{diag } (\mu_1(\lambda), \ldots, \mu_n(\lambda))$. Note that $u_1, \ldots, u_q$ span $\ker L(i\lambda_0)$. On $\ker L(i\lambda_0)$, the quadratic form given by $\hat{L}'(i\lambda_0)$ has, therefore, $q/2$ positive squares and $q/2$ negative squares. Now by (5.2) $\ker L(i\lambda_0)$ is span $\{e_1, \ldots, e_q\}$, and for $x, y \in \ker L(i\lambda_0)$, we have

$$\frac{d}{d\lambda} \langle L(i\lambda) x, y \rangle \big|_{\lambda = \lambda_0} = -2\lambda_0 \langle \hat{B}_{11}(i\lambda_0) x, y \rangle.$$

Therefore we conclude that there is an invertible matrix V such that

$$\hat{B}_{11}(i\lambda_0) = V^* \left\{ \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \right\} V, \tag{5.3}$$

where the block $\left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$ is repeated $q/2$ times. Moreover, a simple argument shows that V can be taken to have a real determinant.

Now we state and prove a lemma, after which we return to the proof of Theorem 5.1.

LEMMA 5.2. *Let W be a complex invertible $n \times n$ matrix with real determinant, and let λ_0 be a nonzero real number. Then there exists a real $n \times n$ matrix polynomial $M(\lambda)$ with constant determinant such that $M(i\lambda_0) = W$.*

Proof. We can decompose W as a product of elementary matrices:

$$W = W_1 \cdot W_2 \cdot \cdots W_k, \tag{5.4}$$

where each W_j is either triangular with ones on the diagonal and exactly one nonzero off-diagonal entry, or W_j is a diagonal invertible matrix. Multiplying each diagonal W_j by a suitable complex number α_j so that $\det(\alpha_j W_j)$ is real, we can assume without loss of generality that $\det W_j$ is real ($j = 1, \ldots, k$); here we use the hypothesis that $\det W$ is real. Furthermore, by writing

$$\begin{aligned} W_j = \operatorname{diag}(c_{j1}, \ldots, c_{jn}) = \operatorname{diag}(c_{j1}, \bar{c}_{j1}, 1, \ldots, 1) \\ \cdot \operatorname{diag}\left(1, c_{j2}\bar{c}_{j1}^{-1}, \bar{c}_{j2}c_{j1}^{-1}, 1, \ldots, 1\right) \cdot \operatorname{diag}(1, 1, \ldots, p_{jn}, c_{jn}) \end{aligned}$$

(here $p_{jn} \in \mathbb{C} - \{0\}$), we can assume that every diagonal matrix W_j in (5.4) has real nonzero determinant and at most two diagonal entries different from one (located in adjacent positions). Clearly, it suffices to construct a polynomial $M(\lambda)$ as required such that $M(i\lambda_0) = W_j$ (for a fixed j). If W_j is triangular, let

$$M(\lambda) = \frac{1}{2}\left(W_j + \bar{W}_j\right) + \frac{\lambda}{2\lambda_0} \cdot \frac{1}{i}\left(W_j - \bar{W}_j\right).$$

If $W_j = \operatorname{diag}(1, \ldots, 1, d_1, d_2, 1, \ldots, 1)$ with d_1, d_2 real, then the constant $M(\lambda) \equiv W_j$ will do. Finally, if $W_j = \operatorname{diag}(1, \ldots, 1, d_1, d_2, 1, \ldots, 1)$ with $d_1 d_2 \in \mathbb{R}$, but $d_1 \notin \mathbb{R}$, then put

$$M(\lambda) = \operatorname{diag}\left(1, \ldots, 1, \begin{bmatrix} p(\lambda) & (\lambda^2 + \lambda_0^2) r \\ \lambda^2 + \lambda_0^2 & q(\lambda) \end{bmatrix}, 1, \ldots, 1\right). \tag{5.5}$$

Here

$$\begin{aligned} p(\lambda) &= d_{1R} + \lambda\lambda_0^{-1} d_{1I}, \qquad r = -d_1 d_2 \left(\lambda_0^2 + \left(\lambda_0 d_{1R} d_{1I}^{-1}\right)^2\right)^{-2}, \\ q(\lambda) &= p(\lambda)^{-1}\left[r\lambda^4 + 2r\lambda^2\lambda_0^2 + r\lambda_0^4 + d_1 d_2\right], \end{aligned}$$

where d_{1R} (respectively, d_{1I}) stands for the real (respectively, imaginary) part of d_1. The 2×2 block

$$\begin{bmatrix} p(\lambda) & (\lambda^2 + \lambda_0^2) r \\ \lambda^2 + \lambda_0^2 & q(\lambda) \end{bmatrix}$$

is in the same position in $M(\lambda)$ as

$$\begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}$$

is in W_j. It is easy to verify that $q(\lambda)$ is a real polynomial, $M(\lambda)$ (defined by (5.5)) is a real matrix polynomial with constant nonzero determinant, and $M(i\lambda_0) = W_j$. □

Now let us return to the proof of Theorem 5.1. Let V be as in (5.3), and choose $M(\lambda)$, a real $q \times q$ polynomial with constant nonzero determinant such that $M(i\lambda_0) = V$. This is possible by Lemma 5.2 (recall $\det V$ is real). Now we may replace $\hat{L}(\lambda)$ by

$$\tilde{L}(\lambda) = \begin{bmatrix} M(-\lambda)^{-T} & 0 \\ 0 & I \end{bmatrix} \hat{L}(\lambda) \begin{bmatrix} M(\lambda)^{-1} & 0 \\ 0 & I \end{bmatrix}.$$

Because M has constant nonzero determinant, $\hat{L}$ is a matrix polynomial, and we may write

$$\tilde{L}(\lambda) = \begin{bmatrix} (\lambda^2 + \lambda_0^2) B_{11}(\lambda) & (\lambda^2 + \lambda_0^2) B_{12}(\lambda) \\ (\lambda^2 + \lambda_0^2) B_{21}(\lambda) & A_{22}(\lambda) \end{bmatrix},$$

where

$$B_{11}(i\lambda_0) = \operatorname{diag}(1, -1, \ldots, 1, -1).$$

Now put

$$K(\lambda) = \begin{bmatrix} \lambda & -\lambda_0 \\ \lambda_0 & \lambda \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} \lambda & -\lambda_0 \\ \lambda_0 & \lambda \end{bmatrix} \oplus I_{n-q},$$

where the leading block is repeated $q/2$ times.

We shall show that $N(\lambda) = K(-\lambda)^{-T} \tilde{L}(\lambda) K(\lambda)^{-1}$ is a matrix polynomial. Note that $N(\lambda)$ may have a pole only at $\pm i\lambda_0$, and it suffices to show it has no pole at either one of these points. Moreover, any pole of $N(\lambda)$ must appear in its leading $q \times q$ block. This leading $q \times q$ block equals

$$(\lambda^2 + \lambda_0^2)^{-1} \left(\begin{bmatrix} -\lambda & -\lambda_0 \\ \lambda_0 & -\lambda \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} -\lambda & -\lambda_0 \\ \lambda_0 & -\lambda \end{bmatrix} \right) B_{11}(\lambda) \left(\begin{bmatrix} \lambda & \lambda_0 \\ -\lambda_0 & \lambda \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} \lambda & \lambda_0 \\ -\lambda_0 & \lambda \end{bmatrix} \right)$$

as an easy computation shows. Now at $\lambda = i\lambda_0$, we have

$$\begin{bmatrix} -i\lambda_0 & -\lambda_0 \\ \lambda_0 & -i\lambda_0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} i\lambda_0 & \lambda_0 \\ -\lambda_0 & i\lambda_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Recalling that

$$B_{11}(i\lambda_0) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

we see that $N(\lambda)$ is a matrix polynomial. Taking determinants we see that $N(\lambda)$ has no eigenvalue at $\pm i\lambda_0$. Applying the same argument at each nonzero, singular point of $L(\lambda)$, we reduce the proof of Theorem 5.1 to the case in which zero is the only possible singular point of $L(\lambda)$. However, for that case a similar argument shows that $L(\lambda)$ admits a representation $L(\lambda) = (K(-\lambda))^T N(\lambda) K(\lambda)$ with $K(\lambda)$ a real matrix polynomial and $N(\lambda)$ a real matrix polynomial without singular points (cf. the proof of Proposition 3.4 given in [GLR2]).

We have finally reduced to the case where L has no singular points. In this case the result follows from Theorem 2.1. □

Next we state the main result of this section.

THEOREM 5.3. *Let $L(\lambda) = (L(-\lambda))^T$ be an $n \times n$ matrix polynomial with real coefficients. Then there is a real $m \times m$ matrix $D = D^T$ and an $m \times n$ matrix polynomial $M(\lambda)$ with real coefficients such that*

$$L(\lambda) = (M(-\lambda))^T D M(\lambda) \tag{5.6}$$

if and only if

$$m \geq m_0 := \max_{\lambda \in i\mathbb{R}} \nu_+(L(\lambda)) + \max_{\lambda \in i\mathbb{R}} \nu_-(L(\lambda)). \tag{5.7}$$

Moreover, when $m = m_0$, the matrix D is unique up to congruence by a real orthogonal matrix.

Analogous to the situation in Theorem 3.1, the matrix polynomial $M(\lambda)$ in Theorem 5.3 can be chosen with additional spectral properties. Given a polynomial $L(\lambda)$ as in Theorem 5.3, a set Λ of numbers with nonzero real parts is called a *d-set* (with respect to $L(\lambda)$) if Λ is a maximal set of singular points of $L(\lambda)$ with nonzero real parts having the property that $\lambda \in \Lambda \Longrightarrow \bar{\lambda} \in \Lambda, -\bar{\lambda} \notin \Lambda$. (A d-set may be empty.) It turns out that under the hypotheses of Theorem 5.3, for every given d-set Λ, there exists a factorization (5.6) where $D = D^T$ is $m_0 \times m_0$ and where the set of nonreal singular points of $M(\lambda)$ coincides with Λ. This follows as a by-product of the proof of Theorem 5.3 (including Theorem 5.1 and Theorem 2.2 with $F = \mathbb{R}, \sigma = \text{identity}, \varepsilon = -1$).

Proof. The uniqueness of D is verified as in the proof of Theorem 3.1. The fact that $m \geq m_0$ is necessary for the existence of real $D = D^T$ and $M(\lambda)$ such that (5.6) is satisfied is seen as in the proof of Theorem 3.1. It remains to prove sufficiency. We may reduce to the regular case again as in §3. In case $L(\lambda)$ has constant signature we are finished, using Theorem 5.1. In case $L(\lambda)$ does not have constant signature on the imaginary axis, it is shown that there exists a real $m_0 \times m_0$ matrix polynomial $\tilde{L}(\lambda)$ such that $\tilde{L}(\lambda) = \tilde{L}(-\lambda)^T$ and

$$\tilde{L}(\lambda) = \begin{bmatrix} L(\lambda) & 0 \\ 0 & * \end{bmatrix},$$

while $\tilde{L}$ is regular and has constant signature on the imaginary axis. Indeed, as $L(i\lambda)$ is selfadjoint for real λ, we can write (using Rellich's theorem [R]; also [GLR3])

$$L(i\lambda) = (U(\lambda))^* \operatorname{diag}(\mu_1(\lambda), \ldots, \mu_n(\lambda)) U(\lambda),$$

where $\mu_j(\lambda)$ is analytic and real valued and $U(\lambda)$ is analytic and unitary. Since $L(i\lambda) = \overline{L(-i\lambda)}, \lambda \in \mathbb{R}$, the matrices $L(i\lambda)$ and $L(-i\lambda)$ have the same eigenvalues, and therefore for every point $\lambda_0 \in \mathbb{R}$ there is a permutation σ on $\{1, \ldots, n\}$ such that

$$\mu_i(-\lambda) = \mu_{\sigma(i)}(\lambda) \quad (i = 1, \ldots, n) \tag{5.8}$$

in a neighborhood of λ_0.

Let $\lambda_0 \in \mathbb{R}$ be such that $\pm i\lambda_0$ are singular points of $L(\lambda)$. Define $\Omega(\lambda_0)$ and $q(\lambda_0)$ as in the proof of Lemma 3.5. It follows from (5.8) that $q(\lambda_0) = -q(-\lambda_0)$; in particular, $q(0) = 0$ (if $\lambda_0 = 0$ is a singular point of $L(\lambda)$). Furthermore (analogous to (3.8) and (3.9))

$$\begin{aligned} m_0 &= \max_{\lambda > 0} \nu_+(L(i\lambda)) + \max_{\lambda > 0} \nu_-(L(i\lambda)) \\ &= n + \max |\nu_+(L(i\lambda_1)) - \nu_+(L(i\lambda_2))| \end{aligned}$$

and

$$\max |\nu_+(L(i\lambda_1)) - \nu_+(L(i\lambda_2))| = \max_{\lambda_1 < \lambda_2} \left| \sum_{\lambda_1 < \lambda_0 < \lambda_2} q(\lambda_0) \right|, \tag{5.9}$$

where the summation is over all $\lambda_0 \in (\lambda_1, \lambda_2)$ such that $i\lambda_0$ is a singular point of L. (Here the real numbers λ_1 and λ_2 are such that $i\lambda_1$ and $i\lambda_2$ are regular points of L.) Denote the number (5.9) by p, as in the proof of Lemma 3.5. Now construct p polynomials $r_1(\lambda), \dots, r_p(\lambda)$ with real coefficients having the following properties:

(i) $r_j(\lambda) = r_j(-\lambda)$ is real for $\lambda \in \mathbb{R}$;

(ii) all zeros of r_j are pure imaginary, nonzero numbers and belong to the set S of pure imaginary singular points λ_0 of L for which $q(\lambda_0) \neq 0$;

(iii) for every $\lambda_0 \in S$ exactly $|q(\lambda_0)|$ polynomials among $r_1, \dots, r_p$ have λ_0 as a zero and for each r_j having λ_0 as a zero we have

$$q(\lambda_0) \frac{d}{d\lambda} r_j(i\lambda)\Big|_{\lambda=\lambda_0} < 0.$$

(Note that because of (i) and $q(-\lambda_0) = -q(\lambda_0)$, condition (iii) is satisfied at $-\lambda_0$ if it is satisfied at λ_0.) Put

$$\tilde{L}(\lambda) = L(\lambda) \oplus \operatorname{diag}(r_1(\lambda), \dots, r_p(\lambda)).$$

Then $\tilde{L}(\lambda)$ is regular and has constant signature on the imaginary axis, as desired. Thus, by Theorem 5.1, $\tilde{L}(\lambda)$ admits a factorization

$$\tilde{L}(\lambda) = (N(-\lambda))^T D N(\lambda)$$

with D an $m_0 \times m_0$ real matrix and $N(\lambda)$ an $m_0 \times n$ real matrix polynomial. Taking for $M(\lambda)$ the matrix polynomial formed by the first n columns of N now finishes the proof. □

Analogous to Theorem 3.2, the result of Theorem 5.3 can be put in terms of additive representations of $L(\lambda)$ via elementary matrix polynomials. Here, a real $n \times n$ matrix polynomial $M(\lambda)$ will be called *elementary* if $r(M) = 1$ and $M(\lambda)$ is positive semidefinite Hermitian for all $\lambda \in i\mathbb{R}$.

THEOREM 5.4. *Let $L(\lambda)$ be as in Theorem 5.3. Then $L(\lambda)$ admits a representation*

$$L(\lambda) = \sum_{j=1}^{q} \varepsilon_j M_j(\lambda), \tag{5.10}$$

where $\varepsilon_j = \pm 1$ and $M_j(\lambda)$ are elementary matrix polynomials, if and only if $q \geq m_0$, where m_0 is defined by (5.7). *Moreover, when $q = m_0$, exactly $\max_{\lambda_0 \in i\mathbb{R}} \nu_+(L(\lambda_0))$ of the ε_j's in* (5.10) *are equal to $+1$, and exactly $\max_{\lambda \in i\mathbb{R}} \nu_-(L(\lambda))$ of them are equal to -1.*

We omit the easy derivation of Theorem 5.4 from Theorem 5.3.

6. Factorization of complex symmetric polynomials. In this section we consider $n \times n$ matrix polynomials $L(\lambda)$ with complex coefficients having the symmetry

$$L(\lambda) = (L(\varepsilon\lambda))^T, \tag{6.1}$$

where $\varepsilon = \pm 1$ is fixed, and their factorizations of the form

$$L(\lambda) = (M(\varepsilon\lambda))^T D M(\lambda), \tag{6.2}$$

where $M(\lambda)$ is an $m \times n$ matrix polynomial (over $\mathbb{C}$), and D is a constant complex symmetric matrix. Observe that every $m \times m$ complex symmetric matrix D can be factored as $D = V^T V$ for some complex matrix V (see, e.g., Corollary 4.4.6 in [HJ]). Therefore, we may assume that $D = I$ in (6.2).

Here (in contrast with §§3–5) signatures of Hermitian matrices do not play a role.

We start with the case where $\varepsilon = -1$.

THEOREM 6.1. *Let $L(\lambda)$ be an $n \times n$ matrix polynomial satisfying* (6.1), *where $\varepsilon = -1$. Then the minimal size m for which $L(\lambda)$ admits a factorization*

$$L(\lambda) = (M(-\lambda))^T M(\lambda) \tag{6.3}$$

with an $m \times n$ matrix polynomial $M(\lambda)$ is equal to the general rank r of $L(\lambda)$.

Proof. We use the same ideas as in the proofs of results in the previous sections. Therefore, the proof of Theorem 6.1 is presented with less detail.

Clearly, a factorization (6.3) is impossible if $m < r$. Therefore we have to prove only that such a factorization exists for $m = r$. We can (and do) assume that $n = r$, i.e., $\det L(\lambda) \not\equiv 0$.

Apply Theorem 2.2 with $F = \mathbb{C}, \sigma = \text{identity}, \varepsilon = -1$. Since the only irreducible monic complex polynomial f satisfying $f = \varepsilon^{\text{degree } f} f_*$ is $f(\lambda) = \lambda$, by Theorem 2.2 we can assume (replacing $L(\lambda)$ by $D(\lambda)$) that the elementary divisors of $L(\lambda)$ are $\lambda, \lambda, \ldots, \lambda$ (k times). Here k is necessarily even. Indeed, the property $L(\lambda) = (L(-\lambda))^T$ ensures that $\det L(\lambda) = (\text{const.})\lambda^k$ is an even function.

If $k = 0$, i.e., $\det L(\lambda) \equiv \text{const.} \neq 0$, an application of Theorem 2.1 gives the desired result. Suppose therefore that $k > 0$. Using the Smith form of $L(\lambda)$, write

$$L(\lambda) = E(\lambda) \begin{bmatrix} \lambda I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} F(\lambda),$$

where $E(\lambda)$ and $F(\lambda)$ are $n \times n$ matrix polynomials with constant nonzero determinants. Replacing $L(\lambda)$ by $(F(-\lambda))^{-T} L(\lambda) F(\lambda)^{-1}$, we can assume that the first k columns (and, by symmetry, also the first k rows) of $L(\lambda)$ are divisible by λ. Thus

$$L(\lambda) = \begin{bmatrix} \lambda L_1(\lambda) & \lambda L_2(\lambda) \\ -\lambda (L_2(-\lambda))^T & L_3(\lambda) \end{bmatrix},$$

where the matrix polynomials L_1, L_2, and L_3 are $k \times k, k \times (n-k)$, and $(n-k) \times (n-k)$, respectively. Moreover, $-L_1(-\lambda) = (L_1(\lambda))^T$ and $L_3(-\lambda) = (L_3(\lambda))^T$. We claim that $L_1(0)$ is invertible. Indeed, if $L_1(0)$ were not invertible, then

$$\det \begin{bmatrix} L_1(\lambda) & L_2(\lambda) \\ -\lambda (L_2(-\lambda))^T & L_3(\lambda) \end{bmatrix}$$

would be divisible by λ, and consequently

$$\det L(\lambda) = \lambda^k \det \begin{bmatrix} L_1(\lambda) & L_2(\lambda) \\ -\lambda (L_2(-\lambda))^T & L_3(\lambda) \end{bmatrix}$$

would be divisible by λ^{k+1}, which is an impossibility. Now $L_1(0)$ is skew-symmetric and therefore admits a factorization

$$L_1(0) = Q^T \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right) Q$$

for some invertible matrix Q. Let

$$M_1(\lambda) = \left(\begin{bmatrix} -\lambda & 0 \\ 0 & 1 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} -\lambda & 0 \\ 0 & 1 \end{bmatrix} \oplus I_{n-k}\right) \begin{bmatrix} Q & 0 \\ 0 & I_{n-k} \end{bmatrix} \tag{6.4}$$

(the summand $\left[\begin{smallmatrix} -\lambda & 0 \\ 0 & 1 \end{smallmatrix}\right]$ is repeated $k/2$ times). Then

$$L(\lambda) = (M_1(-\lambda))^T \tilde{L}(\lambda) M_1(\lambda)$$

for some matrix polynomial $\tilde{L}(\lambda)$ such that $\tilde{L}(-\lambda) = (\tilde{L}(\lambda))^T$ and $\det \tilde{L}(\lambda) \equiv \text{const.} \neq 0$. The only thing not immediate here is the claim that $\tilde{L}(\lambda)$ is indeed a polynomial. But the only point in $\mathbb{C}$ where $\tilde{L}(\lambda)$ could have a pole is $\lambda_0 = 0$. We have

$$\begin{aligned} Z(\lambda) &:= (M_1(-\lambda))^{-T} L(\lambda) M_1(\lambda)^{-1} \\ &= \left(\begin{bmatrix} \lambda^{-1} & 0 \\ 0 & 1 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} \lambda^{-1} & 0 \\ 0 & 1 \end{bmatrix} \oplus I_{n-k}\right) \begin{bmatrix} Q^{-T} & 0 \\ 0 & I_{n-k} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \lambda L_1(\lambda) & \lambda L_2(\lambda) \\ -\lambda (L_2(-\lambda))^T & L_3(\lambda) \end{bmatrix} \begin{bmatrix} Q^{-1} & 0 \\ 0 & I_{n-k} \end{bmatrix} \\ &\quad \cdot \left(\begin{bmatrix} -\lambda^{-1} & 0 \\ 0 & 1 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} -\lambda^{-1} & 0 \\ 0 & 1 \end{bmatrix} \oplus I_{n-k}\right). \end{aligned}$$

Clearly, $\lambda Z(\lambda)$ is analytic at $\lambda_0 = 0$, and the coefficient of λ^{-1} in the Laurent series of $Z(\lambda)$ in a neighborhood of $\lambda_0 = 0$ is

$$\begin{aligned} &\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \oplus 0\right) \begin{bmatrix} Q^{-T} & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} L_1(0) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q^{-1} & 0 \\ 0 & I_{n-k} \end{bmatrix} \\ &\quad \cdot \left(\begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \oplus 0\right) \\ &\qquad = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \oplus 0 = 0. \end{aligned}$$

To finish the proof, apply Theorem 2.1 to $Z(\lambda)$. □

Finally, we consider matrix polynomials $L(\lambda)$ having the symmetry (6.1) with $\varepsilon = 1$.

THEOREM 6.2. *Let $L(\lambda) = L(\lambda)^T$ be an $n \times n$ matrix polynomial over $\mathbb{C}$, and let r be the general rank of $L(\lambda)$. If the product of the invariant polynomials of $L(\lambda)$ is a square of a complex polynomial, then $L(\lambda)$ admits a factorization*

$$L(\lambda) = (M(\lambda))^T M(\lambda) \tag{6.5}$$

for some $m \times n$ matrix polynomial $M(\lambda)$ if and only if $m \geq r$. If the product of invariant polynomials of $L(\lambda)$ is not a square of any complex polynomial, then $L(\lambda)$ admits a factorization (6.5) *if and only if $m \geq r + 1$.*

Proof. Again, we omit many details here. We assume that $r = n$. If $L(\lambda)$ admits a factorization (6.5) with $M(\lambda)$ $n \times n$, then $\det L(\lambda) = (\det M(\lambda))^2$, and so the product of invariant polynomials of $L(\lambda)$ must be a square as well. This implies the only if part.

To prove the if part, first observe that it suffices to consider only the case in which $\det L(\lambda)$ is the square of a polynomial (if it is not, replace $L(\lambda)$ by $\left[\begin{smallmatrix} L(\lambda) & 0 \\ 0 & f(\lambda) \end{smallmatrix}\right]$, where

$f(\lambda)$ is a scalar polynomial chosen so that $f(\lambda)$ det $L(\lambda)$ is a square). By Theorem 2.2 we may assume that all elementary divisors of $L(\lambda)$ are first-degree polynomials. Since det $L(\lambda)$ is a square, the number $k = k(a)$ of elementary divisors $\lambda - a, \ldots, \lambda - a$ of $L(\lambda)$ (where $a \in \mathbb{C}$ is fixed) is even. As in the proof of Theorem 6.1, we can further assume that

$$L(\lambda) = \begin{bmatrix} (\lambda - a) L_1(\lambda) & (\lambda - a) L_2(\lambda) \\ (\lambda - a) (L_2(\lambda))^T & L_3(\lambda) \end{bmatrix}$$

for some matrix polynomials $L_1(\lambda) = L_1(\lambda)^T, L_2(\lambda)$, and $L_3(\lambda) = L_3(\lambda)^T$ of sizes $k \times k, k \times (n-k)$, and $(n-k) \times (n-k)$, respectively. Moreover, $L_1(a)$ is invertible and symmetric, and therefore

$$L_1(a) = Q^T \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \oplus \cdots \oplus \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right) Q$$

for some invertible matrix Q (the direct summand $\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ is repeated here $k/2$ times). As in the proof of Theorem 6.1, we verify that

$$L(\lambda) = (M_1(\lambda - a))^T \tilde{L}(\lambda) M_1(\lambda - a),$$

where $M_1(\lambda)$ is defined by (6.4), and $\tilde{L}(\lambda)$ is a matrix polynomial such that $\tilde{L}(\lambda) = (\tilde{L}(\lambda))^T$ and $\tilde{L}(\lambda)$ has no elementary divisors of the form $\lambda - a$. Apply the above procedure to $\tilde{L}(\lambda)$ in place of $L(\lambda)$, using elementary divisors $\lambda - b, \ldots, \lambda - b$ of $\tilde{L}(\lambda)$ for some $b \in \mathbb{C}$, and so on, until a matrix polynomial $L_1(\lambda) = (L_1(\lambda))^T$ with det $L_1(\lambda) \equiv$ const. $\neq 0$ is obtained. Now apply Theorem 2.1 to get the desired factorization of $L_1(\lambda)$. □

Theorems 6.1 and 6.2 can be recast in terms of elementary matrices (analogous to Theorem 3.2). An $n \times n$ matrix polynomial $M(\lambda)$ (over $\mathbb{C}$) is called ε-elementary if

$$M(\lambda) = (x(\varepsilon\lambda))^t x(\lambda)$$

for some $1 \times n$ row polynomial $x(\lambda) \neq 0$ (here $\varepsilon = \pm 1$ is fixed).

THEOREM 6.3. *Let $L(\lambda)$ be an $n \times n$ matrix polynomial satisfying* (6.1), *and let r be the general rank of $L(\lambda)$. Then $L(\lambda)$ can be written as a sum of r ε-elementary matrix polynomials, unless $\varepsilon = 1$ and the product of the elementary divisors of $L(\lambda)$ is not a square of polynomial. In this latter case, $L(\lambda)$ can be written as a sum of $r+1$ 1-elementary matrix polynomials, and it cannot be represented as a sum of any r 1-elementary matrix polynomials.*

Acknowledgment. The problem of finding the minimal possible size of D in factorization (1.1) for complex selfadjoint matrix polynomials has been posed by Professor I. Gohberg.

REFERENCES

[AM] B. D. O. ANDERSON AND J. B. MOORE, *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[Co] W. A. COPPEL, *Linear systems*, Notes on Pure Math., 6, Australian National University, Canberra, 1972.

[GLR1] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Spectral analysis of selfadjoint matrix polynomials*, Ann. Math., 112 (1980), pp. 33–71.

[GLR2] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Factorization of selfadjoint matrix polynomials with constant signature*, Linear and Multilinear Algebra, 11 (1982), pp. 209–224.

[GLR3] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrix Polynomials*, Academic Press, New York, 1982.

[HJ] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[Ja] V. A. JAKUBOVIČ, *Factorization of symmetric matrix polynomials*, Soviet Math. Doklady 11 (1970), pp. 1261–1264.

[Lyu1] B. D. LYUBACHEVSKII, *Factorization of symmetric matrices with elements from a ring with involution* I, Siberian Math. J., 14 (1973), pp. 233–246.

[Lyu2] ———, *Factorization of symmetric matrices with elements from a ring with involution* II, Siberian Math. J., 14 (1973), pp. 423–433.

[M] C. C. MACDUFFEE, *The Theory of Matrices*, Chelsea Publications Company, New York, 1946.

[R] R. RELLICH, *Perturbation Theory for Eigenvalue Problems*, Gordon & Breach, New York, 1969.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 865–880, July 1994

010

DECOMPOSABILITY AND QUOTIENT SUBSPACES FOR THE PENCIL $sL - M$*

V. L. SYRMOS† AND F. L. LEWIS‡

Abstract. This paper introduces the notion of *decomposability* of the domain and the codomain relative to the generalized nonsquare matrix pencil $\Pi(s) = sL - M$. Its importance is justified rigorously and it is demonstrated that a special case of these new results is the familiar notion of decomposability for the pencil $sI - M$. This definition is motivated by the concepts of strict equivalence and quotient subspaces. By decomposing both the domain and the codomain of L and M the close relation between decomposability and the Kronecker invariants of the pencil $sL - M$ is shown. Finally, an application of the notion of decomposabilty in controls design is presented.

Key words. matrix pencil theory, decomposability, generalized Sylvester equations

AMS subject classifications. 15A04, 15A06, 15A22

1. Introduction.

The notion of decomposability of the domain, which actually is precisely the same as the codomain, for the map $sI - M$ was presented in [6], [19] from two different perspectives. In [6] it was presented in the spirit of revealing how the finite elementary divisors of the pencil $sI - M$ are related to the concept of decomposability. On the other hand, in [19] the main motive was to show how this property can be exploited to develop some geometric controls design techniques. Although the motives were different the goal was the same, that is, to exhibit the significance of decomposability to the matrix pencil theory as well as to control theory.

In this paper we present the notion of decomposabilty of both the *domain* and the *codomain* of the map $sL - M$. The introduced geometric theory extends the concept of decomposability to the general nonsquare pencil $sL - M$. Moreover, it finds applications to the Kronecker structure of the map $sL - M$ and to control system theory.

Motivated by the work in [6], [19] for decomposability and the matrix pencil theory [1], [6], [7], [17], in this paper we introduce the notion of decomposability of the domain and the codomain of the map $\Pi(s) = sL - M$, where $sL - M$ is generally nonsquare. The first major difference between the traditional case where $\Pi(s) = sI - M$ and the proposed one is that the domain and the codomain are not the same. As a result, the notion of similarity (e.g., $T(sI - M)T^{-1}$) is replaced by the notion of strict equivalence (e.g., $U(sL - M)V$). Our first result exhibits the direct relationship between the concepts of strict equivalence and quotient subspaces.

The definition of decomposability for $(sL - M)$ requires a nontraditional decomposition of both the domain and the codomain. This leads us to the key theorem of the paper, which presents necessary and sufficient conditions for decomposabilty. Moreover, further elaboration of these conditions results in two *generalized Sylvester* equations. It has been recently shown in the literature that the generalized Sylvester

*Received by the editors November 13, 1989; accepted for publication (in revised form) December 11, 1992. This research was supported by National Science Foundation grant ECS–8805932 and by the Alexander Onassis Foundation GROUP L–89.

†Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, Hawaii 96822 (hellas@euclid.eng.hawaii.edu).

‡Automation and Robotics Research Institute, The University of Texas, Arlington, Fort Worth, Texas 76118.

equations proposed in this paper can be solved in a computationally stable manner using generalized Schur methods with condition estimators [8]. Throughout our exploration we show that the results of [6], [19] may be recovered as a special case of our results. Finally, we present a relation of decomposabilty to the Kronecker structure of the pencil $sL - M$.

The above remarks demonstrate the significance of decomposability for the pencil $sL - M$, but they do not exhibit its application to control theory. In order to show how the proposed notion of decomposability can find applications to control theory we will refer to the importance of singular systems, which have attracted the interest of researchers in control theory [3], [15]. Singular systems involve the study of the generalized pencil $sL - M$. Therefore, the decomposability condition is a useful tool in design for singular systems. Specifically, in [12] a sufficient condition is presented for the solution of the problem of proportional feedback, which under thorough consideration is seen to be a sufficient condition of decomposability. Moreover, as in the state-variable case, for the output-regulation problem decomposability was of primary importance [19]; it is also expected that decomposability will be of great importance for the same problem in singular systems.

Finally, in order to further demonstrate the usefulness of decomposability of generalized pencils, we refer to the fact that state-variable systems can always be transformed to an equivalent nonsquare system [7], which is easier to study. In order to show the importance of decomposability in control applications, in §4 we explore the closed-loop eigenstructure problem with state feedback in a computationally stable fashion, under the setting of nonsquare descriptions. In addition, in the proposed technique we show how the reduced-order nonsquare system implies more computationally stable algorithms. Consequently, classical design problems that have already been solved can be reconsidered under the notion of nonsquare systems. In that case the decomposability condition will certainly find applications that will exploit its most general form.

2. The notion of decomposability for the pencil $sL - M$. Consider the pencil

$$\Pi(s) = [sL - M], \tag{2.1}$$

where $sL - M \in \Re^{l\times n}[s]$ is generally nonsquare and not necessarily of full rank. We shall represent the domain $\Re^n$ of $(sL - M)$ by $\mathcal{X}$ and its codomain $\Re^l$ by $\mathcal{Z}$. If the pencil (2.1) is square and

$$\Delta(s) = \det(sL - M) \not\equiv 0 \quad \text{for some } s, \tag{2.2}$$

then it will be called *regular*. In any other case it will be called *singular*. If $(sL - M)$ is regular, those isolated values of s where (2.2) fails to hold will be said to comprise the *spectrum* of $(sL - M)$, denoted $\sigma(L, M)$. The usual spectrum of $(sI - M)$ will be denoted $\sigma(M)$. Note that $\sigma(L, M)$ may contain finite and infinite values of s.

Define two pencils $sL - M$ and $s\tilde{L} - \tilde{M}$ of dimension $l \times n$ to be *strictly equivalent* [6], [17], [18] when there exist nonsingular constant matrices U and V of orders l and n respectively such that

$$U(sL - M)V = s\tilde{L} - \tilde{M}. \tag{2.3}$$

If $\mathcal{S} \subset \mathcal{X}$, $x \in \mathcal{X}$, define the equivalence class $\bar{x} = \{y \in \mathcal{X} : x - y \in \mathcal{S}\}$ and the *quotient* space $\mathcal{X}/\mathcal{S}$ as the set of all $\bar{x}$. Then the *canonical* projection $P : \mathcal{X} \to \mathcal{X}/\mathcal{S}$ is defined by $Px = \bar{x}$. We may also write $\bar{x}$ as $x + \mathcal{S}$. See [19].

In the next result, we are motivated by the fact that the equivalence of matrix pencils is defined in terms of two constant nonsingular maps, one acting on the domain and one acting on the codomain. This is in contrast to the case of a single matrix operator, where similarity is defined in terms of the same matrix in both the domain and the codomain (e.g., $T(sI - M)T^{-1}$).

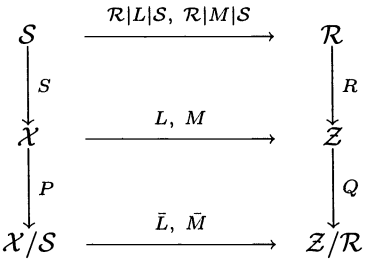
The result is that for $sL - M$ the domain $\mathcal{X}$ and the codomain $\mathcal{Z}$ must be considered as separate spaces. The definition of strict equivalence motivates the next lemma, which is key for the theory presented in this paper. As will be shown, this lemma relates the algebraic condition for equivalence of pencils ($P[sL - M]Q = [s\tilde{L} - \tilde{M}]$) to the geometric approach.

LEMMA 2.1. *Let $\mathcal{S} \subset \mathcal{X}$ and $\mathcal{R} \subset \mathcal{Z}$, such that $L\mathcal{S} + M\mathcal{S} \subset \mathcal{R}$ where $L, M : \mathcal{X} \to \mathcal{Z}$ and $sL - M \in \mathfrak{R}^{l\times n}[s]$ is not necessarily regular. Let P and Q be the canonical projections $P : \mathcal{X} \to \mathcal{X}/\mathcal{S}$ and $Q : \mathcal{Z} \to \mathcal{Z}/\mathcal{R}$. Then there exist unique maps $\bar{L}$ and $\bar{M}$ such that*

$$\bar{L}P = QL, \tag{2.4}$$

$$\bar{M}P = QM. \tag{2.5}$$

That is, the following diagram commutes



where R and S are the insertion maps of $\mathcal{R}$ in $\mathcal{X}$ and $\mathcal{S}$ in $\mathcal{Z}$ respectively.

Proof. Let $\mathcal{T}$ be any complement of $\mathcal{S}$ in $\mathcal{X}$ so that $\mathcal{X} = \mathcal{T} \oplus \mathcal{S}$. Choose $\{t_i\}_{i=1}^{\tau}$ to be a basis for $\mathcal{T}$. Then, if $\bar{t}_i = Pt_i$, a basis for $\mathcal{X}/\mathcal{S}$ is $\{\bar{t}_i\}_{i=1}^{\tau}$, where $\tau = \dim \mathcal{T}$. Define $\bar{L}$ and $\bar{M}$ by

$$\bar{L}\bar{t}_i = QLt_i, \tag{2.6}$$

$$\bar{M}\bar{t}_i = QMt_i. \tag{2.7}$$

To show that $\bar{L}$ is well defined, suppose $\bar{x}_1, \bar{x}_2 \in \mathcal{X}/\mathcal{S}$ with $\bar{x}_1 = \bar{x}_2$. Then $\bar{x}_1 = x_1 + \mathcal{S}$ and $\bar{x}_2 = x_2 + \mathcal{S}$ for some $x_i \in \mathcal{T}$, and $x_1 + \mathcal{S} = x_2 + \mathcal{S}$, or $x_1 - x_2 \in \mathcal{S}$. Thus $QL(x_1 - x_2) \in QL\mathcal{S} \subset Q\mathcal{R} = 0$. Therefore $QLx_1 = QLx_2$.

Now let $x \in \mathcal{X}/\mathcal{S}$ and $x = t + s$ with $t \in \mathcal{T}$, $s \in \mathcal{S}$. Then $QLx = QL(t + s)$. But $QL\mathcal{S} \in Q\mathcal{R} = 0$, therefore by (2.16) $QLx = QLt = \bar{L}\bar{t} = \bar{L}\bar{P}(t + s)$, which verifies (2.6). Similarly we can prove that $\bar{M}$ is well defined. Compare this proof with that in [13]. □

At this point we introduce some notation that we will use later. The induced maps will be denoted as $L_1 = \bar{L} : \mathcal{X}/\mathcal{S} \to \mathcal{Z}/\mathcal{R}$ and $M_1 = \bar{M} : \mathcal{X}/\mathcal{S} \to \mathcal{Z}/\mathcal{R}$. We will denote as $L_2 = \mathcal{R}|L|\mathcal{S}$, $M_2 = \mathcal{R}|M|\mathcal{S}$, Also $\dim\mathcal{X} = n$, $\dim\mathcal{Z} = l$, $\dim\mathcal{S} = \sigma$, $\dim\mathcal{X}/\mathcal{S} = n - \sigma = s$, $\dim\mathcal{R} = \rho$, and $\dim\mathcal{Z}/\mathcal{R} = l - \rho = r$. In a suitable basis representation for $\mathcal{S}$ and $\mathcal{R}$, the matrices L and M can be written

$$\begin{pmatrix} L_1^{r\times s} & 0^{r\times\sigma} \\ L_3^{\rho\times s} & L_2^{\rho\times\sigma} \end{pmatrix}, \qquad \begin{pmatrix} M_1^{r\times s} & 0^{r\times\sigma} \\ M_3^{\rho\times s} & M_2^{\rho\times\sigma} \end{pmatrix}. \tag{2.8}$$

Note that in the case where the matrices L and M are square and $L = I$ then Lemma 2.1 boils down to the well-known theory in [19].

DEFINITION 2.2. *Let $\mathcal{S} \subset \mathcal{X}$, $\mathcal{R} \subset \mathcal{Z}$ and $L\mathcal{S} + M\mathcal{S} \subset \mathcal{R}$. If there exists a subspace $\mathcal{T}$ such that $L\mathcal{T} + M\mathcal{T} \subset \mathcal{V}$, $\mathcal{X} = \mathcal{S} \oplus \mathcal{T}$, and $\mathcal{Z} = \mathcal{R} \oplus \mathcal{V}$, then $\mathcal{S}$, $\mathcal{R}$ decompose $\mathcal{X}$, $\mathcal{Z}$ relative to L, M.*

Definition 2.2 means that there are coordinate system transformations such that L_3 and M_3 are zero. It is rather interesting to see how this definition is expressed in some special cases. Assume that $\mathcal{S}$ is an (L, M)–invariant subspace, that is, $L\mathcal{S} \subset M\mathcal{S}$. Then the decomposability condition takes the following form. Let $L\mathcal{S} \subset M\mathcal{S} \subset \mathcal{R}$; if there exist subspaces $\mathcal{T}$ and $\mathcal{V}$ such that $L\mathcal{T} \subset M\mathcal{T} \subset \mathcal{V}$, $\mathcal{X} = \mathcal{S} \oplus \mathcal{T}$, and $\mathcal{Z} = \mathcal{R} \oplus \mathcal{V}$, then $\mathcal{S}$, $\mathcal{R}$ decompose $\mathcal{X}$, $\mathcal{Z}$ relative to L, M.

Assuming in addition that the pencil is regular and $L = I$ gives the same results as those in [19]. Indeed, for the case where $L = I$, the condition $L\mathcal{S} + M\mathcal{S} \subset \mathcal{R}$ in Definition 2.2 becomes $\mathcal{S} + M\mathcal{S} \subset \mathcal{R}$. This last relation implies $\mathcal{S} \subset \mathcal{R}$. As a result there exists a subspace $\mathcal{S}_1$ such that $\mathcal{R} = \mathcal{S} \oplus \mathcal{S}_1$. Similarly, there exists a subspace $\mathcal{T}_1$ such that $\mathcal{V} = \mathcal{T} \oplus \mathcal{T}_1$. By adding these two equations and taking into consideration the fact that in this case the domain is the same as the codomain, that is, $\mathcal{X} = \mathcal{R} \oplus \mathcal{V} = \mathcal{S} \oplus \mathcal{T}$, it follows that $\mathcal{S} = \mathcal{R}$ and $\mathcal{T} = \mathcal{V}$. Finally under this argument the conditions $L\mathcal{S} + M\mathcal{S} \subset \mathcal{R}$ and $L\mathcal{T} + M\mathcal{T} \subset \mathcal{V}$ boil down to the conditions $M\mathcal{S} \subset \mathcal{S}$ and $M\mathcal{T} \subset \mathcal{T}$, so the definition of decomposability coincides with that presented in [19].

We now close this small digression of special cases and return to our general discussion. The definition of decomposability simply states that selecting $\mathcal{T}$, $\mathcal{V}$ as the complements for $\mathcal{S}$, $\mathcal{R}$, we have that $L_1 = \mathcal{V}|L|\mathcal{T}$ and $M_1 = \mathcal{V}|M|\mathcal{T}$. It is natural now to draw connections between Lemma 2.1 and Definition 2.2 relative to the pencil $sL - M$. These connections are presented through the proof of the next theorem along with an algebraic interpretation of the decomposability property presented in Definition 2.2.

THEOREM 2.3. *Consider $\mathcal{S} \subset \mathcal{X}$, $\mathcal{R} \subset \mathcal{Z}$, and $L\mathcal{S} + M\mathcal{S} \subset \mathcal{R}$. Let $S : \mathcal{S} \to \mathcal{X}$ and $R : \mathcal{R} \to \mathcal{Z}$ be the insertion maps of $\mathcal{S}$ in $\mathcal{X}$ and of $\mathcal{R}$ in $\mathcal{Z}$ respectively. Then $\mathcal{S}$, $\mathcal{R}$ decompose $\mathcal{X}$, $\mathcal{Z}$ relative to L, M if and only if there exist maps $S^+ : \mathcal{X} \to \mathcal{S}$ and $R^+ : \mathcal{Z} \to \mathcal{R}$ such that*

$$R^+R = I_\rho, \tag{2.9}$$

$$S^+S = I_\sigma, \tag{2.10}$$

$$R^+L = L_2S^+, \tag{2.11}$$

$$R^+M = M_2S^+, \tag{2.12}$$

where I_σ and I_ρ are the identity maps on $\mathcal{S}$ and $\mathcal{R}$ respectively.

Proof. If (2.9)–(2.12) hold, set $\mathcal{T} = \operatorname{Ker} S^+$ and $\mathcal{V} = \operatorname{Ker} R^+$. Then if $x \in \mathcal{X}$ and $z \in \mathcal{Z}$

$$x = SS^+x + (I_\sigma - SS^+)x, \tag{2.13}$$

$$z = RR^+z + (I_\rho - RR^+)z. \tag{2.14}$$

Since $S^+(I - SS^+)x = 0$ and $R^+(I - RR^+)z = 0$, we have that $x \in \mathcal{S} + \mathcal{T}$, so that $\mathcal{X} = \mathcal{S} + \mathcal{T}$ and $z \in \mathcal{R} + \mathcal{V}$, so that $\mathcal{Z} = \mathcal{R} + \mathcal{V}$. Also $x \in \mathcal{S} \cap \mathcal{T}$ and $z \in \mathcal{R} \cap \mathcal{V}$ imply that $x = Ss$ and $z = Rr$, say, and $S^+x = 0$ and $R^+z = 0$. Thus $0 = S^+Ss = s$ and

$0 = R^+Rr = r$, therefore $x = 0$ and $z = 0$. Hence $\mathcal{S} \cap \mathcal{T} = 0$ and $\mathcal{R} \cap \mathcal{V} = 0$. Finally $S^+x = 0$ and $R^+z = 0$ implies that

$$L_2S^+x = R^+Lx, \tag{2.15}$$

$$M_2S^+x = R^+Mx. \tag{2.16}$$

Hence $L\mathcal{T} + M\mathcal{T} \subset \mathcal{V}$ and $\mathcal{R} \oplus \mathcal{V} = \mathcal{Z}$.

Conversely if $L\mathcal{S} + M\mathcal{S} \subset \mathcal{R}$, $L\mathcal{T} + M\mathcal{T} \subset \mathcal{V}$ with $\mathcal{S} \oplus \mathcal{T} = \mathcal{X}$ and $\mathcal{R} \oplus \mathcal{V} = \mathcal{Z}$, let S^+ and R^+ be the natural projections $S^+ : \mathcal{S} \oplus \mathcal{T} \to \mathcal{S}$ and $R^+ : \mathcal{R} \oplus \mathcal{V} \to \mathcal{R}$ and using Lemma 2.1 the proof follows. □

Equations (2.9)–(2.12) can take a simpler and more useful form. We will show how (2.9)–(2.12) can be represented by two equivalent equations. Let $\tilde{\mathcal{T}} \oplus \mathcal{S} = \mathcal{X}$ and $\tilde{\mathcal{V}} \oplus \mathcal{R} = \mathcal{Z}$ be arbitrary complements of $\mathcal{S}$ and $\mathcal{R}$, respectively. In a compatible basis L, M, $\mathcal{S}$, and $\mathcal{R}$ have the following matrix representation:

$$\begin{pmatrix} L_1^{r\times s} & 0^{r\times\sigma} \\ L_3^{\rho\times s} & L_2^{\rho\times\sigma} \end{pmatrix}, \qquad \begin{pmatrix} M_1^{r\times s} & 0^{r\times\sigma} \\ M_3^{\rho\times s} & M_2^{\rho\times\sigma} \end{pmatrix},$$

$$R = \begin{pmatrix} 0 \\ I_\rho \end{pmatrix}, \qquad S = \begin{pmatrix} 0 \\ I_\sigma \end{pmatrix}.$$

Then

$$R^+ = [R_2 \;\; I_\rho], \tag{2.17}$$

$$S^+ = [S_2 \;\; I_\sigma] \tag{2.18}$$

for arbitrary R_2, S_2. Then using this in (2.9)–(2.12) yields

$$L_2S_2 - R_2L_1 = L_3, \tag{2.19}$$

$$M_2S_2 - R_2M_1 = M_3. \tag{2.20}$$

Equations (2.19) and (2.20) will be called the *generalized Sylvester* equations. Thus to check whether $\mathcal{S}$ and $\mathcal{R}$ decompose $\mathcal{X}$ and $\mathcal{Z}$ relative to L and M, it is enough to verify that these two equations have a solution S_2, R_2. It has been recently shown in the literature that the solution of these two generalized bilateral Sylvester equations can be achieved in a computationally stable manner by utilizing the generalized Schur methods with condition estimators [8].

For purposes of comparison it is interesting to see how (2.9)–(2.12) and (2.19)–(2.20) are transformed in the special case where the pencil $sL - M$ is regular and $L = I$. Notice that in this case the domain and the codomain are precisely the same. Consequently in this case $R = S$ and $R^+ = S^+$. By observing now (2.9)–(2.10), we see that the first two equations merge to one equation, while the third states a trivial equality. Specifically these four equations under the above concepts can take the form

$$S^+S = I_\sigma, \tag{2.21}$$

$$R^+M = M_2S^+, \tag{2.22}$$

which are precisely the equations presented in [19] for the notion of decomposability. Finally, we end this digression by considering how the two generalized Sylvester

equations transformed in this special case. It is clear that (2.19) is a trivial equality ($0 = 0$), while (2.20) takes the well-known form

$$M_2 S_2 - S_2 M_1 = M_3, \tag{2.23}$$

which constitutes the decomposability criterion for the pencil $sI - M$ [19].

3. Relation of decomposability to Kronecker invariants. The next step in our exploration of decomposability for the pencil $sL - M$ is to reveal the important role that the Kronecker invariants of this pencil play in the concept of decomposability. Specifically, we desire to find a condition that guarantees decomposabilty, as presented in Definition 2.2, in terms of the Kronecker invariants [6], [7], [17] of the pencil $sL - M$. The next proposition reveals this condition.

THEOREM 3.1. *$\mathcal{S}$ and $\mathcal{R}$ decompose $\mathcal{X}$ and $\mathcal{Z}$ relative to L and M if and only if the Kronecker invariants of the pencil $sL_1 - M_1$ together with those of the pencil $sL_2 - M_2$ (see (2.8)), give all the Kronecker invariants of the pencil $sL - M$.*

Proof. Assume that $\mathcal{S}$ and $\mathcal{R}$ decompose $\mathcal{X}$ and $\mathcal{Z}$ relative to L and M, then (2.9)–(2.12) hold true. Let P and Q be the canonical projections $P : \mathcal{X} \to \mathcal{X}/\mathcal{S}$ and $Q : \mathcal{Z} \to \mathcal{Z}/\mathcal{R}$. Select $P = [P_1 \;\; 0]$, $Q = [Q_1 \;\; 0]$, $S^+ = [S_2 \;\; I_\sigma]$, $R^+ = [R_2 \;\; I_\rho]$, $S = [0 \;\; I_\sigma]^T$, and $R = [0 \;\; I_\rho]^T$. As a result the matrices

$$\begin{pmatrix} P \\ S^+ \end{pmatrix}, \qquad \begin{pmatrix} Q \\ R^+ \end{pmatrix} \tag{3.1}$$

are full rank. Now by using Lemma 2.1 and (2.11) and (2.12), the following equations hold true

$$Q(sL - M) = (sL_1 - M_1)P, \tag{3.2}$$

$$R^+(sL - M) = (sL_2 - M_2)S^+. \tag{3.3}$$

Then by combining (3.2) and (3.3) in one equation, we get

$$\begin{pmatrix} P \\ S^+ \end{pmatrix} (sL - M) = \begin{pmatrix} sL_1 - M_1 & 0 \\ 0 & sL_2 - M_2 \end{pmatrix} \begin{pmatrix} Q \\ R^+ \end{pmatrix}, \tag{3.4}$$

where $(Q^T \;\; R^{+^T})^T$ and $(P^T \;\; S^{+^T})^T$ are full rank matrices of dimensions $l \times l$ and $n \times n$, respectively. Hence (3.4) can be written in the form

$$\tilde{P}(sL - M)\tilde{Q} = \begin{pmatrix} sL_1 - M_1 & 0 \\ 0 & sL_2 - M_2 \end{pmatrix}. \tag{3.5}$$

But we know that strictly equivalence property preserves the Kronecker structure of the pencil [6], [7], [17]. Hence the Kronecker invariants of the pencil $sL_1 - M_1$, together with those of the pencil $sL_2 - M_2$, give all the Kronecker invariants of the pencil $sL - M$.

Conversely, if the pencils $sL_1 - M_1$ and $sL_2 - M_2$ contain all the Kronecker invariants of the pencil $sL - M$, then there exist $\tilde{P}$ and $\tilde{Q}$ such that (3.5) holds true. To see this, follow the procedure of elimination proposed in [17]. Then partition $\tilde{Q}$ and $\tilde{P}$ as in (3.4). Hence (3.2) and (3.3) hold true, and consequently from (3.3), (2.11) and (2.12) follow immediately. Moreover since R^+ and S^+ are full row rank there exist

R and S such that (2.9) and (2.10) hold true. As a result, there exist $\mathcal{S}$ and $\mathcal{R}$ that decompose $\mathcal{X}$ and $\mathcal{Z}$ relative to L and M. □

If the pencil $sL - M$ is regular, that is, (2.2) holds true, then a sufficient condition for the decomposability is that the spectra of the pencils $sL_1 - M_1$ and $sL_2 - M_2$ are disjoint. This is a fairly good assumption in practice and has been used for the solution of generalized Lyapunov equations in the pole-placement problem using proportional feedback in singular systems [12].

This sufficient condition can also be used for the characterization of a semiregular system in the sense of [5]. In particular, by solving the proposed generalized equations under the assumption of disjoint spectra, we compute in a computationally stable fashion a partition as defined in [5, Lemma 4.1]. Moreover, the solvability of these two equations is a sufficient criterion for semiregular systems. Therefore, under these considerations, decomposabilty may find an application to the disturbance decoupling problem in singular systems from the computational point of view.

In the case where the pencil is regular and $L = I$, Theorem 3.1 reduces to the following corollary.

COROLLARY 3.2. : *$\mathcal{S}$ decomposes $\mathcal{X}$ relative to M if and only if the finite elementary divisors of the pencil $sI - M_1$ together with those of the pencil $sI - M_2$, give all the finite elementary divisors of the pencil $sI - M$.*

Moreover, a sufficient condition for decomposability in this case is that the spectra of M_1 and M_2 are disjoint, that is, $\sigma(M_1) \cap \sigma(M_2) = 0$. These results, not surprisingly, are the same as those presented in [19].

It is worthwhile pointing out some remarks as far as the partition of the spectrum is concerned, for both the cases of pencils of the form $sI - M$ and regular pencils of the form $sL - M$. In the case of the pencil $sI - M$ the partition of the spectrum implies a decomposition of the matrix M defined as in (2.8) (note that $L = I$). This decomposition does not imply that we can decompose the domain according to the matrices M_1 and M_2. However, if we can define row and column operations Q on M_1 and M_2 such that $M_1Q - QM_2 = M_3$, that is, we can eliminate M_3, then the partition of the spectrum implies the decomposition of the domain [19]. Similarly, in the case of a regular pencil $sL - M$, the partition of the spectrum does not imply the decomposition of the domain and the codomain through L_1, L_2, M_1 and M_2. However, if there exist row operations (codomain) and column operations (domain) such that (2.19) and (2.20) hold, then the partition of the spectrum implies the decomposition of the domain and the codomain.

An application of Theorem 3.1 to the Kronecker decomposition of the pencil $sL - M$ is presented next. The Kronecker decomposition of the pencil $sL - M$, as it is known [6], [7], [17], imposes a decomposition of the domain. Specifically, let $\mathcal{S}_r$, $\mathcal{S}_c$, $\mathcal{S}_\alpha$ and $\mathcal{S}_\infty$ be the subspaces of the row minimal indices (rmi), the column minimal indices (cmi), the finite elementary divisors (fed), and the infinite elementary divisors (ied), respectively, of the pencil $sL - M$ [7]. Then this implies a decomposition of the domain as

$$\mathcal{S} = \mathcal{S}_r \oplus \mathcal{S}_\infty \oplus \mathcal{S}_\alpha \oplus \mathcal{S}_c. \tag{3.6}$$

The next result shows that a similar decomposition is imposed on the codomain. The properties and relationships between these subspaces and their corresponding Kronecker invariants are extensively discussed in [14].

THEOREM 3.3. *Consider*

$$\mathcal{R}_r = L\mathcal{S}_r + M\mathcal{S}_r, \tag{3.7}$$

$$\mathcal{R}_c = L\mathcal{S}_c + M\mathcal{S}_c, \tag{3.8}$$

$$\mathcal{R}_\alpha = L\mathcal{S}_\alpha + M\mathcal{S}_\alpha, \tag{3.9}$$

$$\mathcal{R}_\infty = L\mathcal{S}_\infty + M\mathcal{S}_\infty, \tag{3.10}$$

where $\mathcal{R}_r$, $\mathcal{R}_c$, $\mathcal{R}_\alpha$ *and* $\mathcal{R}_\infty$ *will be called the* rmi, cmi, fed, *and* ied *subspaces of the codomain. Then*

$$\mathcal{R} = \mathcal{R}_r \oplus \mathcal{R}_\infty \oplus \mathcal{R}_\alpha \oplus \mathcal{R}_c. \tag{3.11}$$

Proof. We know [7] that the Kronecker decomposition of the pencil $sL - M$ imposes a decomposition on the domain. Consider now P and Q such that $P(sL-M)Q$ is in its Kronecker canonical form. Then Q can be partitioned as $Q = [\, Q_r |\, Q_c |\, Q_\infty |\, Q_\alpha]$ where the columns of Q_r, Q_c, Q_∞, Q_α span $\mathcal{S}_r$, $\mathcal{S}_c$, $\mathcal{S}_\infty$, $\mathcal{S}_\alpha$ respectively. Then we have

$$P[(sL - M)Q_r|(sL - M)Q_c|(sL - M)Q_\infty|(sL - M)Q_\alpha]. \tag{3.12}$$

Note that (3.12) using (3.7)–(3.10) can be written as $[P\mathcal{R}_r|P\mathcal{R}_c|P\mathcal{R}_\infty|P\mathcal{R}_\alpha]$. Keeping in mind the Kronecker canonical form and that P is a full rank matrix, (3.11) follows immediately. □

The following corollary is an application of Theorem 3.1 and reveals how closely related the Kronecker form of the pencil $sL - M$ and the concept of decomposabilty relative to the maps L and M are. The proof of this corollary is based on a process of *sequential decomposition* of the generalized pencil $sL - M$. This corollary is not presented for computational purposes, but for revealing the relation between the Kronecker invariants and decomposability conditions. Efficient computation techniques for finding the Kronecker invariants have been proposed in [17], [9].

COROLLARY 3.4. *Consider the decomposition of the domain defined in* (3.6) *and the decomposition of the codomain as defined in* (3.11). *Then there exist matrices* P *and* Q *that can be constructed using* (2.9)–(2.12) *such that the pencil* $P(sL - M)Q$ *is in its Kronecker canonical form.*

Remark. The geometric interpretation of this corollary is shown in Fig. 3.1.

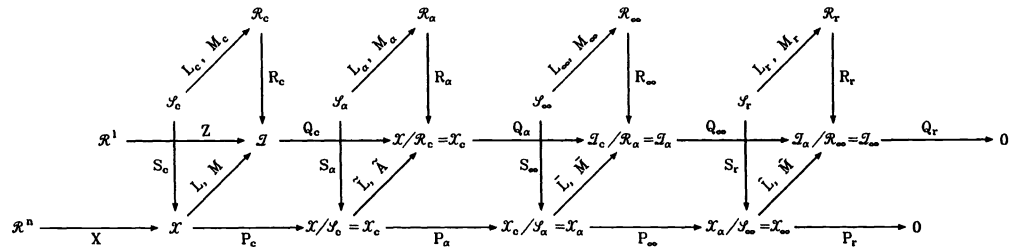


FIG. 3.1

Proof. The proof of this corollary is constructive and based on Theorem 3.1. Define

$$P_c : \mathcal{X} \to \mathcal{X}_c = \mathcal{X}/\mathcal{S}_c, \qquad Q_c : \mathcal{Z} \to \mathcal{Z}_c = \mathcal{Z}/\mathcal{R}_c, \tag{3.13}$$

$$P_\alpha : \mathcal{X}_c \to \mathcal{X}_\alpha = \mathcal{X}_c/\mathcal{S}_\alpha, \qquad Q_\alpha : \mathcal{Z}_c \to \mathcal{Z}_\alpha = \mathcal{Z}_c/\mathcal{R}_\alpha, \tag{3.14}$$

$$P_\infty : \mathcal{X}_\alpha \to \mathcal{X}_\infty = \mathcal{X}_\alpha/\mathcal{S}_\infty, \qquad Q_\infty : \mathcal{Z}_\alpha \to \mathcal{Z}_\infty = \mathcal{Z}_\alpha/\mathcal{R}_\infty, \tag{3.15}$$

$$S_c : \mathcal{S}_c \to \mathcal{X}, \qquad S_c^+ : \mathcal{X} \to \mathcal{S}_c \tag{3.16}$$

$$S_\alpha : \mathcal{S}_\alpha \to \mathcal{X}_c, \qquad S_\alpha^+ : \mathcal{X}_c \to \mathcal{S}_\alpha, \tag{3.17}$$

$$S_\infty : \mathcal{S}_\infty \to \mathcal{X}_\alpha, \qquad S_\infty^+ : \mathcal{X}_\alpha \to \mathcal{S}_\infty, \tag{3.18}$$

$$S_r : \mathcal{S}_r \to \mathcal{X}_\infty, \qquad S_r^+ : \mathcal{X}_\infty \to \mathcal{S}_r, \tag{3.19}$$

$$R_c : \mathcal{R}_c \to \mathcal{Z}, \qquad R_c^+ : \mathcal{Z} \to \mathcal{R}_c, \tag{3.20}$$

$$R_\alpha : \mathcal{R}_\alpha \to \mathcal{Z}_c, \qquad R_\alpha^+ : \mathcal{Z}_c \to \mathcal{R}_\alpha, \tag{3.21}$$

$$R_\infty : \mathcal{R}_\infty \to \mathcal{Z}_\alpha, \qquad R_\infty^+ : \mathcal{Z}_\alpha \to \mathcal{R}_\infty, \tag{3.22}$$

$$R_r : \mathcal{R}_r \to \mathcal{Z}_\infty, \qquad R_r^+ : \mathcal{Z}_\infty \to \mathcal{R}_r. \tag{3.23}$$

Define also $\dim \mathcal{S}_\psi = \sigma_\psi$ where $\psi \in \{c, r, \alpha, \infty\}$. Similarly define $\dim \mathcal{R}_\psi = \rho_\psi$ where $\psi \in \{c, r, \alpha, \infty\}$. Having defined these maps and using Theorem 2.3, the following equations hold true. (Variables above and beside matrices denote dimensions.)

$$\begin{matrix} & l & \\ q_c & \begin{pmatrix} Q_c \\ R_c^+ \end{pmatrix} & (sL - M) = \end{matrix} \begin{pmatrix} s\tilde{L} - \tilde{M} & 0 \\ 0 & sL_c - M_c \end{pmatrix} \begin{matrix} & n & \\ & \begin{pmatrix} P_c \\ S_c^+ \end{pmatrix} & \begin{matrix} p_c \\ \sigma_c \end{matrix} \end{matrix}, \tag{3.24}$$

$$\begin{matrix} & q_c & \\ q_\alpha & \begin{pmatrix} Q_\alpha \\ R_\alpha^+ \end{pmatrix} & (s\tilde{L} - \tilde{M}) = \end{matrix} \begin{pmatrix} s\bar{L} - \bar{M} & 0 \\ 0 & sL_\alpha - M_\alpha \end{pmatrix} \begin{matrix} & p_c & \\ & \begin{pmatrix} P_\alpha \\ S_\alpha^+ \end{pmatrix} & \begin{matrix} p_\alpha \\ \sigma_\alpha \end{matrix} \end{matrix}, \tag{3.25}$$

$$\begin{matrix} & q_\alpha & \\ q_\infty & \begin{pmatrix} Q_\infty \\ R_\infty^+ \end{pmatrix} & (s\bar{L} - \bar{M}) = \end{matrix} \begin{pmatrix} s\hat{L} - \hat{M} & 0 \\ 0 & sL_\infty - M_\infty \end{pmatrix} \begin{matrix} & p_\alpha & \\ & \begin{pmatrix} P_\infty \\ S_\infty^+ \end{pmatrix} & \begin{matrix} p_\infty \\ \sigma_\infty \end{matrix} \end{matrix}, \tag{3.26}$$

$$\begin{matrix} & q_r & \\ \rho_r & R_r^+ \, (s\hat{L} - \hat{L}) = & \end{matrix} \begin{matrix} & p_r & \\ (sL_r - M_r) & S_r^+ & \sigma_r \, . \end{matrix} \tag{3.27}$$

By combining (3.24)–(3.27) into one equation we get

$$\begin{pmatrix} R_r & 0 & 0 & 0 \\ 0 & I_{\rho_\infty} & 0 & 0 \\ 0 & 0 & I_{\rho_\alpha} & 0 \\ 0 & 0 & 0 & I_{\rho_c} \end{pmatrix} \begin{pmatrix} \begin{pmatrix} Q_\infty \\ R_\infty^+ \end{pmatrix} & 0 & 0 \\ 0 & I_{\rho_\alpha} & 0 \\ 0 & 0 & I_{\rho_c} \end{pmatrix} \begin{pmatrix} \begin{pmatrix} Q_\alpha \\ R_\alpha^+ \end{pmatrix} & 0 \\ 0 & I_{\rho_c} \end{pmatrix} \begin{pmatrix} Q_c \\ R_c^+ \end{pmatrix},$$

$$(sL - M) = \begin{pmatrix} sL_r - M_r & 0 & 0 & 0 \\ 0 & sL_\infty - M_\infty & 0 & 0 \\ 0 & 0 & sL_\alpha - M_\alpha & 0 \\ 0 & 0 & 0 & sL_c - M_c \end{pmatrix},$$

$$\begin{pmatrix} S_r & 0 & 0 & 0 \\ 0 & I_{\sigma_\infty} & 0 & 0 \\ 0 & 0 & I_{\sigma_\alpha} & 0 \\ 0 & 0 & 0 & I_{\sigma_c} \end{pmatrix} \begin{pmatrix} \begin{pmatrix} P_\infty \\ S_\infty^+ \end{pmatrix} & 0 & 0 \\ 0 & I_{\sigma_\alpha} & 0 \\ 0 & 0 & I_{\sigma_c} \end{pmatrix} \begin{pmatrix} \begin{pmatrix} P_\alpha \\ S_\alpha^+ \end{pmatrix} & 0 \\ 0 & I_{\sigma_c} \end{pmatrix} \begin{pmatrix} P_c \\ S_c^+ \end{pmatrix}.$$

Hence there exist matrices P and Q that can be constructed using (2.7)–(2.9) such that the pencil $P(sL - M)Q$ is in its Kronecker canonical form. That is

$$P(sL - M)Q = \begin{pmatrix} sL_r - M_r & 0 & 0 & 0 \\ 0 & sL_\infty - M_\infty & 0 & 0 \\ 0 & 0 & sL_\alpha - M_\alpha & 0 \\ 0 & 0 & 0 & sL_c - M_c \end{pmatrix},$$

which concludes the proof. □

4. An application of decomposability to controls design. Consider the linear time-invariant system described by

$$\dot{x} = Ax + Bu, \tag{4.1}$$

where $x \in \mathcal{X}$, $u \in \mathcal{U}$, $A \in \Re^{n\times n}$, $B \in \Re^{n\times m}$. We assume that B has full column rank $m < n$, and denote by N and B^+ a left annihilator and a left inverse of B, respectively (i.e., $NB = 0$, $B^+B = I_m$).

Consider the system

$$N\dot{x} = NAx, \tag{4.2a}$$

$$u = B^+(\dot{x} - Ax). \tag{4.2b}$$

Systems (4.1) and (4.2) are related in the following sense. For given $u(t)$, a solution $x(t)$ of (4.1) is also a solution of (4.2a). For a given $x(t)$ satisfying (4.2a), there exists a control $u(t)$ such that $x(t)$ is also a solution of (4.1), namely, the control given by (4.2b).

Since (4.2a) is not affected by any feedback of the form $u = Kx$, it is termed the *feedback-free representation* of the state-variable system (4.1). A subspace $\mathcal{V}$ is an (A, B)-*invariant* [19] subspace if

$$A\mathcal{V} \subset \mathcal{V} + \mathrm{Im}B, \tag{4.3}$$

or equivalently if there exists a map $K : \mathcal{X} \to \mathcal{U}$ such that

$$(A + BK)\mathcal{V} \subset \mathcal{V}. \tag{4.4}$$

We let $\mathcal{K}(\mathcal{V})$ denote the set of all K satisfying (4.4). Selecting V as a basis for $\mathcal{V}$, (4.3) can be written as

$$AV = VF - BG \tag{4.5}$$

for some F and G. Equation (4.5) is a *Sylvester* equation. Equivalently, (4.5) can be written as

$$(A + BK)V = VF$$

for some K where $KV = G$

The *controllability* pencil of (4.1) is defined as

$$C(s) = [sI - A \quad B]. \tag{4.6}$$

Similarly the *controllability pencil* of the nonsquare system (4.2) is defined as

$$\Gamma(s) = [sN - NA]. \tag{4.7}$$

It has been traditional to use $C(s)$ for pole-placement feedback design. Our contention, on the other hand, is that the pole placement problem is more conveniently solved using $\Gamma(s)$, which is a nonsquare pencil of lower order than $(sI - A)$.

The structure and the properties of these two pencils have been extensively studied in [7]. The aim of this section is to find computationally stable methods that

relate these two pencils, and then use the reduced-order pencil for the closed-loop eigenstructure assignment problem. Therefore we seek a pair $(N,\ B^+)$ such that (4.7) follows from (4.6) by only performing row and column eliminations.

Performing the algorithm proposed in [17], [18], which is based on orthogonal transformations, the controllability pencil (4.6) can be written as

$$[sI - U^TAU \quad U^TB] = \begin{array}{c} \\ r_f \\ r_c \end{array} \begin{array}{c} \begin{array}{ccc} r_f & r_c & m \end{array} \\ \begin{pmatrix} sI_f - A_f & 0 & 0 \\ -\bar{A} & sI_c - A_c & B_c \end{pmatrix} \end{array}, \tag{4.8}$$

where the pencil $sI_f - A_f$ contains all the uncontrollable modes of (4.1), while the pencil $[sI_c - A_c \ \ B]$ is full row rank for every s. Moreover, A_f may be taken in lower Schur form and A_c, B_c have the following form

$$A_c = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_{k-1} \\ t_k \end{array} \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_3 & \cdots & t_k \end{array} \\ \begin{pmatrix} A_{11} & A_{12} & 0 & \cdots & 0 \\ A_{22} & A_{22} & A_{23} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A_{k-1,1} & \cdots & \cdots & A_{k-1,k-1} & A_{k-1,k} \\ A_{k1} & \cdots & \cdots & A_{k,k-1} & A_{kk} \end{pmatrix} \end{array}, \qquad B_c = \begin{array}{c} \\ \\ \\ \\ \\ t_k \end{array} \begin{array}{c} \begin{array}{c} m \end{array} \\ \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ B_m \end{pmatrix} \end{array}. \tag{4.9}$$

That is, A_c is in lower block Hessenberg form, where the blocks $A_{j,j+1}$ $j \in \{1, 2, \ldots, k-1\}$ have full row rank t_j. Moreover B_m has full row rank t_k, but from our assumption that B is of full column rank ($t_k = m$) it follows that B_m is a square nonsingular matrix of dimensions $m \times m$. The pencil in (4.9), where A_f is in lower Schur form and A_c in lower block Hessenberg form, will be referred as the Schur–Hessenberg form of the system.

Now choose $(N,\ B^+)$ as follows

$$\begin{pmatrix} N \\ B^+ \end{pmatrix} = \begin{array}{c} \\ r_f \\ t_1 \\ \vdots \\ t_k \\ m \end{array} \begin{array}{c} \begin{array}{ccccc} r_f & t_1 & \cdots & t_k & m \end{array} \\ \begin{pmatrix} I_f & 0 & \ldots & 0 & 0 \\ 0 & I_{t_1} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & I_{t_{k-1}} & 0 \\ 0 & 0 & \ldots & 0 & B_m^{-1} \end{pmatrix} \end{array}. \tag{4.10}$$

Then premultiplication of (4.8) by (4.10) yields

$$\begin{array}{c} \\ n - m \\ m \end{array} \begin{array}{c} \begin{array}{cc} n & m \end{array} \\ \begin{pmatrix} sN - NA & 0 \\ sB^+ - B^+A & I_m \end{pmatrix} \end{array}, \tag{4.11}$$

where $sN - NA$ has the following form

(4.12)

$$\begin{array}{c} \\ sI_f - A_f \\ \\ \\ -A_3 \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} r_f & t_1 & t_2 & t_3 & \cdots & t_k \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ sI_{11} - A_{11} & -A_{12} & 0 & \ldots & 0 \\ -A_{22} & sI_{22} - A_{22} & -A_{23} & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -A_{k-1,1} & \ldots & \ldots & sI_{k-1,k-1} - A_{k-1,k-1} & -A_{k-1,k} \end{pmatrix} \end{array}$$

and $sB^+ - B^+A$ has the form

$$(-B_m^{-1}A_{k0} \quad -B_m^{-1}A_{k1} \quad \dots \quad -B_m^{-1}A_{k,k-1} \quad -B_m^{-1}(sI_{kk} - A_{kk})), \tag{4.13}$$

with A_{k0} defined by partitioning $\bar{A}$ in (4.8) as

$$\bar{A} = \begin{pmatrix} A_3 \\ A_{k0} \end{pmatrix}.$$

It is now important to note that we can obtain from the controllability pencil (4.8) the controllability pencil for the nonsquare system by simply *eliminating the last m rows and m columns* of the pencil in (4.11).

Another important remark is that the specific selection of the pair (N, B^+) does not destroy the structure of the identity matrix, the Schur form of A_f, or the Hessenberg form of the matrix A_c in the original controllability pencil (4.8). This property, as we shall show, is very important for the computational aspects of the problem.

Let us denote the pencil (4.12) by

$$sN - NA = \begin{pmatrix} sI_f - A_f & 0 \\ -A_3 & s\tilde{I} - A_2 \end{pmatrix}, \tag{4.14}$$

where $\tilde{I} = [I_l \ \ 0_k]$, ($I_l$ is the identity matrix of dimensions $l \times l$ and $l = n - m - r_f$, 0_m is the zero matrix with dimensions $l \times m$ ($t_k = m$)) and A_2, as can be seen from (4.11) and (4.12), is obtained from A_c by eliminating its m last rows. It is known [7], [19] that the pencil $sI_f - A_f$ in (4.12) contains the finite elementary divisors of the pencil $sN - NA$. Moreover, the pencil $s\tilde{I} - A_2$ contains all the column minimal indices of the pencil $sN - NA$. The next corollary now follows readily from Theorem 3.1 and the above discussion.

COROLLARY 4.1. *The generalized Sylvester equations* (2.19) *and* (2.20) *always have a solution for the pencil as decomposed in* (4.12), (4.14). *That is, for* $sL_1 - M_1 = sI_f - A_f$, $sL_2 - M_2 = s\tilde{I} - A_2$, *and* $sL_3 - M_3 = -A_3$.

Next, we exploit the special form of these two generalized Sylvester equations in order to formulate a computationally stable algorithm that calculates a solution ($X = S_2$, $Y = R_2$) for the decomposition in (4.14) that utilizes the special form of the matrices L and M.

Notice that for the pencil given in (4.14) $L_1 = I_f$, $L_3 = 0$, and $L_2 = \tilde{I}$. As a result (2.19) becomes

$$Y = \tilde{I}X. \tag{4.15}$$

Similarly, by setting $M_1 = A_f$, $L_3 = -A_3$, and $L_2 = A_2$, (2.20) takes the form

$$A_2X - YA_f = -A_3. \tag{4.16}$$

We can partition the matrix X (according to $\tilde{I}$) as follows:

$$X = \begin{matrix} \\ l \\ m \end{matrix} \begin{matrix} r_f \\ \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \end{matrix}, \qquad l + m = r_c. \tag{4.17}$$

Then (4.15) yields that $Y = X_1$. Also notice that (4.15) is satisfied for arbitrary X_2. Substituting (4.15) into (4.16) we get

$$HX - \tilde{I}XS = C, \tag{4.18}$$

where $H = A_2$, $S = A_f$, $C = -A_3$, and $X = [x_1|x_2|\cdots|x_{r_f}]$. A computationally stable algorithm for solving (4.18) is presented in [15].

If the proportional feedback

$$u = Kx \tag{4.19}$$

is applied to (2.1), the resulting closed-loop system is

$$\dot{x} = (A + BK)x. \tag{4.20}$$

Define the set of *uncontrollable* (i.e., *fixed*) *modes* of (4.1) as

$$\sigma_f(A) = \{\alpha \in \sigma(A) \mid \operatorname{rank}(C(\alpha)) < n\} \tag{4.21a}$$

and the set of the *controllable modes* of (4.1)

$$\sigma_c(A) = \{\alpha \in \sigma(A) \mid \operatorname{rank}(C(\alpha)) = n\}, \tag{4.21b}$$

where $\sigma(A)$ denotes the spectrum of the matrix A.

The next result [15] shows the most that may be achieved using proportional feedback in terms of pole assignment. It is our main result of this section and the proof provides a computationally stable design technique for computing the feedback gain K that assigns the desired closed-loop on $\mathcal{V}$, and relies on the Schur–Hessenberg form of the pencil $[sI - A \quad B]$. It uses the reduced-order nonsquare pencil (4.7), the concept of decomposabilty, and the algorithm proposed in [15].

THEOREM 4.2. *Given a desired closed-loop structure, select a self-conjugate set* Σ *of* $n = \dim \mathcal{V}$ *desired poles of* (4.1) *such that*

$$\sigma_f(A) \subset \Sigma. \tag{4.22}$$

Then there exists a feedback (4.19) *that assigns* Σ *as the closed-loop spectrum of* (4.20) *on an* (A, B)*-invariant subspace* $\mathcal{V}$.

Proof. Restricting (4.11) to $\Re^n$ and proposing a lower triangular form as the basis for $\mathcal{V}$ and a diagonal form for F, (4.5) becomes

$$\begin{pmatrix} A_f & 0 \\ A_3 & A_2 \end{pmatrix} \begin{pmatrix} V_f & 0 \\ V_3 & V_c \end{pmatrix} = \begin{pmatrix} I_f & 0 \\ 0 & \tilde{I} \end{pmatrix} \begin{pmatrix} V_f & 0 \\ V_3 & V_c \end{pmatrix} \begin{pmatrix} F_f & 0 \\ 0 & F_c \end{pmatrix}, \tag{4.23}$$

$$[G_f \; G_c] = -B_m^{-1}\{(A_{k0} \; A_{k1} \ldots A_{kk})V - (0_{k0} \; 0_{k1} \ldots I_{kk})VF\}. \tag{4.24}$$

Notice that in the proposed basis representation the state feedback is

$$[K_f \; K_c] = [G_f \; G_c]V^{-1}.$$

We now consider the three component equations of (4.23), solving them one at a time.

(a) *Uncontrollable modes*: Consider the diagonal term

$$A_f V_f = V_f F_f. \tag{4.25}$$

We select F_f as a matrix such that $\sigma(F_f) = \sigma(A_f)$. Using (4.18), where $\tilde{I} = I_{r_f}$ and $C = 0$, we compute V_f. Note that A_f is in its Schur form, a fact that further simplifies the solution of (4.18).

(b) *Coupling equation*: Now, consider the off-diagonal portion of (4.23),

$$A_2 V_3 - \tilde{I} V_3 F_f = -A_3 V_f. \tag{4.26}$$

Using (4.25), (4.26) can be written as

$$A_2 X - \tilde{I} X A_f = -A_3, \tag{4.27}$$

where $X = V_3 V_f^{-1}$. But (4.27) is exactly the same as (4.18), which, due to the decomposability property has always a solution, that can be easily computed.

(c) *Controllable modes*: The third portion of (4.23) is

$$A_2 V_c = \tilde{I} V_c F_c. \tag{4.28}$$

We select F_c in a lower Schur form such that $\sigma(F_c) = \Sigma - \sigma_f(A)$. The selection F_c is based on the fact that no further transformations will be necessary for the solution of (4.28). That is, we use (4.18), where $C = 0$, $H = A_2$, and $S = F_c$, to compute the solution of (4.28).

(d) *Driving equation*: We call (4.24) the driving equation since it is the one that incorporates the influence of the input $u(t)$ in the pole-placement problem. Equation (4.24) can be written as

$$[K_f \; K_c] = [G_f \; G_c] V^{-1} = -B_m^{-1}\{(A_{k0} \; A_{k1} \ldots A_{kk}) - (0_{k0} \; 0_{k1} \ldots I_{kk}) V F V^{-1}\}. \tag{4.29}$$

□

The proposed technique not only computes the feedback gain for the controllable part of the system (2.1) but also for the uncontrollable one. That is, we extend the feedback to the uncontrollable subspace, showing that the uncontrollable eigenvectors can be selected within limits. This is in contrast with Tsui [16] and Datta [4] where complete controllability is imposed on the system. In addition to that we do not require that the desired closed-loop spectrum and the uncontrollable modes must be disjoint as in [19]. The presented technique is exploiting the ideas presented in [10]. There a similar methodology is used. Specifically, the rows of B are compressed and then robust solutions are derived for (4.5). This is achieved by using the extra freedom that the input matrix B provides for assigning the closed-loop right eigenvectors.

Remarks. Several remarks must be made about the above algorithm [15].

(a) The proposed design technique is based on the reduced-order nonsquare pencil (4.7).

(b) The algorithm is based on unitary transformations that guarantee the computational stability of the technique.

(c) If the spectrum assignability objectives change there is no need to carry out the computations for the uncontrollable part of the sytem. This fact is due to the notion of decomposability. As a result, a considerable amount of computation is avoided.

(d) The design generalized Sylvester equations can be solved easily in a computationally effecient manner due to the specific choice of the pair $(N,\ B^+)$ and the idea of decomposability. Thus, once the controllability pencil has been brought to its Schur–Hessenberg form, the solution of the Sylvester equation can be easily achieved.

(e) The feedback-gain matrix is extended also to the uncontrollable subspace of the system.

(f) The solution V is implicitly dependent on B through N. However, the feedback gain matrix K is explicitly dependent on B.

For more details on the computational aspects of the proposed application and numerical examples see [15]. This application of decomposabilty for the closed-loop eigenstructure problem can be easily extended to the case of singular systems.

5. Conclusion. In this paper we presented the connection between strict equivalence of two pencils and the concept of quotient subspaces. This connection led us to the definition of decomposability for nonsquare pencils of the form $sL - M$. We proposed a set of necessary and sufficient conditions for the decomposability property. Furthermore, we showed that these algebraic conditions are equivalent to two coupled generalized Sylvester equations.

Throughout the paper we showed how the well-known notions for decomposability for the pencil $sI - M$ can be recovered as a special case of our general approach. We related the concept of decomposability to the Kronecker structure of the pencil $sL - M$. We drew connections between the Kronecker invariants and the decomposability property.

Finally we presented an application of the notion of decomposability to the control systems theory. In particular, we showed the advantages that provides for the closed-loop eigenstructure problem with state-feedback, by utilizing nonsquare pencils of reduced order.

Acknowledgments. We would like to thank the reviewers for their critical and incisive comments, which have done much to help improve the quality of this paper. We also want to thank Dr. N. Nichols for her input in the presentation of this work.

REFERENCES

[1] J. D. APLEVICH, *Minimal representations of implicit linear systems*, Automatica, 21 (1985), pp. 259–269.

[2] A. BUNSE-GERSTNER, V. MEHRMANN, AND N. K. NICHOLS, *Regularization of descriptor systems by derivative and proportional state feedback*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 46–47.

[3] L. DAI, *Singular Control Systems*, Lecture Notes in Control and Information Sciences, 118, Springer-Verlag, Berlin, 1989.

[4] B. N. DATTA, *An algorithm to assign eigenvalues in a Hessenberg matrix: Single input case*, IEEE Trans. Automat. Control, AC–32 (1987), pp. 414–417.

[5] L. R. FLETCHER AND A. AASARAAI, *On disturbance decoupling in descriptor systems*, SIAM J. Control Optim., 27 (1989), pp. 1319–1332.

[6] F. R. GANTMACHER, *The Theory of Matrices*, Chelsea Publishing Co., New York, 1959.

[7] S. JAFFE AND N. KARCANIAS, *Matrix pencil characterizations of almost (A, B)–invariant subspaces: a classification of geometric concepts*, Int. J. Control, 33 (1981), pp. 51–93.

[8] B. KÅGSTRÖM AND L. WESTIN, *Generalized Schur methods with condition estimators for solving the generalized Sylvester equation*, IEEE Trans. Automat. Control, AC–34 (1989), pp. 745–751.

[9] B. KÅGSTRÖM AND P. VAN DOOREN, *A generalized state space approach for the additive decomposition of a transfer matrix*, Report: University of Umea, Institute for Information Processing, S–901 87 Sweden, 1991.

[10] J. KAUTSKY, N. K. NICHOLS, AND P. VAN DOOREN, *Robust pole assignment in linear state feedback*, Internat. J. Control, 41 (1985), pp. 1129–1155.

[11] F. L. LEWIS, *A survey of linear singular systems*, J. Circuits. System Signal Process., 5 (1986), pp. 3–36.

[12] F. L. LEWIS AND K. ÖZÇALDIRAN, *Geometric structure and feedback in singular systems*, IEEE Trans. Automat. Control, AC–34 (1989), pp. 450–455.

[13] F. L. LEWIS, M. A. CHRISTODOULOU, B. G. MERTZIOS, AND K. ÖZÇALDIRAN, *Chained aggregation for singular systems*, IEEE Trans. Automat. Control, AC–34 (1989), pp. 1007–1013.

[14] J. J. LOISEAU, *Some geometric considerations about the Kronecker normal form*, Internat. J. Control, 42 (1985), pp. 1411–1431.

[15] V. L. SYRMOS AND F. L. LEWIS, *State feedback design techniques using nonsquare descriptions*, 26th Conference on Decision and Control, Honolulu, Hawaii, 1990.

[16] C. C. TSUI, *An algorithm for computing state feedback in multiinput linear systems*, IEEE Trans. Automat. Control, AC–31 (1986), pp. 243–246.

[17] P. VAN DOOREN, *The computation of Kronecker's canonical form of a singular system*, Linear Algebra Appl., 27 (1979), pp. 103–140.

[18] ———, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Automat. Control, AC–26 (1989), pp. 111–129.

[19] W. M. WONHAM, *Linear Multivariable Control: a Geometric Approach*, Springer-Verlag, New York, 1979.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 881–902, July 1994

011

NUMERICAL GRADIENT ALGORITHMS FOR EIGENVALUE AND SINGULAR VALUE CALCULATIONS*

J.B. MOORE†, R.E. MAHONY†, AND U. HELMKE‡

Abstract. Recent work has shown that the algebraic question of determining the eigenvalues, or singular values, of a matrix can be answered by solving certain continuous-time gradient flows on matrix manifolds. To obtain computational methods based on this theory, it is reasonable to develop algorithms that iteratively approximate the continuous-time flows. In this paper the authors propose two algorithms, based on a double Lie-bracket equation recently studied by Brockett, that appear to be suitable for implementation in parallel processing environments. The algorithms presented achieve, respectively, the eigenvalue decomposition of a symmetric matrix and the singular value decomposition of an arbitrary matrix. The algorithms have the same equilibria as the continuous-time flows on which they are based and inherit the exponential convergence of the continuous-time solutions.

Key words. eigenvalue decomposition, singular value decomposition, numerical gradient algorithm

AMS subject classifications. 15A18, 65F10

1. Introduction.

A traditional algebraic approach to determining the eigenvalue and eigenvector structure of an arbitrary matrix is the QR-algorithm. In the early 1980s it was observed that the QR-algorithm is closely related to a continuous-time differential equation that has become known through study of the Toda lattice. Symes [13] and Deift, Nanda, and Tomei [6] showed that for tridiagonal real symmetric matrices, the QR-algorithm is a discrete-time sampling of the solution to a continuous-time differential equation. This result was generalised to full complex matrices by Chu [3], and Watkins and Elsner [14] provided further insight in the late 1980s.

Brockett [2] studied dynamic matrix flows generated by the double Lie-bracket equation

$$\dot{H} = [H, [H, N]], \qquad H(0) = H_0$$

for constant symmetric matrices N and H_0, and where we use the Lie-bracket notation $[X, Y] = XY - YX$. We call this differential equation the *double-bracket* equation, and we call solutions of this equation *double-bracket flows.* Similar matrix differential equations in the area of Physics were known and studied prior to the references given above. An example, is the Landau–Lifschitz–Gilbert equation of micromagnetics

$$\frac{d\hat{m}}{dt} = \frac{\gamma}{1+\alpha^2}(\hat{m} \times \overline{H} - \alpha \hat{m} \times (\hat{m} \times \overline{H})) \quad |\hat{m}|^2 = 1,$$

as $\alpha \to \infty$ and $\gamma/\alpha \to k$, a constant. In this equation $\hat{m}, \overline{H} \in \mathbb{R}^3$ and the cross-product is equivalent to a Lie-bracket operation. The relevance of such equations

* Received by the editors April 13, 1992; accepted for publication (in revised form) September 25, 1992. The authors acknowledge the funding of the activities of the Cooperative Research Centre for Robust and Adaptive Systems by the Australian Commonwealth Government under the Cooperative Research Centres Program. The authors also acknowledge additional support from Boeing Commercial Aircraft Corporation, Inc.

† Department of Systems Engineering, Research School of Physical Sciences and Systems Engineering, Australian National University, A.C.T., 0200, Australia (robert.mahony@anu.edu.au).

‡ Department of Mathematics, University of Regensburg, 8400 Regensburg, Germany.

to traditional linear algebra problems, however, has only recently been studied and discretisations of such flows have not been investigated.

The double-bracket equation is not known to be a continuous-time version of any previously existing linear algebra algorithm; however, it exhibits exponential convergence to an equilibrium point on the manifold of self-equivalent symmetric matrices [2], [5], [9]. Brockett [2] was able to show that this flow could be used to diagonalise real symmetric matrices, and thus, to find their eigenvalues, sort lists, and even to solve linear programming problems. Part of the flexibility and theoretical appeal of the double-bracket equation follows from its dependence on the arbitrary matrix parameter N, which can be varied to control the transient behaviour of the differential equation.

In independent work by Driessel [7], Chu and Driessel [5], Smith [12] and Helmke and Moore [8], a similar gradient flow approach is developed for the task of computing the singular values of a general nonsymmetric, nonsquare matrix. The differential equation obtained in these approaches is almost identical to the double-bracket equation. In [8], it is shown that these flows can also be derived as special cases of the double-bracket equation for a nonsymmetric matrix, suitably augmented to be symmetric.

With the theoretical aspects of these differential equations becoming known, and with applications in the area of balanced realizations [10], [11] along with the more traditional matrix eigenvalue problems, there remains the question of efficiently computing their solutions. No explicit solutions to the differential equations have been obtained and a direct numerical estimate of their integral solutions seems unlikely to be an efficient computational algorithm. Iterative algorithms that approximate the continuous-time flows, however, seem more likely to yield useful numerical methods. Furthermore, discretisations of such isospectral matrix flows are of general theoretical interest in the field of numerical linear algebra. For example, the algorithms proposed in this paper involve adjustable parameters, such as step-size selection schemes and a matrix parameter N, which are not present in traditional algorithms such as the QR-algorithm or the Jacobi method.

In this paper, we propose a new algorithm termed the *Lie-bracket algorithm*, for computing the eigenvalues of an arbitrary symmetric matrix

$$H_{k+1} = e^{-\alpha_k[H_k,N]} H_k e^{\alpha_k[H_k,N]}.$$

For suitably small α_k, termed *time-steps*, the algorithm is an approximation of the solution to the continuous time double-bracket equation. Thus, the algorithm represents an approach to developing new recursive algorithms based on approximating suitable continuous-time flows. We show that for suitable choices of time-steps, the Lie-bracket algorithm inherits the same equilibria as the double-bracket flow. Furthermore, exponential convergence of the algorithm is shown. This paper presents only theoretical results on the Lie-bracket algorithm and does not attempt to compare its performance to that of existing methods for calculating the eigenvalues of a matrix.

Continuous-time gradient flows that compute the singular values of arbitrary nonsymmetric matrices, such as those covered in [5], [8], [9], [12], have a similar form to the double-bracket equation on which the Lie-bracket algorithm was based. We use this similarity to generate a new scheme for computing the singular values of a general matrix termed the *singular value algorithm*. The natural equivalence between the Lie-bracket algorithm and the singular value algorithm is demonstrated and exponential convergence results follow almost directly.

Associated with the main algorithms presented for the computation of the eigenvalues or singular values of matrices are algorithms that compute the full eigenspace decompositions of given matrices. These algorithms are closely related to the Lie-bracket algorithm and also display exponential convergence.

The paper is divided into eight sections including the Introduction and an Appendix. In §2 of this paper, we consider the Lie-bracket algorithm and prove a proposition that ensures the algorithm converges to a fixed point. Section 3 deals with choosing step-size selection schemes and proposes two valid deterministic functions for defining the time-steps. Considering the particular step-size selection schemes presented in §3 we return to the question of stability in §4 and show that the Lie-bracket algorithm has a unique exponentially attractive fixed point, though several of the technical proofs are deferred to the Appendix. This completes the discussion for the symmetric case and §5 considers the nonsymmetric case and the singular value decomposition. Section 6 presents associated algorithms that compute the eigenspace decompositions of given initial conditions. A number of computational issues are briefly mentioned in §7, while §8 provides a conclusion.

2. The Lie-bracket algorithm. In this section, we begin by introducing the least squares potential that underpins the recent gradient flow results and then we describe the double Lie-bracket equation first derived by Brockett [2]. The Lie-bracket recursion is introduced and conditions are given that guarantee convergence of the algorithm.

Let N and H be real symmetric matrices and consider the potential function

$$\begin{aligned} \psi(H) &:= ||H - N||^2 \\ &= ||H||^2 + ||N||^2 - 2\mathrm{tr}(NH), \end{aligned} \tag{1}$$

where the norm used is the Frobenius norm $||X||^2 := \mathrm{tr}(X^T X) = \sum x_{ij}^2$, with x_{ij} the elements of X. Note that $\psi(H)$ measures the least squares difference between the elements of H and the elements of N. Let $M(H_0)$ be the set of orthogonally similar matrices, generated by some symmetric initial condition $H_0 = H_0^T \in \mathbb{R}^{n\times n}$. Then

$$M(H_0) = \{U^T H_0 U \mid U \in O(n)\}, \tag{2}$$

where $O(n)$ denotes the group of all $n \times n$ real orthogonal matrices. It is shown in [9, p. 48] that $M(H_0)$ is a smooth compact Riemannian manifold with explicit forms given for its tangent space and Riemannian metric. Furthermore, in [1], [5] the gradient of $\psi(H)$, with the respect to the normal Riemannian metric on $M(H_0)$ [9, p. 50], is shown to be $\nabla\psi(H) = -[H,[H,N]]$. Consider the gradient flow given by the solution of

$$\begin{aligned} \dot{H} &= -\nabla\psi(H) \\ &= [H,[H,N]], \quad \text{with } H(0) = H_0, \end{aligned} \tag{3}$$

which we call the *double-bracket flow* [2], [5]. Thus, the double-bracket flow is a gradient flow that acts to decrease or minimise the least squares potential ψ on the manifold $M(H_0)$. Note that from (1), this is equivalent to increasing or maximising $\mathrm{tr}(NH)$. We refer to the matrix H_0 as the *initial condition* and the matrix N as the *target matrix*.

The *Lie-bracket algorithm* proposed in this paper is

$$H_{k+1} = e^{-\alpha_k[H_k,N]} H_k e^{\alpha_k[H_k,N]} \tag{4}$$

for arbitrary symmetric $n \times n$ matrices H_0 and N and some suitably small scalars α_k termed *time-steps*. To motivate the Lie-bracket algorithm, consider the curve $H_{k+1}(t) = e^{-t[H_k,N]} H_k e^{t[H_k,N]}$. Thus, $H_{k+1}(0) = H_k$ and $H_{k+1} = H_{k+1}(\alpha_k)$, the $(k+1)$th iteration of (4). Observe that

$$\frac{d}{dt}(e^{-t[H_k,N]} H_k e^{t[H_k,N]})\Big|_{t=0} = [H_k, [H_k, N]],$$

and thus, $e^{-t[H_k,N]} H_k e^{t[H_k,N]}$ is a first approximation of the double-bracket flow at $H_k \in M(H_0)$. It follows that for small α_k, the solution to (3) evaluated at time $t = \alpha_k$ with $H(0) = H_k$ is approximately $H_{k+1} = H_{k+1}(\alpha_k)$.

It is easily seen from above that stationary points of (3) are fixed points of (4). In general, (4) may have more fixed points than just the stationary points of (3), however, Proposition 2.1 shows that this is not the case for a suitable choice of time-step α_k. We use the term *equilibrium point* to mean a fixed point of the algorithm that is also a stationary point of (3).

To implement (4) it is necessary to specify the time-steps α_k. We do this by considering functions $\alpha_N : M(H_0) \to \mathbb{R}_+$ and setting $\alpha_k := \alpha_N(H_k)$. We refer to the function α_N as the *step-size selection scheme*. We require that the step-size selection scheme satisfies the following condition.

CONDITION 2.1. *Let $\alpha_N : M(H_0) \to \mathbb{R}_+$ be a step-size selection scheme for the Lie-bracket algorithm on $M(H_0)$. Then α_N is well defined and continuous on all of $M(H_0)$, except possibly those points $H \in M(H_0)$ where $HN = NH$. Furthermore, there exist real numbers $B, \gamma > 0$, such that $B > \alpha_N(H) \geq \gamma$ for all $H \in M(H_0)$ where α_N is well defined.*

Remark 2.1. We find that the variable step-size selection scheme proposed in this paper, which provides the best simulation results, is discontinuous at all the points $H \in M(H_0)$, such that $[H, N] = 0$.

Remark 2.2. Note that the definition of a step-size selection scheme depends implicitly on the matrix parameter N. Indeed, α_N can be thought of as a function in two matrix variables N and H.

CONDITION 2.2. *Let N be a diagonal $n \times n$ matrix with distinct diagonal entries $\mu_1 > \mu_2 > \ldots > \mu_n$.*

Remark 2.3. This condition on N, along with Condition 2.1 on the step-size selection scheme, is chosen to ensure that the Lie-bracket algorithm converges to a diagonal matrix from which the eigenvalues of H_0 can be directly determined.

Let $\lambda_1 > \lambda_2 > \ldots > \lambda_r$ be the eigenvalues of H_0 with associated algebraic multiplicities $n_1, \ldots, n_r$ satisfying $\sum_{i=1}^{r} n_i = n$. Note that as H_0 is symmetric, the eigenvalues of H_0 are all real. Thus, the diagonalisation of H_0 is

$$\Lambda := \begin{bmatrix} \lambda_1 I_{n_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_r I_{n_r} \end{bmatrix}, \tag{5}$$

where I_{n_i} is the $n_i \times n_i$ identity matrix. For generic initial conditions and a target matrix N that satisfies Condition 2.2, the continuous-time equation (3) converges exponentially fast to Λ [2], [9]. Thus, the eigenvalues of H_0 are the diagonal entries of the limiting value of the infinite time solution to (3). The Lie-bracket algorithm behaves similarly to (3) for small α_k and, given a suitable step-size selection scheme, should converge to the same equilibrium as the continuous-time equation.

PROPOSITION 2.1. *Let H_0 and N be $n \times n$ real symmetric matrices where N satisfies Condition* 2.2. *Let $\psi(H)$ be given by* (1) *and let $\alpha_N : M(H_0) \to \mathbb{R}_+$ be a step-size selection scheme that satisfies Condition* 2.1. *For $H_k \in M(H_0)$, let $\alpha_k = \alpha_N(H_k)$ and define*

$$\triangle \psi(H_k, \alpha_k) := \psi(H_{k+1}) - \psi(H_k), \tag{6}$$

where H_{k+1} is given by (4). *Suppose*

$$\triangle \psi(H_k, \alpha_k) < 0 \quad \text{when } [H_k, N] \neq 0. \tag{7}$$

Then (a) *The iterative equation* (4) *defines an isospectral (eigenvalue preserving) recursion on the manifold $M(H_0)$.*

(b) *The fixed points of* (4) *are characterised by matrices $H \in M(H_0)$ satisfying*

$$[H, N] = 0. \tag{8}$$

(c) *Every solution H_k, for $k = 1, 2, \ldots$, of* (4), *converges as $k \to \infty$, to some $H_\infty \in M(H_0)$ where $[H_\infty, N] = 0$.*

Proof. To prove part (a), note that the Lie-bracket $[H, N]^T = -[H, N]$ is skew-symmetric. As the exponential of a skew-symmetric matrix is orthogonal, (4) is an orthogonal conjugation of H_k and hence is isospectral.

For part (b) note that if $[H_k, N] = 0$, then by direct substitution into (4) we see $H_{k+1} = H_k$ and thus, $H_{k+l} = H_k$ for $l \geq 1$, and H_k is a fixed point of (4). Conversely if $[H_k, N] \neq 0$, then from (7), $\triangle\psi(H_k, \alpha_k) \neq 0$, and thus $H_{k+1} \neq H_k$. By inspection, points satisfying (8) are stationary points of (3), and indeed are known to be the only stationary points of (3) [9, pg. 50]. Thus, the fixed points of (4) are equilibrium points in the sense that they are all stationary points of (3). To prove part (c) we need the following lemma.

LEMMA 2.2. *Let N satisfy Condition* 2.2 *and α_N satisfy Condition* 2.1 *such that the Lie-bracket algorithm satisfies* (7). *The Lie-bracket algorithm* (4) *has exactly $n!/\prod_{i=1}^{r}(n_i!)$ distinct equilibrium points in $M(H_0)$. These equilibrium points are characterised by the matrices $\pi^T \Lambda \pi$, where π is an $n \times n$ permutation matrix, a rearrangement of the rows of the identity matrix, and Λ is given by* (5).

Proof. Note that part (b) of Proposition 2.1 characterises equilibrium points of (4) as $H \in M(H_0)$ such that $[H, N] = 0$. Evaluating this condition componentwise for $H = \{h_{ij}\}$ gives

$$h_{ij}(\mu_j - \mu_i) = 0,$$

and hence by Condition 2.2, $h_{ij} = 0$ for $i \neq j$. Using the fact that (4) is isospectral, it follows that equilibrium points are diagonal matrices that have the same eigenvalues as H_0. Such matrices are distinct and can be written in the form $\pi^T \Lambda \pi$ for π an $n \times n$ permutation matrix. A simple counting argument yields the number of matrices that satisfy this condition to be $n!/\prod_{i=1}^{r}(n_i!)$. □

Consider for a fixed initial condition H_0, the sequence H_k generated by the Lie-bracket algorithm. Observe that condition (7) implies that $\psi(H_k)$ is strictly monotonic decreasing for all k where $[H_k, N] \neq 0$. Also, since ψ is a continuous function on the compact set $M(H_0)$, then ψ is bounded from below and $\psi(H_k)$ will converge to some nonnegative value ψ_∞. As $\psi(H_k) \to \psi_\infty$ then $\triangle\psi(H_k, \alpha_k) \to 0$.

For an arbitrary positive number ϵ, define the open set $D_\epsilon \subset M(H_0)$, consisting of all points of $M(H_0)$, within an ϵ neighbourhood of some equilibrium point of (4).

The set $M(H_0) - D_\epsilon$ is a closed, compact subset of $M(H_0)$ on which the matrix function $H \mapsto [H, N]$ does not vanish. As a consequence, the difference function (6) is continuous and strictly negative on $M(H_0) - D_\epsilon$, and thus can be over bounded by some strictly negative number $\delta_1 < 0$. Moreover, as $\triangle\psi(H_k, \alpha_k) \to 0$, then there exists a $K = K(\delta_1)$ such that for all $k > K$ then $0 \geq \triangle\psi(H_k, \alpha_k) > \delta_1$. This ensures that $H_k \in D_\epsilon$ for all $k > K$. In other words, H_k is converging to some subset of possible equilibrium points.

Imposing the upper bound B on the step-size selection scheme α_N, Condition 2.2, it follows that $\alpha_N(H_k)[H_k, N] \to 0$ as $k \to \infty$. Thus, $e^{\alpha_N(H_k)[H_k,N]} \to I$, the identity matrix, and hence, $e^{-\alpha_N(H_k)[H_k,N]} H_k e^{\alpha_N(H_k)[H_k,N]} \to H_k$ as $k \to \infty$. As a consequence $||H_{k+1} - H_k|| \to 0$ for $k \to \infty$ and this combined with the distinct nature of the fixed points, Lemma 2.2, and the partial convergence already shown, completes the proof. □

Remark 2.4. In Condition 2.2 it was required that N have distinct diagonal entries. If this condition is not satisfied, the equilibrium condition $[H, N] = 0$ may no longer force H to be diagonal, and thus, though the algorithm will converge, it is unlikely to converge to a diagonal matrix.

3. Step-size selection. The Lie-bracket algorithm (4) requires a suitable step-size selection scheme before it can be implemented. To generate such a scheme, we use the potential (1) as a measure of the convergence of (4) at each iteration. Thus, we aim to choose each time-step to maximise the absolute change in potential $|\triangle\psi|$ of (6), such that $\triangle\psi < 0$. Optimal time-steps can be determined at each step of the iteration by completing a line search to maximise the absolute change in potential as the time-step is increased. Such an approach, however, involves high computational overheads and we aim rather to obtain a step-size selection scheme in the form of a scalar equation depending on known values.

Using the Taylor expansion, we express $\triangle\psi(H_k, \tau)$ for a general time-step τ, as a linear term plus a higher order error term. By estimating the error term we obtain a mathematically simple function $\triangle\psi_U(H_k, \tau)$, which is an upper bound to $\triangle\psi(H_k, \tau)$ for all τ. Then, choosing a suitable time-step α_k based on minimising $\triangle\psi_U$, we guarantee that the actual change in potential, $\triangle\psi(H_k, \alpha_k) \leq \triangle\psi_U(H_k, \alpha_k) < 0$, satisfies (7). Due to the simple nature of the function $\triangle\psi_U$, there is an explicit form for the time-step α_k depending only on H_k and N. We begin by deriving an expression for the error term.

LEMMA 3.1. *For the kth step of the recursion* (4) *the change in potential* $\triangle\psi(H_k, \tau)$ *of* (6), *for a time-step* τ *is*

$$\triangle\psi(H_k, \tau) = -2\tau||[H_k, N]||^2 - 2\tau^2 \mathrm{tr}(N\mathcal{R}_2(\tau)) \tag{9}$$

with

$$\mathcal{R}_2(\tau) := \int_0^1 (1-s) H''_{k+1}(s\tau) ds, \tag{10}$$

where $H''_{k+1}(\tau)$ *is the second derivative of* $H_{k+1}(\tau)$ *with respect to* τ.

Proof. Let $H_{k+1}(\tau)$ be the $(k+1)$th recursive estimate for an arbitrary time-step τ. Thus $H_{k+1}(\tau) = e^{-\tau[H_k,N]} H_k e^{\tau[H_k,N]}$. It is easy to verify that the first and second time derivatives of H_{k+1} are exactly

$$H'_{k+1}(\tau) = [H_{k+1}(\tau), [H_k, N]],$$
$$H''_{k+1}(\tau) = [[H_{k+1}(\tau), [H_k, N]], [H_k, N]].$$

Applying Taylor's theorem, then

$$\begin{aligned} H_{k+1}(\tau) &= H_{k+1}(0) + \tau \frac{d}{d\tau} H_{k+1}(0) + \tau^2 \int_0^1 (1-s) H''_{k+1}(s\tau) ds, \\ &= H_k + \tau [H_k, [H_k, N]] + \tau^2 \mathcal{R}_2(\tau). \end{aligned} \tag{11}$$

Consider the change in the potential $\psi(H)$ between the points H_k and $H_{k+1}(\tau)$,

$$\begin{aligned} \triangle\psi(H_k, \tau) &= \psi(H_{k+1}(\tau)) - \psi(H_k) \\ &= -2\mathrm{tr}(N(H_{k+1}(\tau) - H_k)) \\ &= -2\mathrm{tr}(N(\tau[H_k, [H_k, N]] + \tau^2 \mathcal{R}_2(\tau))) \\ &= -2\tau ||[H_k, N]||^2 - 2\tau^2 \mathrm{tr}(N\mathcal{R}_2(\tau)). \end{aligned} \tag{12}$$

□

Note that for $\tau = 0$, then $\triangle\psi(H_k, 0) = 0$ and also that

$$\frac{d}{d\tau} \triangle\psi(H_k, \tau)\Big|_{\tau=0} = -2||[H_k, N]||^2.$$

Thus, for sufficiently small τ the error term $\tau^2 \mathrm{tr}(N\mathcal{R}_2(\tau))$ becomes negligible and $\triangle\psi(H_k, \tau)$ is strictly negative. Let $\alpha_{\mathrm{opt}} > 0$ be the first time for which

$$\frac{d}{d\tau} \triangle\psi(H_k, \tau)\Big|_{\tau=\alpha_{\mathrm{opt}}} = 0,$$

then $\triangle\psi(H_k, \alpha_{\mathrm{opt}}) < \triangle\psi(H_k, \tau) < 0$ for all strictly positive $\tau < \alpha_{\mathrm{opt}}$. It is not possible, however, to estimate α_{opt} directly from (12) due to the transcendental nature of the error term $\mathcal{R}_2(\tau)$. By considering two separate estimates of the error term, we obtain two step-size selection schemes $\alpha_k \leq \alpha_{\mathrm{opt}}$. The first and constant step-size selection scheme follows from a loose bound of the error, whereas the second variable step-size selection scheme follows from a more sophisticated argument and results in faster convergence of (4).

LEMMA 3.2 (Constant step-size selection scheme). *The constant time-step*

$$\alpha_N^c = \frac{1}{4||H_0|| \cdot ||N||} \tag{13}$$

satisfies Condition 2.1. *Furthermore, the Lie-bracket algorithm, equipped with the step-size selection scheme* α_N^c, *satisfies* (7).

Proof. Recall that for the Frobenius norm $|\mathrm{tr}(XY)| \leq ||X|| \cdot ||Y||$. Then

$$\begin{aligned} \triangle\psi(H_k, \tau) &\leq -2\tau ||[H_k, N]||^2 + 2\tau^2 |\mathrm{tr}(N\mathcal{R}_2(\tau))| \\ &\leq -2\tau ||[H_k, N]||^2 + 2\tau^2 ||N|| \cdot ||\mathcal{R}_2(\tau)|| \\ &\leq -2\tau ||[H_k, N]||^2 + 2\tau^2 ||N|| \cdot \\ &\qquad \int_0^1 (1-s) |||[[H_{k+1}(s\tau), [H_k, N]], [H_k, N]]|| ds \\ &\leq -2\tau ||[H_k, N]||^2 + 4\tau^2 ||N|| \cdot ||H_0|| \cdot |||[H_k, N]||^2 \\ &=: \triangle\psi_U(H_k, \tau). \end{aligned} \tag{14}$$

Thus $\triangle\psi_U(H_k, \tau)$ is an upper bound for $\triangle\psi(H_k, \tau)$ and has the property that for sufficiently small τ, it is strictly negative; see Fig. 1. Due to the quadratic form of

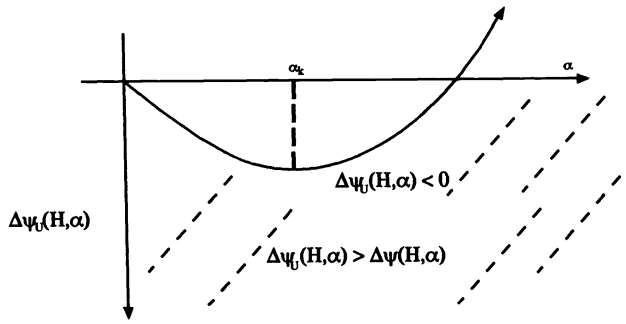


FIG. 1. *The upper bound on* $\triangle\psi(H_k,\alpha)$ *viz* $\triangle\psi_U(H_k,\alpha)$.

$\triangle\psi_U(H_k,\tau)$ in τ, it is immediately clear that $\alpha_k^c = \alpha_N^c(H_k) = 1/(4||H_0||||N||)$ of (13) is the minimum of (14). □

A direct norm bound of the integral error term is not likely to be a tight estimate of the error and the function $\triangle\psi_U$ is a fairly crude bound for $\triangle\psi$. The following more sophisticated estimate results in a step-size selection scheme that causes the Lie-bracket algorithm to converge an order of magnitude faster.

LEMMA 3.3 (An improved bound for $\triangle\psi(H_k,\tau)$). *Note the difference function* $\triangle\psi(H_k,\tau)$ *can be over bounded by*

$$\begin{aligned}\triangle\psi(H_k,\tau) &\le -2\tau||[H_k,N]||^2 \\ &\quad + \frac{||H_0||\cdot|||[N,[H_k,N]]||}{||[H_k,N]||}\left(e^{2\tau||[H_k,N]||}-1-2\tau||[H_k,N]||\right) \\ &=: \triangle\psi_U^*(H_k,\tau).\end{aligned} \tag{15}$$

Proof. Consider the Taylor series expansion of the matrix exponential

$$e^A = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \cdots.$$

It is easily verified that

$$\begin{aligned}e^A B e^{-A} &= B + [A,B] + \frac{1}{2!}[A,[A,B]] + \frac{1}{3!}[A,[A,[A,B]]] + \cdots \\ &= \sum_{i=0}^{\infty}\frac{1}{i!}ad_A^i B.\end{aligned} \tag{16}$$

Here $ad_A^i B = ad_A\,(ad_A^{i-1}B)$, $ad_A^0 B = B$, where $ad_A : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is the linear map $X \mapsto AX - XA$. Substituting $-\tau[H_k,N]$ and H_k for A and B in (16) and comparing with (11), gives

$$\tau^2\mathcal{R}_2(\tau) = \sum_{j=2}^{\infty}\frac{1}{j!}ad_{-\tau[H_k,N]}^j(H_k).$$

Considering $|\mathrm{tr}(N\mathcal{R}_2(\tau))|$ and using the readily established identity $\mathrm{tr}(Nad_{-A}^j B) =$

$\mathrm{tr}((ad_A^j N)B)$ gives

$$\begin{aligned}
|\tau^2 \mathrm{tr}(N\mathcal{R}_2(\tau))| &= \left| \sum_{j=2}^{\infty} \frac{1}{j!} \mathrm{tr}\left(ad^j_{\tau[H_k,N]}(N) H_k \right) \right| \\
&\leq \sum_{j=2}^{\infty} \frac{1}{j!} ||ad^j_{\tau[H_k,N]}(N)|| \cdot ||H_0|| \\
&\leq \sum_{j=2}^{\infty} \frac{1}{j!} (2||\tau[H_k,N]||)^{j-1} ||ad_{\tau[H_k,N]}(N)|| \cdot ||H_0|| \\
&= \frac{||H_0|| \cdot ||ad_{\tau[H_k,N]}(N)||}{2\tau||[H_k,N]||} \sum_{j=2}^{\infty} \frac{1}{j!} (2\tau||[H_k,N]||)^j \\
&= \frac{||H_0|| \cdot ||[N,[H_k,N]]||}{2||[H_k,N]||} \left(e^{2\tau||[H_k,N]||} - 1 - 2\tau||[H_k,N]|| \right).
\end{aligned}$$

Thus combining this with the first line of (14) gives (15). □

The variable step-size selection scheme is derived from this estimate of the error term in the same manner the constant step-size selection scheme was derived in Lemma 3.2.

LEMMA 3.4 (Variable step-size selection scheme). *The step-size selection scheme* $\alpha_N^* : M(H_0) \to \mathbb{R}_+$

$$\alpha_N^*(H) = \frac{1}{2||[H,N]||} \log\left(\frac{||[H,N]||^2}{||H_0||\ ||[N,[H,N]]||} + 1 \right), \tag{17}$$

where all norms are Frobenius norms, satisfies Condition 2.1. *Furthermore, the Lie-bracket algorithm, equipped with the step-size selection scheme* α_N^*, *satisfies* (7).

Proof. We first show that α_N^* satisfies the requirements of Condition 2.1. As the Frobenius norm is a continuous function, then α_N^* is well defined and continuous at all points $H \in M(H_0)$ such that $[H,N] \neq 0$. Note that when $[H,N] = 0$, then α_N^* is not well defined. To show that there exists a positive constant γ, such that $\alpha_N^*(H) > \gamma$, consider the following lower bound,

$$\begin{aligned}
\alpha_N^L &:= \frac{1}{2||[H_k,N]||} \log\left(\frac{||[H_k,N]||}{2||[H_0||\ ||N||} + 1 \right) \\
&\leq \frac{1}{2||[H_k,N]||} \log\left(\frac{||[H_k,N]||^2}{2||[H_0||\ ||N||\ ||[H_k,N]||} + 1 \right) \\
&\leq \frac{1}{2||[H_k,N]||} \log\left(\frac{||[H_k,N]||^2}{||[H_0||\ ||[N,[H_k,N]]||} + 1 \right),
\end{aligned} \tag{18}$$

which is just α_N^*. Using L'Hôpital's rule it can be seen that the limit of α_N^L at an equilibrium point, $H \in M(H_0)$ such that $[H,N] = 0$, is $1/(4||H_0|| \cdot ||N||)$. Including these points in the definition of α_N^L, gives that α_N^L is a continuous, strictly positive, well-defined function for all $H \in M(H_0)$. Thus, as $M(H_0)$ is compact, there exists a real number $\gamma > 0$ such that

$$\alpha_N^* \geq \alpha_N^L \geq \gamma > 0$$

on $M(H_0) - \{H_\infty \mid [H_\infty, N] = 0\}$.

To show that there exists a real number $B > 0$, such that $\alpha_N^*(H) < B$, $H \in M(H_0)$, set $[H, N] = X = \{x_{ij}\}$. For N given by Condition 2.2, then $|||[N, X]||| = \sum_{i=j}(\mu_i - \mu_j)^2 x_{ij}^2$, where $x_{ii} = 0$ as $[H, N]$ is skew-symmetric. Observe that

$$\begin{aligned} ||X||/|||[N, X]||| &= \frac{\sum_{i=j} x_{ij}^2}{\sum_{i=j}(\mu_i - \mu_j)^2 x_{ij}^2} \\ &\leq \max_{i=j}(\mu_i - \mu_j)^{-2} =: b \end{aligned}$$

for all choices of $X = -X^T$. It follows that

$$\begin{aligned} \alpha_N^*(H) &= \frac{1}{2||X||} \log\left(\frac{||X||^2}{||H_0||\,|||[N, X]|||} + 1\right) \\ &\leq \frac{1}{2||X||} \log\left(\frac{||X||b}{||H_0||} + 1\right) \\ &\leq \frac{b}{2||H_0||} =: B \end{aligned}$$

since $\log(x + 1) \leq x$ for $x > 0$.

Finally, for a matrix $H_k \in M(H_0)$, $[H_k, N] \neq 0$, the time-step $\alpha_N^*(H_k) = \alpha_k^* > 0$ minimises (15), and from Lemma 3.3 it follows that $0 \geq \triangle\psi_U^*(H_k, \tau) \geq \triangle\psi(H_k, \tau)$. Thus, the Lie-bracket algorithm, equipped with the step-size selection scheme α_N^*, satisfies (7) and the proof is complete. □

4. Stability analysis. In this section we study the stability of equilibria of the Lie-bracket algorithm (4). It is shown that for generic initial conditions and any step-size selection scheme that satisfies Condition 2.1 and (7), the solution H_k of the Lie-bracket algorithm converges to the unique equilibrium point Λ given by (5). Furthermore, we derive local exponential bounds on the rate of convergence. To improve the readability of the paper the proofs of a number of the more technical results have been deferred to an appendix. We begin by showing that Λ is the unique locally asymptotically stable equilibrium point of (4).

LEMMA 4.1. *Let N satisfy Condition* (2.2) *and α_N be some selection scheme that satisfies Condition* 2.1 *and* (7). *The Lie-bracket algorithm* (4) *has a unique locally asymptotically stable equilibrium point Λ given by* (5). *All other equilibrium points of* (4) *are unstable.*

Proof. It is known that Λ is the unique local and global minimum of the potential function ψ on $M(H_0)$ [9]. By assumptions on N and α_N, $\psi(H_k)$ is monotonically decreasing. Thus the domain of attraction of Λ contains an open neighbourhood of Λ, and hence, Λ is a locally asymptotically stable equilibrium point of (4).

All other equilibrium points H_∞ are either saddle points or maxima of ψ [9]. Thus for any neighbourhood D of some equilibrium point $H_\infty \neq \Lambda$, there exists some $H_0 \in D$ such that $\psi(H_0) < \psi(H_\infty)$. It follows that the solution to the Lie-bracket algorithm, with initial condition H_0, will not converge to H_∞ and thus H_∞ is unstable. □

Lemma 4.1 is sufficient to conclude that for generic initial conditions the Lie-bracket algorithm will converge to the unique matrix Λ. It is difficult to characterise the set of initial conditions for which the algorithm converges to some unstable equilibrium point $H_\infty \neq \Lambda$. For the continuous-time double-bracket flow, however, it is

known that the unstable basins of attraction of such points are of zero measure in $M(H_0)$ [9].

LEMMA 4.2. *Let N satisfy Condition* 2.2. *Let $d \in \mathbb{R}_+$ be a constant such that $0 < d < 1/2||H_0||_2||N||_2$ and consider the constant step-size selection scheme, $\alpha_N^d : M(H_0) \to \mathbb{R}_+$,*

$$\alpha_N^d(H) = d.$$

The Lie-bracket algorithm (4) *equipped with the step-size selection scheme α_N^d has a unique locally exponentially asymptotically stable equilibrium point Λ given by* (5).

Proof. Since α_N^d is a constant function, the time-step $\alpha_k^d = \alpha_N^d(H_k) = d$ is constant. Thus, the map

$$H_k \mapsto e^{-d[H_k,N]} H_k e^{d[H_k,N]}$$

is a differentiable map on all $M(H_0)$, and we may consider the linearisation of this map at the equilibrium point Λ given by (5). The linearisation of this recursion expressed in terms of $\Xi_k \in T_\Lambda M(H_0)$ (the tangent space of the equilibrium point Λ) is

$$\Xi_{k+1} = \Xi_k - d[(\Xi_k N - N\Xi_k)\Lambda - \Lambda(\Xi_k N - N\Xi_k)]. \tag{19}$$

Thus for the elements of Ξ_k, we have

$$(\xi_{ij})_{k+1} = [1 - d(\lambda_i - \lambda_j)(\mu_i - \mu_j)](\xi_{ij})_k \quad \text{for } i,j = 1,\dots,n. \tag{20}$$

The tangent space $T_\Lambda M(H_0)$ at Λ consists of those matrices $\Xi = [\Lambda, \Omega]$ where $\Omega \in \mathrm{Skew}(n)$, the class of skew-symmetric matrices [9, p. 53]. Thus, the matrices Ξ are parameterised by their components ξ_{ij}, where $i < j$, and $\lambda_i \neq \lambda_j$. This is a linearly independent parameterisation of $T_\Lambda M(H_0)$ and the eigenvalues of the linearisation (19) can be read directly from (20) as $1 - d(\lambda_{\pi(i)} - \lambda_{\pi(j)})(\mu_i - \mu_j)$, for $i < j$ and $\lambda_i \neq \lambda_j$. Since $\lambda_i \geq \lambda_j$ when $i > j$, then if $d < 1/2||H_0||_2||N||_2$ it follows that

$$|1 - d(\lambda_i - \lambda_j)(\mu_i - \mu_j)| < 1$$

for all $i < j$ with $\lambda_i \neq \lambda_j$. Classical stability theory gives that Λ is a locally exponentially asymptotically stable equilibrium point of the recursion (4) with an exponential rate of convergence of $\max_{i<j,\lambda_i=\lambda_j}\{d(\lambda_i - \lambda_j)(\mu_i - \mu_j)\}$. □

Remark 4.1. As $||N||_2||H_0||_2 < 2||N||||H_0||$, the constant step-size selection scheme α_N^c is an example of such a selection scheme where $c = 1/(4||H_0|| \cdot ||N||)$.

Remark 4.2. Let $\alpha_N : M(H_0) \to \mathbb{R}_+$ be a step-size selection scheme that satisfies Condition 2.1 and (7) and is also continuous on all $M(H_0)$. Let Λ be the locally asymptotically stable equilibrium point given by (5). Set $\alpha_\infty = \alpha_N(\Lambda)$ and observe that the linearisation of the Lie-bracket algorithm will be of the form (19) with d replaced by α_∞. Recall that the α_N^L, scheme defined in (18) is continuous with limit $\alpha_N^L(H_\infty) = 1/(4||H_0|| \cdot ||N||)$. Thus, Λ is an exponentially asymptotically stable equilibrium point for the Lie-bracket recursion equipped with the step-size selection scheme α_N^L.

To show that the Lie-bracket algorithm is exponentially stable at Λ for the α_N^* step-size selection scheme is technically difficult due to the discontinuous nature of α_N^* at equilibrium points. The proof of the following proposition is given in the Appendix.

PROPOSITION 4.3. *Let N satisfy assumption* (2.2) *and α_N^* be the step-size selection scheme given by Lemma* 3.4. *The iterative algorithm* (4), *has a unique exponentially attractive equilibrium point Λ given by* (5).

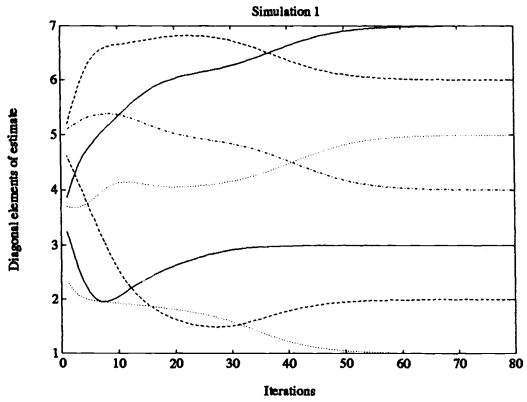


FIG. 2. *A plot of the diagonal elements h_{ii} of each iteration H_k of the Lie-bracket algorithm run on a 7×7 initial condition H_0 with eigenvalues $(1, \ldots, 7)$. The target matrix N was chosen to be* $\operatorname{diag}(1, \ldots, 7)$.

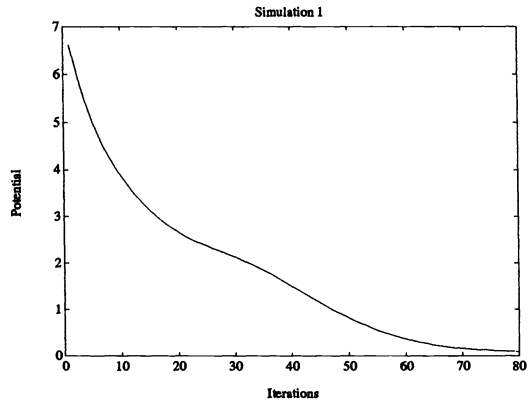


FIG. 3. *The potential $\psi(H_k) = ||H_k - N||^2$ for the Lie-bracket recursion.*

To give an indication of the behaviour of the Lie-bracket algorithm, two plots of a simulation have been included as Figs. 2 and 3. The simulation was run on a random 7×7 symmetric initial value matrix with eigenvalues $1, \ldots, 7$. The target matrix N is chosen as $\operatorname{diag}(1, \ldots, 7)$ and as a consequence the minimum potential is $\psi_\infty = 0$. Figure 2 is a plot of the diagonal entries of the recursive estimate H_k. The off-diagonal entries converge to zero as the diagonal entries converge to the eigenvalues of H_k. Figure 3 is a plot of the potential $||H_k - N||^2$ verses the iteration k. This plot clearly shows the monotonic decreasing nature of the potential at each step of the algorithm.

We summarise the results of §§2–4 in Theorem 4.4.

THEOREM 4.4. *Let $H_0 = H_0^T$ be a real symmetric $n \times n$ matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$. Let $N \in \mathbb{R}^{n \times n}$ satisfy Condition 2.2 and let α_N be either the constant step-size selection* (13) *or the variable step-size selection* (17). *The Lie-bracket*

recursion

$$H_{k+1} = e^{-\alpha_k[H_k,N]} H_k e^{\alpha_k[H_k,N]},$$
$$\alpha_k = \alpha_N(H_k),$$

with initial condition H_0, *has the following properties*:

(i) *The recursion is isospectral.*

(ii) *If* H_k *is a solution of the Lie-bracket algorithm, then* $\psi(H_k) = ||H_k - N||^2$ *is strictly monotonically decreasing for every* $k \in \mathbb{N}$, *where* $[H_k, N] \neq 0$.

(iii) *Fixed points of the recursive equation are characterised by matrices* $H \in M(H_0)$ *such that*

$$[H, N] = 0.$$

(iv) *Fixed points of the recursion are exactly the stationary points of the double-bracket equation. These points are termed* equilibrium *points.*

(v) *Let* H_k, $k = 1, 2, \ldots$, *be a solution to the Lie-bracket algorithm, then* H_k *converges to a matrix* $H_\infty \in M(H_0)$, $[H_\infty, N] = 0$, *an equilibrium point of the recursion.*

(vi) *All equilibrium points of the Lie-bracket algorithm are strictly unstable except* $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, *which is locally exponentially asymptotically stable.*

5. Singular value computations. In this section we consider discretisations of continuous-time flows to compute the singular values of an arbitrary matrix.

A singular value decomposition of a matrix $H_0 \in \mathbb{R}^{m\times n}$, $m \geq n$ is a matrix decomposition

$$H_0 = V^T \Sigma U, \tag{21}$$

where $V \in O(m)$, $U \in O(n)$ and

$$\Sigma = \begin{pmatrix} \sigma_1 I_{n_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r I_{n_r} \\ & 0_{(m-n)\times n} & \end{pmatrix}. \tag{22}$$

Here $\sigma_1 > \sigma_2 > \cdots > \sigma_r \geq 0$ are the distinct singular values of H_0 occurring with multiplicities $n_1, \ldots, n_r$, such that $\sum_{i=1}^r n_i = n$. By convention the singular values of a matrix are chosen to be nonnegative. It should be noted that although such a decomposition always exists and Σ is unique, there is no unique choice of orthogonal matrices V and U. The approach we take is to define an algorithm that converges to Σ and thus computes the singular values of H_0 without directly generating the orthogonal decomposition.

Let $S(H_0)$ be the set of all orthogonally equivalent matrices to H_0,

$$S(H_0) = \{ V^T H_0 U \in \mathbb{R}^{m\times n} \mid V \in O(m),\ U \in O(n)\}. \tag{23}$$

It is shown in [9, p. 89] that $S(H_0)$ is a smooth compact Riemannian manifold with explicit forms given for its tangent space and Riemannian metric. Following [4], [5], [8], [9], and [12] we consider the task of calculating the singular values of a matrix H_0 by minimising the least squares cost function $\psi : S(H_0) \to \mathbb{R}_+$, $\psi(H) = ||H - N||^2$. It is shown in [8] and [9] that ψ achieves a unique local and global minimum at the

point $\Sigma \in S(H_0)$. Moreover, in [8], [9], and [12] the explicit form for the gradient $\nabla\psi$ is calculated. The gradient flow is

$$\begin{aligned} \dot{H} &= -\nabla\psi(H) \\ &= H\{H,N\} - \{H^T,N^T\}H, \end{aligned} \tag{24}$$

with $H(0) = H_0$ the initial condition. Here we have used a generalisation of the Lie-bracket $\{X,Y\} := X^TY - Y^TX = -\{X,Y\}^T$.

To accomplish the task of computing the singular values of a matrix we require N to satisfy the following.

CONDITION 5.1. *Let N be an $m \times n$ matrix*

$$N = \begin{bmatrix} \mu_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_n \\ & 0_{(m-n)\times n} & \end{bmatrix},$$

where $\mu_1 > \mu_2 > \cdots > \mu_n > 0$ are strictly positive, distinct real numbers.

For generic initial conditions and a target matrix N that satisfies Condition 5.1, it is known that (24) converges exponentially fast to $\Sigma \in S(H_0)$ [8], [12]. A recursive version of this flow follows from an analogous argument to that used in the derivation of the Lie-bracket algorithm. For H_0 and N constant $m \times n$ matrices, the *singular value algorithm* proposed is

$$H_{k+1} = e^{-\alpha_k\{H_k^T,N^T\}}H_k e^{\alpha_k\{H_k,N\}}. \tag{25}$$

The singular value algorithm and the Lie-bracket algorithm are closely linked as is shown in the following lemma.

LEMMA 5.1. *Let H_0, N be $m \times n$ matrices. For any $H \in \mathbb{R}^{m\times n}$ define a map $H \mapsto \widehat{H} \in \mathbb{R}^{(m+n)\times(m+n)}$, where*

$$\widehat{H} = \begin{pmatrix} 0_{m\times m} & H \\ H^T & 0_{n\times n} \end{pmatrix}. \tag{26}$$

For any sequence of real numbers α_k, $k = 1,\ldots,\infty$ the iterations

$$H_{k+1} = e^{-\alpha_k\{H_k^T,N^T\}}H_k e^{\alpha_k\{H_k,N\}} \tag{27}$$

with initial condition H_0 and

$$\widehat{H}_{k+1} = e^{-\alpha_k[\widehat{H}_k,\widehat{N}]}\widehat{H}_k e^{\alpha_k[\widehat{H}_k,\widehat{N}]} \tag{28}$$

with initial condition $\widehat{H}_0$ are equivalent.

Proof. Consider the iterative solution to (28) and evaluate the multiplication in the block form of (26). This gives two equivalent iterative solutions, one the transpose of the other, both of which are equivalent to the iterative solution to (27). □

Remark 5.1. Note that $\widehat{H}_0$ and $\widehat{N}$ are symmetric $(m+n)\times(m+n)$ matrices and that, as a result, the iteration (28) is just the Lie-bracket algorithm.

Remark 5.2. The equivalence given by Lemma 5.1 is complete in every way. In particular, H_∞ is an equilibrium point of (27) if and only if $\widehat{H}_\infty$ is an equilibrium point of (28). Similarly, $H_k \to H_\infty$ if and only if $\widehat{H}_k \to \widehat{H}_\infty$ as $k \to \infty$.

This leads us directly to consider step-size selection schemes for the singular value algorithm induced by selection schemes that we have already considered for the Lie-bracket algorithm. Indeed if $\alpha_{\widehat{N}} : M(\widehat{H_0}) \to \mathbb{R}_+$ is a step-size selection scheme for (4) on $M(\widehat{H}_0)$, and $H_k \in S(H_0)$, then we can define a time-step α_k for the singular value algorithm by

$$\alpha_k = \alpha_{\widehat{N}}(\widehat{H}_k). \tag{29}$$

Thus, if (28) equipped with a step-size selection scheme $\alpha_{\widehat{N}}$ satisfies Condition 2.1 and (7), then from Lemma 5.1, (27) will satisfy similar conditions. For simplicity, we deal only with the step-size selection schemes induced by the constant step-size selection (13) and the variable step-size selection (17). Thus we may state the main convergence theorem for the singular value algorithm.

THEOREM 5.2. *Let H_0, N be $m \times n$ matrices where $m \geq n$ and N satisfies Condition* 5.1. *Let $\alpha_{\widehat{N}} : M(\widehat{H_0}) \to \mathbb{R}_+$ be either the constant step-size selection* (13), *or the variable step-size selection* (17). *The singular value algorithm*

$$\begin{aligned} H_{k+1} &= e^{-\alpha_k\{H_k^T, N^T\}} H_k e^{\alpha_k\{H_k, N\}}, \\ \alpha_k &= \alpha_{\widehat{N}}(\widehat{H}_k), \end{aligned}$$

with initial condition H_0, has the following properties:

(i) *The singular value algorithm is a self-equivalent* (*singular value preserving*) *recursion on the manifold $S(H_0)$.*

(ii) *If H_k is a solution of the singular value algorithm, then $\psi(H_k) = ||H_k - N||^2$ is strictly monotonically decreasing for every $k \in \mathbb{N}$, where $\{H_k, N\} \neq 0$ and $\{H_k^T, N^T\} \neq 0$.*

(iii) *Fixed points of the recursive equation are characterised by matrices $H \in S(H_0)$ such that*

$$\{H_k, N\} = 0 \quad \textit{and} \quad \{H_k^T, N^T\} = 0. \tag{30}$$

Fixed points of the recursion are exactly the stationary points of the singular value gradient flow (24) *and are termed* equilibrium *points.*

(iv) *Let H_k, $k = 1, 2, \ldots$, be a solution to the singular value algorithm, then H_k converges to a matrix $H_\infty \in S(H_0)$, an equilibrium point of the recursion.*

(v) *All equilibrium points of the Lie-bracket algorithm are strictly unstable except Σ given by* (22), *which is locally exponentially asymptotically stable.*

Proof. To prove part (i), note that the generalised Lie-bracket $\{X, Y\} = -\{X, Y\}^T$ is skew-symmetric and thus (25) is an orthogonal conjugation and preserves the singular values of H_k. Also note that the potential $\psi(H_k) = \frac{1}{2}\psi(\widehat{H}_k)$. Moreover, Lemma 5.1 shows that the sequence $\widehat{H}_k$ is a solution to the Lie-bracket algorithm and thus from Proposition 2.1, $\frac{1}{2}\psi(\widehat{H}_k)$ must be monotonically decreasing for all $k \in \mathbb{N}$ such that $[\widehat{H}_k, \widehat{N}] \neq 0$, which is equivalent to (30). This proves part (ii) and part (iii) follows by noting that if $\{H_k^T, N^T\} = 0$ and $\{H_k, N\} = 0$, then $H_{k+l} = H_k$ for $l = 1, 2, \ldots$, and H_k is a fixed point of (25). Moreover, since $\psi(H_k)$ is strictly monotonic decreasing for all $\{H_k, N\} \neq 0$ and $\{H_k^T, N^T\} \neq 0$, then these points can be the only fixed points. It is known that these are the only stationary points of (24) [8], [9], [12].

To prove (iv), we need the following characterisation of equilibria of the singular value algorithm.

LEMMA 5.3. *Let N satisfy Condition 5.1 and $\alpha_{\widehat{N}}$ be either the constant step-size selection (13) or the variable step-size selection (17). The singular value algorithm (25) equipped with time-steps $\alpha_k = \alpha_{\widehat{N}}(\widehat{H}_k)$ has exactly $2^n n!/\prod_{i=1}^r (n_i!)$ distinct equilibrium points in $S(H_0)$. Furthermore, these equilibrium points are characterised by the matrices*

$$\begin{pmatrix} \pi^T & 0_{n\times(m-n)} \\ 0_{(m-n)\times n} & 0_{(m-n)\times(m-n)} \end{pmatrix} \Sigma S \pi,$$

where π is an $n \times n$ permutation matrix and $S = \operatorname{diag}(\pm 1, \ldots, \pm 1)$ a sign matrix.

Proof. Equilibrium points of (25) are characterised by the two conditions (30). For $H = (h_{ij})$, $\{H, N\} = 0$ is equivalent to

$$\mu_j h_{ji} - \mu_i h_{ij} = 0 \quad \text{for } i = 1, \ldots, n, \qquad j = 1, \ldots, n.$$

Similarly, the condition $\{H^T, N^T\} = 0$ is equivalent to

$$\begin{aligned} \mu_j h_{ij} - \mu_i h_{ji} = 0 \quad &\text{for } i = 1, \ldots, n, \qquad j = 1, \ldots, n, \\ h_{ij}\mu_j = 0 \quad &\text{for } i = n+1, \ldots, m, \qquad j = 1, \ldots, n. \end{aligned}$$

By manipulating the relationships, and using the distinct, positive nature of the μ_i, it is easily shown that $h_{ij} = 0$ for $i \neq j$. Using the fact that (25) is self equivalent, the only possible matrices of this form that have the same singular values as H_0 are characterised as above. A simple counting argument shows that the number of distinct equilibrium points is $2^n n!/\prod_{i=1}^r (n_i!)$. □

The proof of Theorem 5.2 part (iv) is now directly analogous to the proof of Proposition 2.1 part (c). It remains only to prove Theorem 5.2 part (v), which involves the stability analysis of the equilibrium points characterised by (30). It is not possible to directly apply the results obtained in §4 to the Lie-bracket recursion $\widehat{H}_k$, since the $\widehat{N}$ does not satisfy Condition 2.2. However, for the constant step-size selection scheme induced by (13), and using analogous arguments to those used in Lemmas 4.1 and 4.2, it follows that Σ is the unique locally exponentially attractive equilibrium point of the singular value algorithm. Thus, for the constant step-size selection scheme, $\widehat{\Sigma}$ is the unique exponentially attractive equilibrium point of the Lie-bracket algorithm on $M(\widehat{H}_0)$, and now the argument from Proposition 4.3 applies directly and $\widehat{\Sigma}$ is exponentially attractive for the variable step-size selection scheme (17). This completes the proof. □

Remark 5.3. Theorem 5.2 holds true for any time-steps $\alpha_k = \alpha_{\widehat{N}}(\widehat{H}_k)$ induced by a step-size selection scheme, $\alpha_{\widehat{N}}$, that satisfies Condition 2.1, such that Theorem 4.4 holds.

Remark 5.4. It is possible that for nongeneric initial conditions, the singular value algorithm may converge to a diagonal matrix with the singular values ordered in a different manner to Σ. However, all simulations run have converged exponentially fast to the unique matrix Σ, and thus it is likely that the attractive basins of the unstable equilibrium points have zero measure. Note that for the continuous-time flows, it is known that the attractive basins of the unstable equilibrium points have zero measure in $S(H_0)$ [9].

6. Associated orthogonal algorithms. In the previous sections we have proposed the Lie-bracket and the singular value algorithms that calculate the eigenvalues and singular values, respectively, of given initial conditions. Associated with these recursions are orthogonal recursions that compute the eigenvectors or singular vectors

of given initial conditions and provide a full spectral decomposition. To simplify the subsequent analysis we impose a genericity condition on the initial condition H_0.

CONDITION 6.1. *If $H_0 = H_0^T \in \mathbb{R}^{n\times n}$ is a real symmetric matrix then assume that H_0 has distinct eigenvalues $\lambda_1 > \cdots > \lambda_n$. If $H_0 \in \mathbb{R}^{m\times n}$, where $m \geq n$, then assume that H_0 has distinct singular values $\sigma_1 > \cdots > \sigma_n > 0$.*

For a sequence of positive real numbers α_k for $k = 1, 2, \ldots$ the *associated orthogonal Lie-bracket algorithm* is

$$U_{k+1} = U_k e^{\alpha_k [U_k^T H_0 U_k, N]}, \qquad U_0 \in O(n), \tag{31}$$

where $H_0 = H_0^T \in \mathbb{R}^{n\times n}$ is symmetric. For an arbitrary initial condition $H_0 \in \mathbb{R}^{m\times n}$ the *associated orthogonal singular value algorithm* is

$$\begin{aligned} V_{k+1} &= V_k e^{\alpha_k \{U_k^T H_0^T V_k, N^T\}}, \qquad V_0 \in O(m) \\ U_{k+1} &= U_k e^{\alpha_k \{V_k^T H_0 U_k, N\}}, \qquad U_0 \in O(n). \end{aligned} \tag{32}$$

Note that in each case the exponents of the exponential terms are skew-symmetric and thus the recursions will remain orthogonal.

Let $H_0 = H_0^T \in \mathbb{R}^{n\times n}$ and consider the map $g : O(n) \to M(H_0)$, $U \mapsto U^T H_0 U$, which is a smooth surjection. If U_k is a solution to (31) it follows that

$$g(U_{k+1}) = e^{-\alpha_k [g(U_k), N]} g(U_k) e^{\alpha_k [g(U_k), N]},$$

which generates the Lie-bracket algorithm (4). Thus, g maps the associated orthogonal Lie-bracket algorithm with initial condition U_0 to the Lie-bracket algorithm with initial condition $U_0^T H_0 U_0$ on $M(U_0^T H_0 U_0) = M(H_0)$.

Remark 6.1. Consider the potential function $\phi : O(n) \to \mathbb{R}_+$, $\phi(U) = ||U^T H_0 U - N||^2$ on the set of orthogonal $n \times n$ matrices. Using the standard induced Riemannian metric from $\mathbb{R}^{n\times n}$ on $O(n)$, the *associated orthogonal gradient flow* is [2], [3], [5], [9]

$$\dot{U} = -\nabla\phi(U) = U[U^T H_0 U, N].$$

THEOREM 6.1. *Let $H_0 = H_0^T$ be a real symmetric $n \times n$ matrix that satisfies Condition* 6.1. *Let $N \in \mathbb{R}^{n\times n}$ satisfy Condition* 2.2, *and let α_N be either the constant step-size selection* (13) *or the variable step-size selection* (17). *The recursion*

$$\begin{aligned} U_{k+1} &= U_k e^{\alpha_k [U_k^T H_0 U_k, N]}, \qquad U_0 \in O(n), \\ \alpha_k &= \alpha_N(H_k) \end{aligned}$$

referred to as the associated orthogonal Lie-bracket algorithm *has the following properties*:

(i) *A solution U_k, $k = 1, 2, \ldots$, to the associated orthogonal Lie-bracket algorithm remains orthogonal.*

(ii) *Let $\phi(U) = ||U^T H_0 U - N||^2$ be a map from $O(n)$ to the set of nonnegative reals $\mathbb{R}_+$. Let U_k, $k = 1, 2, \ldots$, be a solution to the associated orthogonal Lie-bracket algorithm. Then $\phi(U_k)$ is strictly monotonically decreasing for every $k \in \mathbb{N}$ where $[U_k^T H_0 U_k, N] \neq 0$.*

(iii) *Fixed points of the algorithm are characterised by matrices $U \in O(n)$ such that*

$$[U^T H_0 U, N] = 0.$$

There are exactly $2^n n!$ distinct fixed points.

(iv) *Let U_k, $k = 1, 2, \ldots$, be a solution to the associated orthogonal Lie-bracket algorithm, then U_k converges to an orthogonal matrix U_∞, a fixed point of the algorithm.*

(v) *All fixed points of the associated orthogonal Lie-bracket algorithm are strictly unstable except those 2^n points $U_* \in O(n)$ such that*

$$U_*^T H_0 U_* = \Lambda,$$

where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Such points U_* are locally exponentially asymptotically stable and $H_0 = U_* \Lambda U_*^T$ is an eigenspace decomposition of H_0.

Proof. Part (i) follows directly from the orthogonal nature of $e^{\alpha_k [U_k^T H_0 U_k, N]}$. Note that in part (ii) the definition of ϕ can be expressed in terms of the map $g(U) = U^T H_0 U$ from $O(n)$ to $M(H_0)$ and the Lie-bracket potential $\psi(H) = ||H - N||^2$ of (1), i.e.,

$$\phi(U_k) = \psi(g(U_k)).$$

Observe that $g(U_0) = U_0^T H_0 U_0$ and thus $g(U_k)$ is the solution of the Lie-bracket algorithm with initial condition $U_0^T H_0 U_0$. As the step-size selection scheme α_N is either (13) or (17), then $g(U_k)$ satisfies (7). This ensures that part (ii) holds.

If U_k is a fixed point of the associated orthogonal Lie-bracket algorithm with initial condition $U_0^T H_0 U_0$, then $g(U_k)$ is a fixed point of the Lie-bracket algorithm. Thus, from Proposition 2.1, $[g(U_k), N] = [U_k^T H_0 U_k, N] = 0$. Moreover, if $[U_k^T H_0 U_k, N] = 0$ for some given $k \in \mathbb{N}$, then by inspection $U_{k+l} = U_k$ for $l = 1, 2, \ldots$, and U_k is a fixed point of the associated orthogonal Lie-bracket algorithm. From Lemma 2.2 it follows that if U is a fixed point of the algorithm then $U^T H_0 U = \pi^T \Lambda \pi$ for some permutation matrix π. By inspection any orthogonal matrix $W = SU\pi^T$, where S is a sign matrix $S = \operatorname{diag}(\pm 1, \ldots, \pm 1)$, is also a fixed point of the recursion, and indeed, any two fixed points are related in this manner. A simple counting argument shows that there are exactly $2^n n!$ distinct matrices of this form.

To prove (iv), note that since $g(U_k)$ is a solution to the Lie-bracket algorithm, it converges to a limit point $H_\infty \in M(H_0)$, $[H_\infty, N] = 0$ (Proposition 2.1). Thus U_k must converge to the preimage set of H_∞ via the map g. Condition 6.1 ensures that a set generated by the preimage of H_∞ is a finite distinct set, any two elements U_∞^1 and U_∞^2 of which are related by $U_\infty^1 = U_\infty^2 S$, $S = \operatorname{diag}(\pm 1, \ldots, \pm 1)$. Convergence to a particular element of this preimage follows since $\alpha_k [U_k^T H_0 U_k, N] \to 0$ as in Proposition 2.1.

To prove part (v), observe that the dimension of $O(n)$ is the same as the dimension of $M(H_0)$ due to genericity Condition 6.1. Thus g is locally a diffeomorphism on $O(n)$ that forms an exact equivalence between the Lie-bracket algorithm and the associated orthogonal Lie-bracket algorithm. Restricting g to a local region, the stability structure of equilibria are preserved under the map g^{-1}. Thus, all fixed points of the associated orthogonal Lie-bracket algorithm are locally unstable except those that map via g to the unique locally asymptotically stable equilibrium of the Lie-bracket recursion. Observe that due to the monotonicity of $\phi(U_k)$ a locally unstable equilibrium is also globally unstable. □

THEOREM 6.2. *Let $H_0 \in \mathbb{R}^{m \times n}$ where $m \geq n$ satisfies Condition* 6.1. *Let $N \in \mathbb{R}^{m \times n}$ satisfy Condition* 5.1. *Let the time-step α_k be given by*

$$\alpha_k = \alpha_{\widehat{N}}(\widehat{H}),$$

where $\alpha_{\widehat{N}}$ is either the constant step-size selection (13) *or the variable step-size selection scheme* (17), *on $M(\widehat{H}_0)$. The recursion*

$$V_{k+1} = V_k e^{\alpha_k \{U_k^T H_0^T V_k, N^T\}}, \qquad V_0 \in O(m),$$
$$U_{k+1} = U_k e^{\alpha_k \{V_k^T H_0 U_k, N\}}, \qquad U_0 \in O(n),$$

referred to as the associated orthogonal singular value algorithm, *has the following properties*:

(i) *Let (V_k, U_k) be a solution to the associated orthogonal singular value algorithm, then both V_k and U_k remain orthogonal.*

(ii) *Let $\phi(V,U) = \|V^T H_0 U - N\|^2$ be a map from $O(m) \times O(n)$ to the set of nonnegative reals $\mathbb{R}_+$, then $\phi(V_k, U_k)$ is strictly monotonically decreasing for every $k \in \mathbb{N}$ where $\{V_k^T H_0 U_k, N\} \neq 0$ and $\{U_k^T H_0^T V_k, N^T\} \neq 0$. Moreover, fixed points of the algorithm are characterised by matrix pairs $(V,U) \in O(m) \times O(n)$ such that*

$$\{V^T H_0 U, N\} = 0 \quad \textit{and} \quad \{U^T H_0^T V, N^T\} = 0.$$

(iii) *Let (V_k, U_k), $k = 1, 2, \ldots$, be a solution to the associated orthogonal singular value algorithm, then (V_k, U_k) converges to a pair of orthogonal matrices (V_∞, U_∞), a fixed point of the algorithm.*

(iv) *All fixed points of the associated orthogonal singular value algorithm are strictly unstable except those points $(V_*, U_*) \in O(m) \times O(n)$ such that*

$$V_*^T H_0 U_* = \Sigma,$$

where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{m \times n}$. Each such point (V_*, U_*) is locally exponentially asymptotically stable and $H_0 = V_* \Sigma U_*^T$ is a singular value decomposition of H_0.

Proof. The proof of this theorem is analogous to the proof of Theorem 6.1. □

7. Computational considerations. There are several issues involved in the implementation of the Lie-bracket algorithm as a numerical tool that have not been dealt with in the body of this paper. Design and implementation of efficient code has not been considered and would depend heavily on the nature of the hardware on which such a recursion would be run. As each iteration requires the calculation of a time-step, an exponential and a $k+1$ estimate, it is likely that it would be best to consider applications in parallel processing environments. Certainly in a standard computational environment the exponential calculation would limit the possible areas of useful application of the algorithms proposed.

It is also possible to consider approximations of the Lie-bracket algorithm that have good computational properties. For example, consider a (1,1) Padé approximation to the matrix exponential

$$e^{\alpha_k [H_k, N]} \approx \frac{2I + \alpha_k [H_k, N]}{2I - \alpha_k [H_k, N]}.$$

Such an approach has the advantage that, as $[H_k, N]$ is skew-symmetric, the Padé approximation will be orthogonal and will preserve the isospectral nature of the Lie-bracket algorithm. Similarly, an (n,n) Padé approximation of the exponential for any n will also be orthogonal. There are difficulties involved in obtaining direct step-size selection schemes based on the Padé approximate Lie-bracket algorithms. To guarantee that the potential ψ is monotonic decreasing for such schemes, direct

estimates of time-step must be chosen prohibitively small. A good heuristic choice of a step-size selection scheme, however, can be made based on the selection schemes given in this paper and simulations indicate that such an approach is viable.

Another approach is to take just the linear term from the Taylor expansion of $H_{k+1}(\alpha_k)$,

$$H_{k+1} \approx H_k + \alpha_k[H_k,[H_k,N]],$$

as an algorithm on $\mathbb{R}^{n\times n}$. An algorithm such as this is similar in form to approximating the curves generated by the Lie-bracket algorithm by straight lines. The approximation will not retain the isospectral nature of the Lie-bracket recursion; however, it is computationally cheap. Furthermore, when the curvature of the manifold $M(H_0)$ is small, then it can be imagined that the linear algorithm would be a good approximation to the Lie-bracket algorithm.

8. Conclusion. In this paper we have proposed two algorithms which, along with their associated orthogonal algorithms, calculate respectively, the eigenvalue decomposition of a symmetric matrix and the singular value decomposition of a general matrix. Moreover, we have presented two suitable step-size selection schemes which ensure that, for generic initial conditions, the algorithms proposed will converge exponentially fast to an asymptotically attractive fixed point.

In future work we hope to improve the theoretical understanding of the step-size selection schemes necessary for the Lie-bracket algorithm as well as to investigate a number of related applications of the double-bracket flow and its discretisation.

9. Appendix. The following discussion is a proof of Proposition 4.3.

Proof. By Lemma 4.1, Λ is the unique locally asymptotically stable equilibrium point and it remains to show that Λ is exponentially attractive. Note that direct linearisation techniques do not apply as the recursion will not necessarily be differentiable at the equilibrium Λ. To proceed we set $c = 1/(4||H_0|| \cdot ||N||)$, the constant time-step, and show that the Lie-bracket algorithm converges faster using the variable step-size selection scheme than it does with the constant time-step c. The proof is divided into a number of lemmas.

LEMMA 9.1. *Let $0 < \beta < \min(1, c)$, where $c = 1/(4||H_0|| \cdot ||N||)$. Then there exists a real number δ_1 such that for $H_k \in M(H_0)$ and $|||[H_k, N]|| < \delta_1$, then*

$$0 > \triangle\psi(H_k,\beta) \geq -3\beta|||[H_k,N]|||^2. \tag{33}$$

Proof. Consider the error term $\tau^2 tr(N\mathcal{R}_2(\tau))$ defined in Lemma 3.1 and recall the estimation argument for Lemma 3.3. Employing a similar argument for $\tau = \beta$ gives

$$|\beta^2 \mathrm{tr}(N\mathcal{R}_2(\beta))| \leq ||H_0|| \cdot ||N|| \cdot \left(e^{2\beta|||[H_k,N]|||} - 1 - 2\beta|||[H_k,N]|||\right).$$

Thus, combining this with (9) it follows that

$$\begin{aligned} \triangle\psi(H_k,\beta) &\geq -2\beta|||[H_k,N]|||^2 - 2\beta^2|tr(N\mathcal{R}_2(\beta))| \\ &\geq -2\beta|||[H_k,N]|||^2 - 2||H_0|| \cdot ||N|| \cdot \\ &\qquad \left(e^{2\beta|||[H_k,N]|||} - 1 - 2\beta|||[H_k,N]|||\right). \end{aligned} \tag{34}$$

It is well known that

$$2(e^y - 1 - y) \sim y^2 \quad \text{for } y \to 0^+,$$

where "$\sim$" indicates that two functions are asymptotically equal. This is equivalent to saying that for any $\epsilon > 0$, there exists $\delta(\epsilon) > 0$, such that for all y, where $\delta(\epsilon) > y > 0$, then $1 - \epsilon < 2(e^y - 1 - y)/y^2 < 1 + \epsilon$. Thus, choosing $\epsilon = \frac{1}{4}$, it follows that for $\delta(\frac{1}{4}) > y > 0$ then $2(e^y - 1 - y) < 2y^2$. Recall that we are restricting $\beta < 1$, and thus, there exists some real number $\delta_1 > 0$ such that if $||[H_k, N]|| < \delta_1$, then $2\beta||[H_k, N]|| < \delta(\frac{1}{4})$, and hence $2(e^{2\beta||[H_k,N]||} - 1 - 2\beta||[H_k, N]||) < 4\beta^2||[H_k, N]||^2$. Substituting this into (34) gives

$$\triangle\psi(H_k, \beta) \geq -2\beta||[H_k, N]||^2 - 4\beta^2||H_0|| \cdot ||N|| \cdot ||[H_k, N]||^2. \tag{35}$$

By additionally requiring that $\beta < c = 1/(4||H_0|| \cdot ||N||)$ the lemma is proved. □

LEMMA 9.2. *Let α_N^* be the step-size selection scheme given by Lemma* 3.4, *and let $\gamma \in \mathbb{R}_+$, such that $\alpha_N^*(H_k) > \gamma > 0$ for all $[H_k, N] \neq 0$. Define $\overline{\gamma} := \min\{\gamma, c\}$ and choose $\beta \in \mathbb{R}_+$ such that*

$$0 < \beta < \min\left\{1, c, \frac{2}{3}(\overline{\gamma} - 2||H_0|| \cdot ||N||\overline{\gamma}^2)\right\}.$$

Then there exists a real number $\delta_2 > 0$ such that for any $H_k \in M(H_0)$ with $||[H_k, N]|| < \delta_2$

$$-3\beta||[H_k, N]||^2 > \triangle\psi_U^*(H_k, \alpha_k^*). \tag{36}$$

Proof. Recall that α_k^* was chosen as the first critical point of the function $\triangle\psi_U^*(H_k, \tau)$. Thus $\triangle\psi_U^*(H_k, \tau)$ is monotonic decreasing on the interval $(0, \alpha_k^*)$. The lower bound $\overline{\gamma} < \gamma$, for α_N^*, must be less than α_k^*, and thus $\triangle\psi_U^*(H_k, \overline{\gamma}) > \triangle\psi_U^*(H_k, \alpha_k^*)$. Substituting $\overline{\gamma}$ into the definition of $\triangle\psi_U^*$ gives

$$\begin{aligned}
\triangle\psi_U^*(H_k, \overline{\gamma}) &= -2\overline{\gamma}||[H_k, N]||^2 \\
&\quad + \frac{||H_0|| \cdot ||[N, [H_k, N]]||}{||[H_k, N]||}\left(e^{2\overline{\gamma}||[H_k,N]||} - 1 - 2\overline{\gamma}||[H_k, N]||\right), \\
&\leq -2\overline{\gamma}||[H_k, N]||^2 \\
&\quad + 2||H_0|| \cdot ||N||\left(e^{2\overline{\gamma}||[H_k,N]||} - 1 - 2\overline{\gamma}||[H_k, N]||\right).
\end{aligned}$$

As shown in Lemma 9.1, there exists $\delta_2 > 0$, such that for any $H_k \in M(H_0)$, where $||[H_k, N]|| < \delta_2$, then $2\left(e^{2\gamma||[H_k,N]||} - 1 - 2\gamma||[H_k, N]||\right) < 4\gamma^2||[H_k, N]||^2$. Using this with the above inequality gives

$$\triangle\psi_U^*(H_k, \overline{\gamma}) \leq 2||[H_k, N]||^2\left(2||H_0|| \cdot ||N||\overline{\gamma}^2 - \overline{\gamma}\right).$$

Note that since $\overline{\gamma} < c$, then the right-hand side of the last inequality is strictly negative. Now as

$$0 < \beta < \frac{2}{3}\left(\overline{\gamma} - 2||H_0|| \cdot ||N||\overline{\gamma}^2\right),$$

then $-3\beta||[H_k, N]||^2 > 2||[H_k, N]||^2\left(2||H_0|| \cdot ||N||\overline{\gamma}^2 - \overline{\gamma}\right)$ and the result follows. □

The proof of Proposition 4.3 now follows by choosing

$$\beta = \min\left\{\begin{array}{l} 1, \\ c, \\ \frac{2}{3}\left(\overline{\gamma} - 2||H_0|| \cdot ||N||\overline{\gamma}^2\right), \end{array}\right. \tag{37}$$

where $\overline{\gamma} = \min(\gamma, c)$. Thus, from Lemmas 9.1 and 9.2, choose δ_1 and δ_2 such that the results hold and set $\delta = \frac{1}{2} \min \{\delta_1, \delta_2\}$. Hence, combining the inequalities (33) and (36) gives

$$\triangle\psi(H_k, \alpha_k^*) < \triangle\psi(H_k, \beta) < 0 \tag{38}$$

for all $H_k \in M(H_0)$ with $|||[H_k, N]||| < \delta$.

Let D_δ be some open set around Λ such that $|||[H_k, N]||| < \delta$. Note that $\beta \leq c$, and thus from Lemma 4.2 the Lie-bracket algorithm equipped with $\alpha_N^\beta = \beta$ as a step-size selection scheme is exponentially stable. Finally, note that within D_δ, and due to (38), $\psi(H_{k+1}(\alpha_k^*))$ will always decrease faster than $\psi(H_{k+1}(\beta))$, regardless of H_k. Since Λ is exponentially attractive for the Lie-bracket algorithm equipped with the selection scheme α_N^β, it follows that Λ must also be exponentially attractive for the same recursion equipped with the selection scheme α_N^*. □

Acknowledgments. The authors would like to thank Kenneth Driessel and Wei-Yong Yan for many useful comments. Also the authors thank an anonymous referee for mentioning the connection of the double-bracket flow to micromagnetics as well as a number of useful comments.

REFERENCES

[1] A. M. Bloch, R. W. Brockett, and T. Ratiu, *A new formulation of the generalised Toda lattice equations and their fixed point analysis via the momentum map*, Bull. Amer. Math. Soc., 23 (1990), pp. 477–485.

[2] R. W. Brockett, *Dynamical systems that sort lists, diagonalise matrices and solve linear programming problems*, Linear Algebra Appl., 146 (1991), pp. 79–91; see, also, Proc. IEEE Conf. Decisions and Control, 1988, pp. 799–803.

[3] M. T. Chu, *The generalized Toda flow, the QR-algorithm and the center manifold theory*, SIAM J. Discrete Math., 5 (1984), pp. 187–201.

[4] ———, *A differential equation approach to the singular value decomposition of bidiagonal matrices*, Linear Algebra Appl., 80 (1986), pp. 71–80.

[5] M. T. Chu and K. R. Driessel, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM J. Numer. Anal., 27 (1990), pp. 1050–1060.

[6] P. Deift, T. Nanda, and C. Tomei, *Ordinary differential equations for the symmetric eigenvalue problem*, SIAM J. Numer. Anal., 20 (1983), pp. 1–22.

[7] K. R. Driessel, *On isospectral gradient flows–solving matrix eigenproblems using differential equations*, in Inverse Problems, J. R. Cannon and U. Hornung, eds., Birkhauser-Verlag, 1986, pp. 69–90.

[8] U. Helmke and J. B. Moore, *Singular value decomposition via gradient flows*, Systems Control Lett., 14 (1990), pp. 369–377.

[9] ———, *Optimization and dynamical systems*, in Communications and Control Engineering, Springer-Verlag, London, 1994.

[10] J. Imae, J. Perkins, and J. Moore, *Towards time varying balanced realisation via Riccati equations*, Math. Control Signals Systems, 5 (1992), pp. 313–326.

[11] J. E. Perkins, U. Helmke, and J. B. Moore, *Balanced realizations via gradient flow techniques*, Systems Control Lett., 14 (1990), pp. 369–380.

[12] S. T. Smith, *Dynamical systems that perform the singular value decomposition*, Systems Control Lett., 16 (1991), pp. 319–327.

[13] W. W. Symes, *The QR algorithm and scattering for the finite nonperiodic Toda lattice*, Phys. D4, (1982), pp. 275–280.

[14] D. Watkins and L. Elsner, *Self-equivalent flows associated with the singular value decomposition*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 244–258.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 903–908, July 1994

012

A NOTE ON EXTREME CORRELATION MATRICES*

CHI-KWONG LI† AND BIT-SHUN TAM‡

Abstract. An $n \times n$ complex Hermitian or real symmetric matrix is a correlation matrix if it is positive semidefinite and all its diagonal entries equal one. The collection of all $n \times n$ correlation matrices forms a compact convex set. The extreme points of this convex set are called extreme correlation matrices. In this note, elementary techniques are used to obtain a characterization of extreme correlation matrices and a canonical form for correlation matrices. Using these results, the authors deduce most of the existing results on this topic, simplify a construction of extreme correlation matrices proposed by Grone, Pierce, and Watkins, and derive an efficient algorithm for checking extreme correlation matrices.

Key words. correlation matrix, extreme point, perturbation, rank, linear span

AMS subject classification. 15A48

Let $\mathcal{M} = \mathcal{H}_n$ or $\mathcal{S}_n$, the real linear space of all $n \times n$ Hermitian matrices and the real linear space of all $n \times n$ symmetric matrices, respectively. A positive semidefinite matrix $A = (a_{ij}) \in \mathcal{M}$ with $a_{11} = \cdots = a_{nn} = 1$ is called a *correlation matrix.* The term correlation matrix comes from statistics, where the entries of a real correlation matrix occur as correlations between pairs of random variables. It is easy to see that the collection of $n \times n$ correlation matrices forms a compact convex set, and we are interested in its extreme points. Recall that an element x in a convex set S is an *extreme point* if $x = ty + (1-t)z$ for $y, z \in S$ and $0 < t < 1$ implies $y = z = x$, that is, if x can be a convex combination of points of S in only trivial ways.

We shall call an extreme point of the set of correlation matrices an *extreme correlation matrix.* This concept has been studied in [1], [4], and [2]. In those papers, different approaches were used to determine all possible ranks of extreme correlation matrices, to construct extreme correlation matrices of different ranks, and to give simple characterizations of extreme correlation matrices in low dimensional cases. In this note, we use an elementary approach to prove several results on the subject. Using our results, one can deduce easily all of the main results in the three papers mentioned above. Moreover, we simplify a construction of extreme correlation matrices proposed in [2], derive an efficient algorithm for checking extreme correlation matrices, and compare our condition with the one given in [4]. A question posed in [2] is also discussed.

In the following we shall concentrate mainly on the Hermitian case, the slightly more difficult case. For the real case, we also give some results that have no analogs in the Hermitian case.

1. Basic results. Given an $n \times n$ correlation matrix A, a nonzero Hermitian matrix B is said to be a *perturbation* of A if $A \pm tB$ are correlation matrices for some

*Received by the editors March 22, 1992; accepted for publication (in revised form) November 20, 1992.

†Department of Mathematics, The College of William and Mary, Williamsburg, Virginia 23187 (ckli@cs.wm.edu). The work of this author was supported in part by National Science Foundation grant DMS 91-00344.

‡Department of Mathematics, Tamkang University, Tamsui, Taiwan 25137, Republic of China (bsm01@twntku10.bitnet). The work of this author was supported by the National Science Council of the Republic of China.

(and hence for all sufficiently small) $t > 0$. Clearly, A is not extreme if and only if A has a perturbation. In fact, if $A = (A_1 + A_2)/2$ for two distinct correlation matrices A_1 and A_2, then $B := (A_1 - A_2)/2$ is such that $A \pm B$ are correlation matrices. We give a characterization of perturbations of a given correlation matrix and a characterization of extreme correlation matrices in the following theorem.

THEOREM 1. *Let $A \in \mathcal{H}_n$ be an $n \times n$ correlation matrix of rank r. Suppose that $A = XQX^*$, where $X \in \mathbb{C}^{n\times r}$ and $Q \in \mathcal{H}_r$. Then*

(a) *$B \in \mathcal{H}_n$ is a perturbation of A if and only if all diagonal entries of B equal zero and $B = XRX^*$ for some nonzero $R \in \mathcal{H}_r$, and*

(b) *A is extreme if and only if*

$$\operatorname{span}\{x_j x_j^* : 1 \le j \le n\} = \mathcal{H}_r,$$

where x_j is the jth column of X^*.

Proof. With the notation of the theorem, one sees easily that $\operatorname{rank} X = r$ and Q is positive definite.

(a) Let B be a matrix with all diagonal entries equal to zero and of the form XRX^* for some nonzero $R \in \mathcal{H}_r$. Since X is a matrix of full column rank, we have $\operatorname{rank} XRX^* = \operatorname{rank} R > 0$. Thus B is nonzero. Since Q is positive definite, $Q \pm tR$ is positive semidefinite for all sufficiently small $t > 0$. It follows that B is a perturbation of A. Conversely, suppose B is a perturbation of A. Evidently, all diagonal entries of B equal zero. Append to X an $n \times (n - r)$ matrix Y such that the $n \times n$ matrix $V = (X|Y)$ is nonsingular. Clearly A can be expressed as $V(Q \oplus 0_{n-r})V^*$. Write B as VCV^*, where $C \in \mathcal{H}_n$. Partition C in the same way as $Q \oplus O_{n-r}$. Since B is a perturbation of A, $(Q \oplus O_{n-r}) \pm tC$ is positive semidefinite for some $t > 0$. It follows that except for its $(1,1)$ block, which we denote by R, the blocks of C are all zero. Hence B is of the form XRX^*, with $0 \ne R \in \mathcal{H}_r$.

(b) Since A is not extreme if and only if it has a perturbation, by part (a) one sees that A is extreme if and only if for any $R \in \mathcal{H}_r$, $R = 0$ whenever all diagonal entries of XRX^* equal zero. In terms of the usual inner product on $\mathcal{H}_r$ defined by $\langle X, Y\rangle = tr(XY)$, we can reformulate the last condition as: $R = 0$ whenever $\langle R, x_j x_j^*\rangle = 0$ for all j, $1 \le j \le n$; or equivalently, $\operatorname{span}\{x_j x_j^* : 1 \le j \le n\} = \mathcal{H}_r$. Thus our result follows. □

Notice that for a given rank r positive semidefinite matrix A, there are two standard ways to decompose it as XQX^*. One way is to take Q to be a diagonal matrix whose diagonal entries are all the nonzero eigenvalues of A and form the matrix X whose columns are the corresponding eigenvectors. Another way is to find X such that $A = XX^*$, i.e., to take $Q = I_r$. In both cases, there are standard algorithms and computer programs to do the decomposition.

From Theorem 1 and its real analog, one easily deduces the following result proved in [1] (for the Hermitian case) and [2] (for the real case).

COROLLARY 2. *If A is an $n \times n$ extreme Hermitian (respectively, real symmetric) correlation matrix of rank r, then $r^2 \le n$ (respectively, $r(r+1) \le 2n$).*

In [4] and [2] it is shown that if r satisfies the inequality in the corollary, then there exists a rank r extreme correlation matrix. One can verify the constructions of extreme correlation matrices in those papers using our Theorem 1. We shall suggest a construction after proving Theorem 3. To state the result, we need the following definition and notation. A matrix is a (real) generalized permutation matrix if it is a unitary (respectively, real orthogonal) matrix with exactly one nonzero entry in each row and each column. Denote by $J_{r,s}$ the $r \times s$ matrix all of whose entries equal 1. For simplicity we use J_r to represent $J_{r,r}$.

THEOREM 3. *Let $A \in \mathcal{H}_n$. Then A is a correlation matrix if and only if there exists a generalized permutation matrix P such that $PAP^* = (B_{st})$, a $p \times p$ block matrix with $B_{st} = b_{st} J_{k(s),k(t)}$, $(k(1)+\cdots+k(p) = n)$, where $(b_{st}) \in \mathcal{H}_p$ is a correlation matrix all of whose off-diagonal entries have moduli less than one. Furthermore, we have*

(a) $\operatorname{rank} A = \operatorname{rank} (b_{st})$;

(b) *A is extreme if and only if (b_{st}) is extreme.*

Proof. To prove the "only if" part, express A in the form XX^*, where $X \in \mathbb{C}^{n \times r}$ and $r = \operatorname{rank} A$. Denote the jth column of X^* by x_j. Since each diagonal entry of A is equal to 1, each x_j is a vector of unit length. Now permute the rows of X and then multiply each row with a suitable scalar of absolute value one so that rows of X that differ by unit multiples are grouped together and become equal. The resulting effect on X is equivalent to applying a generalized permutation similarity to A. Since the inner product between any two linearly independent unit vectors always has modulus less than one, A is transformed to the required form. That (b_{st}) is a correlation matrix follows from the observation that it is a principal submatrix of A of the required form.

To prove the "If" part, suppose $A = (B_{st})$ as described in the theorem. It is clear that $A \in \mathcal{H}_n$ and its main diagonal entries are all equal to one. Note that we can write $J_{k(s),k(t)}$ as $\tilde{e}_{k(s)} \tilde{e}_{k(t)}^t$, where $\tilde{e}_j$ denotes the $j \times 1$ vector of all 1's. If $x_j \in \mathbb{C}^{k(j)}$ for $j = 1, \ldots, p$, then by direct calculations, the value of the quadratic form of A at x with $x^* = (x_1^*, \ldots, x_p^*)$ is equal to the value of the quadratic form of $(b_{st}) \in \mathcal{H}_p$ at $\tilde{x} = (\tilde{e}_{k(1)}^t x_1, \ldots, \tilde{e}_{k(p)}^t x_p)^t$ and so is nonnegative.

It is not difficult to show that $\operatorname{rank} A = \operatorname{rank} (b_{st})$. That A is extreme if and only if (b_{st}) is extreme follows readily from Theorem 1. □

Notice that the real analog of Theorem 3 also holds. To obtain the statement and the proof for the real case, one only needs to replace $\mathcal{H}_n$ by $\mathcal{S}_n$, generalized permutation matrices by real generalized permutation matrices, complex scalars by real scalars, etc.

Notice that the matrix (B_{st}) in Theorem 3 is a *block Kronecker product* of the matrices (b_{st}) and $(J_{k(s),k(t)})$. We refer the readers to [3] and its references for the definition and properties of this product.

2. A construction and an algorithm. There are at least two ways that Theorem 3 can help to study extreme correlation matrices. First, it helps to reduce the dimension of a problem under consideration. Second, if one can find an $n \times n$ rank r extreme correlation matrix, then one can use Theorem 3 to construct an $m \times m$ rank r extreme correlation matrices for any $m \geq n$. We illustrate the latter idea by describing a construction of extreme correlation matrices. (Note that this construction is a modification of the one given in [2].)

2.1. Construction of extreme correlation matrices. By the preceding discussion, for a given r it suffices to construct an $n \times n$ rank r extreme correlation matrix for $n = r^2$ in the Hermitian case, and for $n = r(r+1)/2$ in the real case. Then one can get an $m \times m$ rank r extreme correlation matrix for any $m \geq n$. We shall again use e_j to denote the jth column of I_r.

For the Hermitian case, assume $n = r^2$. Set $A = XX^*$ with $X \in \mathbb{C}^{n \times r}$ such that the first r columns of X^* form I_r, the next $r(r-1)/2$ columns consist of vectors of the form $(e_s + e_t)/\sqrt{2}$ with $1 \leq s < t \leq r$, and the rest of the $r(r-1)/2$ columns consist of vectors of the form $(e_s + ie_t)/\sqrt{2}$. Using Theorem 1, one verifies readily that $A \in \mathcal{H}_n$ is an extreme correlation matrix.

For the real case, assume $n = r(r+1)/2$. Let $\tilde{A} = \tilde{X}\tilde{X}^t$ where $\tilde{X}$ is obtained

from X constructed in the Hermitian case by deleting the last $r(r-1)/2$ rows. Again by Theorem 1, one can show easily that $\tilde{A} \in \mathcal{S}_n$ is an extreme correlation matrix.

2.2. An algorithm for checking extreme correlation matrices. By Theorems 1, 3 (and its proof), and Corollary 2, one derives readily the following algorithm to determine whether a given Hermitian correlation matrix is extreme. A similar algorithm also holds for the real case.

Step 1. Express A as XX^*, where $X \in \mathbb{C}^{n\times r}$, $r = \operatorname{rank} A$.

Step 2. Form a matrix Y from the distinct (up to unit multiples) rows of X. Say $Y \in \mathbb{C}^{p\times r}$. (Then YY^* is equal to the matrix (b_{st}) as given in Theorem 3.)

Step 3. Determine $\operatorname{rank} Y$. If $\operatorname{rank} Y = r$ satisfies $r^2 > p$, then A is not extreme. Otherwise, proceed to Step 4.

Step 4. Determine the dimension of $\operatorname{span}\{y_j y_j^* : 1 \le j \le p\}$, where y_j is the jth column of Y^*. It is r^2 if and only if A is extreme.

An efficient way to perform Step 4 is to construct a $p \times r^2$ matrix F as follows. For each j between 1 and p, the first r entries of the jth row of F are $|y_{j1}|^2, |y_{j2}|^2, \dots, |y_{jr}|^2$, arranged in the natural order, and its remaining $r^2 - r$ entries $y_{jk}\overline{y}_{jl}, \overline{y}_{jk}y_{jl}$, $1 \le k < l \le r$ (indexed by ordered pairs (k,l), and with conjugate entries adjacent) are arranged in the usual lexicographic order. Then $\operatorname{rank} F = \dim \operatorname{span}\{y_j y_j^* : 1 \le j \le p\}$.

Explanation. Consider the following real subspace of $\mathbb{C}^{r^2}$:

$$W = \{(t_1, \dots, t_{r^2})^t : t_j \in \mathbb{R}, \quad j = 1, \dots, r; \qquad \text{and}$$
$$t_{r+2m-1} = \overline{t}_{r+2m}, \quad m = 1, \dots, (r^2 - r)/2\}.$$

Note that the real span of the row vectors of F is included in W, and is isomorphic with the subspace of $\mathcal{H}_r$ spanned by $\{y_j y_j^* : 1 \le j \le p\}$. But any set of vectors in W that is linearly independent over $\mathbb{R}$ is also linearly independent over $\mathbb{C}$, so $\operatorname{rank} F = \dim \operatorname{span}\{y_j y_j^* : 1 \le j \le p\}$.

Notice that the equivalent condition in [4] for an extreme correlation matrix can also be deduced readily as follows. Denote by f_j^t the jth row of F. Note that the vectors $f_1, \dots, f_p$ all lie in the (real) hyperplane $\{f = (f_1, \dots, f_{r^2})^t \in W : \sum_{i=1}^{r} f_j = 1\}$ of W (since the row vectors of Y are of unit length, as YY^* is a correlation matrix). But this hyperplane does not contain the zero vector, so we have

$$\begin{aligned}\dim \operatorname{span}\{f_j : 1 \le j \le p\} &= 1 + \dim \operatorname{span}\{f_j - f_p : 1 \le j \le p-1\} \\ &= 1 + \dim \operatorname{span}\{f_j - f_{j+1} : 1 \le j \le p-1\}.\end{aligned}$$

Denote by D_A the $(p-1)\times r^2$ matrix whose jth row is $(f_j - f_{j+1})^t$. Then A is extreme if and only if $r^2 = \dim \operatorname{span}\{f_j : 1 \le j \le p\} (= \operatorname{rank} F)$ if and only if $\operatorname{rank} D_A = r^2 - 1$, which is the condition given in [4]. (In [4] the matrix D_A is obtained from the matrix X instead of from Y. But this does not affect our argument.)

3. Further results. We first consider two results that are valid only for the real case.

COROLLARY 4. *Let $A \in \mathcal{S}_n$ be a rank two correlation matrix. Suppose P is a real generalized permutation matrix such that PAP^t is equal to (B_{st}), a $p \times p$ block matrix that satisfies the conditions as given in Theorem* 3. *Then A is extreme if and only if $p \ge 3$.*

Proof. "Only if" part. Since $\operatorname{rank} A = \operatorname{rank}(b_{st})$, p cannot be 1. If $p = 2$, then (b_{st}) is not extreme since it is nonsingular, and hence A is also not extreme according to Theorem 3.

"If" part. Suppose that A is not extreme. Since $\operatorname{rank} A = 2$, each (relative) boundary point of the face (of the set of $n \times n$ correlation matrices) generated by A is a matrix of rank one. So there exist two rank one correlation matrices A_1, A_2 such that $A = \lambda A_1 + (1-\lambda)A_2$ for some λ with $0 < \lambda < 1$. By applying a generalized permutation similarity to A, we may assume that $A_1 = J_n$ and

$$A_2 = \begin{pmatrix} J_k & -J_{k,n-k} \\ -J_{k,n-k} & J_{n-k} \end{pmatrix}$$

for some k between 1 and $n-1$. But then we have

$$A = \begin{pmatrix} J_k & \alpha J_{k,n-k} \\ \alpha J_{n-k,k} & J_{n-k} \end{pmatrix},$$

where $\alpha = \lambda + (-1)(1-\lambda)$ is of absolute value less than one. So in this case, $p = 2$. □

COROLLARY 5. *A 3×3 real symmetric correlation matrix of rank two is extreme if and only if its off-diagonal entries all have absolute values less than one.*

Two remarks are in order. First, by Corollaries 4 and 5, one sees that a rank two correlation matrix $A \in \mathcal{S}_n$ is extreme if and only if A has a principal submatrix that is an extreme 3×3 correlation matrix.

Second, the "if" parts of Corollaries 4 and 5 are both invalid in the Hermitian case. Indeed, for any $n \geq 2$, if we take A_n to be the matrix $(J_n + uu^*)/2$, where $u = (1, \mu, \ldots, \mu^{n-1})^t$, μ a primitive nth root of unity, then A_n is a nonextreme Hermitian correlation matrix of rank two, all of whose off-diagonal entries have moduli less than one.

By Theorem 1, we have the following observation for rank two correlation matrices in $\mathcal{H}_n$.

OBSERVATION. *Suppose $A = XQX^* \in \mathcal{H}_n$ with $X \in \mathbb{C}^{n\times 2}$ and $Q \in \mathcal{H}_2$ is a rank two correlation matrix. Let S be a 2×2 nonsingular submatrix of X^*. Then A is extreme if and only if there are two column vectors $u = (u_1, u_2)^t$ and $v = (v_1, v_2)^t$ of the matrix $S^{-1}X^*$, such that $\overline{u}_1 u_2$ and $\overline{v}_1 v_2$ are complex numbers that are not nonzero real multiples of each other.*

In the lemma in [2], it was shown that an equivalent condition for a real symmetric correlation matrix to be extreme is that its nullspace is maximal among the nullspaces of all correlation matrices. (The corresponding result for the Hermitian case also holds.) Clearly another equivalent condition is that the range space of the matrix is minimal among the range spaces of all correlation matrices. In [2] the authors also posed the question of determining the structure of the nullspace of a correlation matrix. Below we give an answer to the dual question of characterizing the linear subspaces of $\mathbb{C}^n$ (also $\mathbb{R}^n$) that can be the range space of a correlation matrix.

THEOREM 6. *A subspace of $\mathbb{C}^n$ (or $\mathbb{R}^n$) is the range space of a correlation matrix if and only if it has a basis (or a spanning set) $\{v_1, \ldots, v_r\}$ such that $\sum_{j=1}^r v_j \circ \overline{v}_j = (1, \ldots, 1)^t \in \mathbb{R}^n$, where $\overline{x}$ denotes the complex conjugate of the vector x, and $x \circ y$ denotes the Schur (Hadamard/entrywise) product of x and y.*

Proof. Suppose W is the range space of the correlation matrix A. Let $A = XX^*$ with $X \in \mathbb{C}^{n\times r}$, where $r = \operatorname{rank} A$. Then the columns of X form a basis for W that satisfies the required properties.

Conversely, if W is a subspace that has a spanning set as described in the theorem, then $A = XX^*$, where the columns of $X \in \mathbb{C}^{n\times r}$ are the vectors from the spanning set, is the required correlation matrix. □

COROLLARY 7. *A subspace of* $\mathbb{C}^n$ *(or* $\mathbb{R}^n$*) is the nullspace of a correlation matrix if and only if its orthogonal complement has a spanning set satisfying the condition in Theorem* 6.

Note added in proof. After the paper had been accepted for publication, the authors found that a slight modification of the proof of Theorem 1 yields the following result that covers [1, Thm. 3].

THEOREM 8. *Under the hypotheses and notation of Theorem* 1, *the face of the convex set of* $n \times n$ *correlation matrices generated by* A *is of dimension*

$$r^2 - \dim \operatorname{span} \{x_j x_j^* : 1 \leq j \leq n\}.$$

Proof. It is clear that the dimension of the face generated by A is equal to the dimension of the space generated by the perturbations of A. According to Theorem 1(a) (or its proof), a nonzero matrix is a perturbation of A if and only if it is of the form XRX^*, where X is $n \times r$ and R is $r \times r$ lying in the orthogonal complement of $\operatorname{span}\{x_j x_j^* : 1 \leq j \leq n\}$. Since X has full column rank, the mapping $R \mapsto XRX^*$ is a linear isomorphism. □

Acknowledgment. The authors wish to thank Dr. H.J. Woerdeman for drawing their attention to this subject. Thanks are also due to Dr. R. Horn and the referees for their helpful comments that improved our exposition.

REFERENCES

[1] J. P. R. CHRISTENSEN AND J. VESTERSTRØM, *A note on extreme positive definite matrices*, Math. Ann., 244 (1979), pp. 65–68.

[2] R. GRONE, S. PIERCE, AND W. WATKINS, *Extremal correlation matrices*, Linear Algebra Appl., 134 (1990), pp. 63–70.

[3] R.A. HORN AND R. MATHIAS, *Block-matrix generalizations of Schur's basic theorems on Hadamard products*, Linear Algebra Appl., 172 (1992), pp. 337–346.

[4] R. LOEWY, *Extreme points of a convex subset of the cone of positive semidefinite matrices*, Math. Ann., 253 (1980), pp. 227–232.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 909–921, July 1994

013

ON PRECONDITIONING FOR FINITE ELEMENT EQUATIONS ON IRREGULAR GRIDS*

ALISON RAMAGE† AND ANDREW J. WATHEN‡

Abstract. Preconditioning methods are widely used in conjunction with the conjugate gradient method for solving large sparse symmetric linear systems arising from the discretisation of selfadjoint linear elliptic partial differential equations. Many different preconditioners have been proposed, and they are generally analysed and compared using model problems: simple discretisations of Laplacian operators on regular computational grids, generally in two space dimensions. For such model problems there are highly competitive multigrid methods, and it is principally for geometrically irregular (nonmodel) problems that the applicability and economy of preconditioned conjugate gradient methods are most useful. This is particularly true for problems on irregular unstructured three-dimensional grids.

This paper is concerned with the comparison of preconditioners for finite element discretisations of three-dimensional selfadjoint elliptic problems on irregular and unstructured computational grids. It is argued that simple preconditioners, which are inferior for regular grid problems in two dimensions, are competitive for irregular grid problems in three dimensions.

Key words. finite elements, irregular grids, preconditioned conjugate gradients

AMS subject classifications. 65N30, 65F10, 65N50

1. Introduction.

When faced with choosing a method for solving a partial differential equation on an irregular domain, one possibility is certainly the finite element method. This is a well established and widely used technique that has many attractive approximation properties, e.g., [20]. As with other numerical methods for solving partial differential equations, using the finite element method involves some form of discretisation of the problem domain. A basic decision must therefore be made: Should this be done in a regular or irregular way? It is true that problems on oddly shaped domains can be tackled by using very fast methods on regular finite element grids that are highly refined near irregular features or perhaps on a series of superimposed regular grids. Nevertheless, practical problems may well be best modelled using unstructured grids: for example, if irregular geometries are fitted exactly they are often easier to deal with in terms of grid visualisation. Furthermore, an irregular discretisation will frequently have the practical advantage of covering the whole domain in a smaller number of elements than will a regular grid. In the light of such observations, here we consider the topic of solving finite element equations with particular reference to unstructured grids.

Applying the Galerkin finite element method to a second order selfadjoint elliptic partial differential equation gives rise to a linear system

$$A\mathbf{x} = \mathbf{b}, \tag{1.1}$$

where the real coefficient matrix A is symmetric, positive definite, and, in most practical applications, large and sparse. As the direct solution of such a system can be

* Received by the editors November 4, 1991; accepted for publication (in revised form) January 26, 1993.

† School of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom. Present address, Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, United Kingdom. The work of this author was supported by the Science and Engineering Research Council of Britain and Nuclear Electric plc. grant GR/G11309.

‡ School of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom (andy.wathen@bristol.ac.uk).

prohibitively expensive, one popular approach is to use the iterative *preconditioned conjugate gradient* (PCG) method (see, e.g., Concus, Golub, and O'Leary [5]).

The construction of appropriate preconditioners for use with this method is a popular topic of current research and a wide range of preconditioners have been developed that work very efficiently on regular grids, particularly in two dimensions (see, e.g., [3], [8]). However, the effect on the performance of a preconditioner of using an irregular finite element grid is one research area that is often neglected. In the literature, the most common comparison of the performance of various preconditioners is on regular grid problems in two dimensions, but results in such cases do not always translate to irregular geometries (or three dimensions). Developing rigorous theory for irregular problems has proved to be much more difficult than for analogous regular grids and, as a result, many aspects of working with irregular finite element geometries are less understood.

In this paper we adopt an algebraic representation of the finite element process that has been used previously by the second author to derive various properties of finite element matrices [25], [26]. Its general nature means that it can be particularly helpful in the analysis of irregular grid problems. We begin by extending the work of the second author to find easily computed eigenvalue bounds for a class of finite element matrices on irregular grids. In addition, we obtain (weak) bounds on the interior eigenvalues of a general finite element matrix. These can be useful for predicting the rate of convergence of the PCG method (see, e.g., [18], [22]). In §3 we discuss asymptotic estimates for the work involved in implementing some preconditioning methods. This introduces different considerations when choosing a preconditioner for problems on three-dimensional irregular grids. Section 4 contains a result concerning the effect of a certain class of element-based preconditioners on the matrix condition number, which is again relevant to the rate of PCG convergence. Finally, we present a numerical comparison of three PCG methods on some highly irregular three-dimensional finite element grids.

2. Eigenvalue bounds. To analyse irregular grids we introduce an algebraic representation of finite element matrices. Suppose that the finite element grid contains elements $e = 1, \ldots, E$ with V_e local unknowns on each element, giving a total of N global unknowns. The Galerkin finite element coefficient matrix A can be written as

$$A = L^T[A_e]L, \tag{2.1}$$

where $[A_e]$ represents an $[E.V_e \times E.V_e]$ block diagonal matrix whose $[V_e \times V_e]$ diagonal blocks are the element-calculated coefficient matrices. L is an $[E.V_e \times N]$ Boolean connectivity matrix that contains all the necessary information about the grid structure [26]. This notation applies to any grid regardless of irregularity and type of element used, as all connectivity information is stored in the Boolean matrix L.

The rate of convergence of the PCG method depends on the eigenvalue distribution of the preconditioned coefficient matrix (see, e.g., [9], [18], [22]), and the standard estimate for the number of iterations required to achieve a certain level of convergence is usually given in terms of the matrix condition number. For a symmetric positive definite matrix, this is the ratio of its largest eigenvalue to smallest eigenvalue. A useful result of Wathen [25], then, is that if $\{B_e\}$ is any set of $[V_e \times V_e]$ symmetric positive definite matrices and a preconditioner B is formed from

$$B = L^T[B_e]L, \tag{2.2}$$

then the eigenvalues of the global matrix $B^{-1}A$ must satisfy

$$\min_e \lambda_{\min}(B_e^{-1}A_e) \leq \lambda(B^{-1}A) \leq \max_e \lambda_{\max}(B_e^{-1}A_e), \tag{2.3}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the extreme eigenvalues of the preconditioned element matrix $B_e^{-1}A_e$.

The bounds given by (2.3) apply to any grid regardless of irregularity. The assembly notation (2.1) has been used by the second author to give eigenvalue bounds for diagonal scaling of mass and stiffness matrices using various elements on regular grids in two and three dimensions [24], [25]. Here we extend this analysis to irregular grids using the example of a two-dimensional grid of linear triangles with preconditioner $B = \text{diag}A$. By examining the diagonally scaled element matrices and applying (2.3), we can obtain global eigenvalue bounds for both the finite element mass and stiffness matrices.

For an arbitrary triangle of area S_e with angles θ_i and opposite sides of length s_i ($i = 1, 2$ or 3), respectively, the element mass matrix is

$$M_e = \frac{S_e}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \tag{2.4}$$

(see, e.g., [3, p. 259]). When diagonal scaling is applied, the eigenvalues of $D_e^{-1}M_e$ (where $D_e = \text{diag}M_e$) are $\frac{1}{2}$, $\frac{1}{2}$, and 2, which means that the eigenvalues of the diagonally scaled global mass matrix all lie in the interval $[\frac{1}{2}, 2]$.

The element stiffness matrix is

$$K_e = \frac{1}{2}\begin{bmatrix} \cot\theta_2 + \cot\theta_3 & -\cot\theta_3 & -\cot\theta_2 \\ -\cot\theta_3 & \cot\theta_1 + \cot\theta_3 & -\cot\theta_1 \\ -\cot\theta_2 & -\cot\theta_1 & \cot\theta_1 + \cot\theta_2 \end{bmatrix} \tag{2.5}$$

(see, e.g., [3, p. 415]). The eigenvalues of $D_e^{-1}K_e$ (where $D_e = \text{diag}K_e$) can be found algebraically to be

$$\begin{Bmatrix} 0 \\ \frac{3}{2} \pm \frac{1}{2}\left\{9 - 8\left[\frac{S_e \sin 2\theta_i}{s_i^2} + \sin^2\theta_i\right]\right\}^{1/2} \end{Bmatrix} \tag{2.6}$$

(see the Appendix). The minimum eigenvalue (and hence the lower bound in (2.3)) will always be zero because the element stiffness matrix is singular. For an upper bound, the maximum value of (2.6) occurs on the triangle in the grid that contains the largest angle. This means that for any finite element grid, a global upper eigenvalue bound can be found from (2.3) by identifying the triangle that contains the largest angle and finding the appropriate element matrix eigenvalues. In the regular grid case this is trivial as all elements (and hence all element matrices) are the same. It is not clear how easy it would be to identify such an element on a very large irregular grid. Note that the minimum value of the maximum eigenvalue in (2.6) occurs when the grid is "most regular"; that is, when all triangles are equilateral (the eigenvalues will be 0, $\frac{3}{2}$, and $\frac{3}{2}$). This result reflects the intuitive view that it is desirable to distort the elements of any grid as little as possible, which is the philosophy behind Delaunay triangularisation (see, e.g., [13]). It is also compatible with the angle condition for triangles given by Strang and Fix [20, p. 106] to guarantee the uniform finite element

approximation of derivatives. Because of the general nature of (2.3), analogous results could be derived for many other element types.

In addition to this estimate of the extreme eigenvalues, we can obtain bounds on the interior eigenvalues of a general finite element matrix using the above notation with the Cauchy Interlace Theorem (see, e.g., [17]). We recall that we have chosen the element matrices B_e to be symmetric and positive definite for $e = 1, \ldots, E$. Since the $[E.V_e \times N]$ matrix L is always of full rank, we can extend L to be an $[E.V_e \times E.V_e]$ nonsingular matrix $[L|J]$ with $L^T[B_e]J = 0$. Thus we can define

$$\widetilde{A} = [L|J]^T[A_e][L|J] = \begin{bmatrix} A & A_1 \\ A_1^T & A_2 \end{bmatrix}, \tag{2.7}$$

where $A_1 = L^T[A_e]J$ and $A_2 = J^T[A_e]J$ and

$$\widetilde{B} = [L|J]^T[B_e][L|J] = \begin{bmatrix} B & 0 \\ 0 & B_2 \end{bmatrix}, \tag{2.8}$$

where $B_2 = J^T[B_e]J$ is symmetric and positive definite.

Now,

$$(\widetilde{A} - \mu\widetilde{B}) \text{ is singular} \Leftrightarrow [L|J]^T([A_e] - \mu[B_e])[L|J] \text{ is singular}, \tag{2.9}$$

which implies that the eigenvalues of $\widetilde{B}^{-1}\widetilde{A}$ are precisely the eigenvalues of the element preconditioned matrices $B_e^{-1}A_e, e = 1, \ldots, E$. That is, the eigenvalues $\mu_1 \leq \cdots \leq \mu_{E.V_e}$ say, of $\widetilde{B}^{-1}\widetilde{A}$ are all obtainable from simple calculations on the elements. Although the number of elements may be large, each calculation will involve finding the eigenvalues of a $V_e \times V_e$ matrix where, in practice, V_e is usually small. Thus the computational complexity of such an operation would be linear in the number of elements. Furthermore, such independent calculations are obviously suited to parallel computation. On a regular grid, all element matrices will be identical so the calculation of the μ_e in this case would be trivial. Now

$$\widetilde{B}^{-1/2}\widetilde{A}\widetilde{B}^{-1/2} = \begin{bmatrix} B^{-1/2}AB^{-1/2} & B^{-1/2}A_1B_2^{-1/2} \\ B_2^{-1/2}A_1^TB^{-1/2} & B_2^{-1/2}A_2B_2^{-1/2} \end{bmatrix} \begin{matrix} \} & N, \\ \} & E.V_e - N, \end{matrix} \tag{2.10}$$

and hence the Cauchy Interlace Theorem [17] can be directly applied to give

$$\mu_i \leq \lambda_i \leq \mu_{i+E.V_e-N} \tag{2.11}$$

for $i = 1, \ldots, N$, where $\lambda_1 \leq \cdots \leq \lambda_N$ are the eigenvalues of $B^{-1}A$.

The eigenvalue bounds (2.11) are usually weak since N is often significantly less than $E.V_e$. Nevertheless, the result has some useful and obvious applications. For example, on an irregular grid with a small number, p say, of "bad" (highly distorted) elements that give rise to p relatively small eigenvalues $\mu_i \ll \mu_{p+1}$ for $i = 1, \ldots, p$, (2.11) implies that there are at most p relatively small eigenvalues of the preconditioned global matrix $B^{-1}A$.

3. Asymptotic work estimates. As stated above, many preconditioners that are used with the PCG method have attractive theoretical convergence estimates for two-dimensional second order elliptic problems [11], [27]. However, extension of these and other methods to three dimensions can involve difficulties with regard to practical

implementation, although theoretical convergence estimates may still be good. On irregular grids in particular, the data structures used by these methods can become increasingly complicated. This and other such considerations when solving very large problems on irregular grids raise an important question: How much work is it practical to do before the gain of accelerated conjugate gradient convergence is outweighed by the amount of work involved in constructing the preconditioner itself?

Consider a second order partial differential equation in three dimensions. By solving such a problem on a regular grid with n nodes in each coordinate direction a distance h apart, giving $N = n^3$ unknowns in total, it is possible to calculate an asymptotic order estimate for the amount of work involved in implementing a certain preconditioner as $N \to \infty$ (or equivalently as $h \to 0$). We note that the condition number of the finite element stiffness matrix with appropriate boundary conditions is $O(h^{-2}) = O(N^{2/3})$ in three dimensions [20]. Given that there are $O(N)$ floating point operations (flops) per PCG iteration and assuming that the total number of iterations required is proportional to the square root of the condition number of the preconditioned matrix [3], using PCG with diagonal scaling (which does not alter the asymptotic order of the condition number) has asymptotic complexity $O(N^{4/3})$. Similarly, any preconditioner that reduces the asymptotic order of the condition number to $O(h^{-1})$ will have complexity $O(N^{7/6})$. For a large problem with, for example, $N = 10^6$ unknowns, $N^{4/3} = 10N^{7/6}$. That is, the more complex preconditioner is even at first glance only ten times better than diagonal scaling. In addition, the constants that are hidden in the order notation in these work estimates can have a significant influence. Thus the effect on these constants of aspects of implementation and architecture, work per iteration, and internal eigenvalue distribution is considerably more pronounced for large three-dimensional problems than for those in lower dimensions.

In practice it is often hard to quantify how difficult it is to implement a particular preconditioner. As stated above, diagonal scaling does not alter the asymptotic order of the condition number. Its benefits lie in the fact that for any particular case it will reduce the condition number by a constant factor [7]. One example of a method that reduces the asymptotic order of the finite element matrix condition number to $O(h^{-1})$ is the popular modified incomplete Cholesky conjugate gradient (MICCG(0)) method of Gustafsson [11]. However, a major drawback of using such a preconditioning for three-dimensional irregular problems is that often the coefficient matrix is not an M-matrix and so existence and stability of the factorisation is not guaranteed [15]. Other sparse factorisation methods such as the Dupont, Kendall, and Rachford (DKR) method [6], the strongly implicit procedure (SIP) [19] and the consistent sparse factorization (CSF) method [4] cannot be used in three dimensions. Many of the attractive vector and parallel capabilities of polynomial preconditioners [1] are lost when using irregular grids. Hierarchical preconditioning gives an asymptotic condition number of $O(\ln N)^2$ for triangular elements in two dimensions [27] and Ong obtains the analogous but less attractive result of $O(N \ln N)$ using tetrahedral elements in three dimensions [16]. She does, however, go on to observe that using such preconditioners in three dimensions is impractical on irregular grids. These observations reinforce the point that standard comparisons of preconditioners that involve only condition number estimates are not always appropriate for large irregular three-dimensional problems.

4. Element-based preconditioners. One common feature of many preconditioners is that they take little or no account of the very specific structure of finite

element matrices. It is natural in a finite element context to look for some sort of preconditoning that can be applied to each individual element. The simplest method that can be considered in this way is diagonal scaling, that is, set

$$B = L^T[\text{diag}A_e]L = \text{diag}A \tag{4.1}$$

(see [26]). Using this preconditioning leaves the asymptotic matrix condition number as $O(h^{-2})$, but has the advantage that it will reduce the condition number by a constant factor and is extremely simple to implement. We now show that it is one of a large class of element-based preconditioners that does not affect the asymptotic order of the condition number.

Suppose we have a preconditioner B of the form (2.2), where all of the eigenvalues of each element matrix B_e are of the same order as the discretisation parameter h tends to zero; that is,

$$c_1 h^p \leq \lambda_{\min}(B_e) \leq c_2 h^p \quad \text{and} \quad c_3 h^p \leq \lambda_{\max}(B_e) \leq c_4 h^p, \tag{4.2}$$

where $p \in \Re$. Throughout the following $c_i, i = 1, 2, \ldots$ represents nonzero positive constants. Result (2.3) can be used to deduce an inequality analogous to (4.2) for the global case, that is,

$$c_5 h^p \leq \lambda(B) \leq c_6 h^p. \tag{4.3}$$

The quantity of interest from our point of view is $\kappa(B^{-1}A)$. Because $B^{-1}A$ is symmetric with respect to the A-inner product, the ratio $\lambda_{\max}(B^{-1}A)/\lambda_{\min}(B^{-1}A)$ gives the A-condition number rather than the usual $(I-)$ condition number [2]. To bound the I-condition number we use the general definition

$$\kappa(B^{-1}A) = \| B^{-1}A \| \| A^{-1}B \| \tag{4.4}$$

to give

$$\kappa(B^{-1}A) \leq \| B^{-1} \| \| A \| \| A^{-1} \| \| B \| = \kappa(A)\kappa(B). \tag{4.5}$$

From (4.3), $\kappa(B)$ must be independent of h so

$$\kappa(B^{-1}A) \leq c_7 \kappa(A). \tag{4.6}$$

Furthermore,

$$\lambda_{\max}(B^{-1}A) = \max_{\mathbf{x}=\mathbf{0}} \frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T B\mathbf{x}} . \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{4.7}$$

and so

$$\lambda_{\max}(B^{-1}A) \geq \frac{\mathbf{v}_{\max}^T\mathbf{v}_{\max}}{\mathbf{v}_{\max}^T B\mathbf{v}_{\max}} \lambda_{\max}(A) = Q_1 \lambda_{\max}(A), \tag{4.8}$$

where $\mathbf{v}_{\max}$ is the eigenvector corresponding to $\lambda_{\max}(A)$. Similarly,

$$\lambda_{\min}(B^{-1}A) \leq \frac{\mathbf{v}_{\min}^T\mathbf{v}_{\min}}{\mathbf{v}_{\min}^T B\mathbf{v}_{\min}} \lambda_{\min}(A) = Q_2 \lambda_{\min}(A), \tag{4.9}$$

where $\mathbf{v}_{\min}$ is the eigenvector corresponding to $\lambda_{\min}(A)$. For any natural matrix norm,

$$| \lambda(M) | \leq \| M \| \tag{4.10}$$

where M is an arbitrary square matrix (e.g., [23, p. 10]) so, from (4.8),

$$Q_1 \lambda_{\max}(A) \leq \| B^{-1}A \| . \tag{4.11}$$

Now, from (4.9),

$$\lambda_{\min}(B^{-1}A) \leq Q_2 \lambda_{\min}(A), \tag{4.12}$$

so

$$\frac{1}{\lambda_{\max}(A^{-1}B)} \leq Q_2 \lambda_{\min}(A). \tag{4.13}$$

Using result (4.10) and inverting both sides gives

$$\| A^{-1}B \| \geq \frac{1}{Q_2 \lambda_{\min}(A)}, \tag{4.14}$$

hence

$$\| B^{-1}A \| \| A^{-1}B \| \geq \frac{\| B^{-1}A \|}{Q_2 \lambda_{\min}(A)}. \tag{4.15}$$

Finally, from (4.11),

$$\| B^{-1}A \| \| A^{-1}B \| \geq \frac{Q_1 \lambda_{\max}(A)}{Q_2 \lambda_{\min}(A)}. \tag{4.16}$$

Again using assumption (4.2) (and the corresponding global result (4.3)) the quotients Q_1 and Q_2 involving B in (4.8) and (4.9) are of the same order as $h \to 0$ and so

$$\kappa(B^{-1}A) \geq c_8 \kappa(A). \tag{4.17}$$

Inequalities (4.6) and (4.17) give rise to the result that if the element matrices B_e satisfy (4.2), then

$$\kappa(B^{-1}A) = c_9 \kappa(A). \tag{4.18}$$

Thus the order of the condition number of the coefficient matrix can never be improved by applying any element-based preconditioner of this form. Note that result (4.18) applies to diagonal scaling and is consistent with what we have already observed, namely, that in such a case the asymptotic order of the condition number of the global finite element stiffness matrix is unchanged by preconditioning.

It is appropriate to mention here the element-by-element (EBE) method of Hughes, Levit, and Winget [12]. Although this method does not fit exactly into the above category, analysis of the condition number indicates that EBE preconditioning is also spectrally equivalent to diagonal scaling [14], [26] and so will not affect the asymptotic condition number estimate. Numerical experiments in the above papers show that in practice the constants in the estimates are significantly better for EBE preconditioning than those for diagonal scaling on regular grids. The performance of the two methods on some irregular grids is compared in §5.

TABLE 1
ARDENT TITAN CPU times: Problem 1.

N		k	V=OFF		V=ON	
			Setup	Solve	Setup	Solve
125	DSCG	81	–	1.39	–	0.58
	ELCG	38	0.26	1.10	0.01	0.98
	SFCG	51	0.07	0.21	0.08	0.21
1000	DSCG	87	–	21.49	–	5.75
	ELCG	38	0.21	14.93	0.14	10.52
	SFCG	59	4.92	3.40	5.18	2.68
3375	DSCG	337	–	324.07	–	83.40
	ELCG	134	0.68	188.40	0.54	129.43
	SFCG	221	58.93	51.11	60.80	42.05
8000	DSCG	379	–	885.25	–	230.07
	ELCG	147	1.73	500.46	1.32	343.15
	SFCG	234	343.43	137.96	351.28	115.69
		Random initial guess				
8000	DSCG	373	–	873.33	–	227.93
	ELCG	150	1.72	510.11	1.30	351.55
	SFCG	239	344.87	141.8	352.36	118.44

5. Numerical experiments. In the past it has been common practice to use test problems on regular finite element grids as a yardstick for comparing the performance of various preconditioners [1], [10], [16]. However, as stated above, the performance of some preconditioning methods can deteriorate rapidly when the grids are distorted, and so this approach may not always give a fair comparison. Here we compare methods using two types of unstructured grid: the set of grids in Problem 1 have been randomly constructed to be extremely irregular, while the grid in Problem 2 was constructed to model a particular irregular physical geometry in a practical situation.

Problem 1. We look for a solution to Laplace's equation as the steady state of the related parabolic partial differential equation

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla^2 u = 0 \quad &\text{in } \Omega, \\ \frac{\partial u}{\partial n} = 0 \quad &\text{on } \delta\Omega \end{aligned} \tag{5.1}$$

for domain Ω with boundary $\delta\Omega$ and unit normal n. Using the Galerkin finite element method with fully implicit backward difference timestepping leads to a matrix system

$$\left[\frac{M}{\delta t} + K\right] \mathbf{u}^{n+1} = \frac{M}{\delta t}\mathbf{u}^{n}, \tag{5.2}$$

where M and K are the finite element mass and stiffness matrices and $\mathbf{u}_n$ and $\mathbf{u}_{n+1}$ are vectors of the solution values at timesteps n and $n+1$, respectively. This system has a symmetric positive definite coefficient matrix and was solved using the PCG method with various preconditioners, terminating in each case when the Euclidean norm of the residual vector, $\mathbf{r}_n = \mathbf{b} - A\mathbf{u}_n$, was less than 10^{-4}. All calculations were done in double precision FORTRAN on an ARDENT TITAN machine with unit roundoff of 10^{-15}.

The results in Table 1 come from irregular three-dimensional grids of linear tetrahedra (with four unknowns per element). The coordinates of the global unknowns were generated using a random number generator (with appropriate scaling) and the

TABLE 2
Irregular grid specifications: Problem 1.

	Grid 1	Grid 2	Grid 3	Grid 4
N	125	1000	3375	8000
E	671	6354	22181	52992
NZ	1775	15820	54655	130172
$NZ_{\min}$	7	6	7	5
$NZ_{\max}$	24	33	37	204
$P_{\min}$	5	5	8	4
$P_{\max}$	42	56	57	388
$x_{\min}$	35.224	64.696	3.378	3.302
$x_{\max}$	66.377	71.379	66.527	83.472
$y_{\min}$	18.791	63.285	0.835	1.905
$y_{\max}$	66.513	71.380	66.527	66.528
$z_{\min}$	24.992	64.384	0.271	10.901
$z_{\max}$	66.444	71.378	66.528	66.528

convex hull of these points was then tetrahedralised using a FORTRAN grid generation code [21]. Some specifications of these grids are given in Table 2, including the maximum and minimum number of elements round each node (P) and the maximum and minimum number of nonzeros in any one row of the assembled matrix (NZ). The extreme values of the x, y, and z coordinates are also listed. The fact that some nodes in the grid are surrounded by a very large number of elements suggests that the angles in these elements will be extremely small, indicating that the tetrahedra are badly distorted. Tetrahedral elements are in reality not particularly practical, especially from the point of view of element-based preconditioners, as the number of elements is much larger than the number of unknowns. They are used here because it is relatively easy to generate very irregular grids with such elements.

We compare the performance of three preconditioning methods:

- DSCG, diagonal scaling;
- ELCG, element-by-element preconditioning [12];
- SFCG, sparse factorisation based on MICCG(0) [11] but taking the absolute value of the pivot at each stage to ensure stability (as the coefficient matrix is not an M-matrix).

All CPU times quoted are for the iterative solver only and do not include matrix setup times. Results are given for each program in scalar (V OFF) and vector (V ON) mode. As the time scales involved in this type of problem can be large, it was appropriate to use timesteps of variable length in the matrix system (5.2). This, however, introduced an unfair imbalance as the sparse factorisation had to be recalculated at every timestep. To remove this bias, a large timestep was chosen ($\delta t = 10^5$) so that the steady state in the runs of Table 1 was achieved in one time step. The *setup* times (in CPU seconds) do not refer to the time taken to construct the finite element equations; this has been omitted in all cases. Instead they give an indication of the amount of work that must be done once (or once per timestep for a variable δt) before the actual iterative solution can proceed. The setup steps involved for each method are:

- DSCG, setup time negligible;
- ELCG, crout factorisation of element matrices;
- SFCG, sparse factorisation of assembled coefficient matrix.

The *solve* times are the number of CPU seconds taken for the k iterations required to achieve convergence.

For the first four cases shown, the initial condition chosen was $\mathbf{u}_0 = \frac{1}{10}(\mathbf{x}+\mathbf{y}+\mathbf{z})$,

TABLE 3
ARDENT TITAN CPU times: Problem 2.

N		k	V=OFF		V=ON	
			Setup	Solve	Setup	Solve
9422	DSCG	88	–	110.48	–	90.36
	ELCG	102	4.89	523.97	3.59	233.68
	SFCG	15	1737.0	23.37	1768.42	20.76

where $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are the coordinates of the node points. To see if this choice of a "smooth" starting vector affected the speed of convergence, the experiments were repeated on the largest grid with a random initial vector. However, the choice of starting vector did not seem to have a significant effect on the overall results.

The results of these numerical experiments give a rough idea of the comparative performance of the three methods. The idea of these tests was to use each method in a straightforward way. In particular, the factorisation time for SFCG may well be improved by using a more sophisticated implementation. It is still, however, true that the overall solution time for SFCG will be dominated by the factorisation step itself. This re-emphasises the point that for large and very irregular three-dimensional problems, implementation plays a very important role.

An equivalent set to the results in Table 1 from a series of regular grid problems would show obvious relationships between the iteration counts and total number of unknowns N (usually expressed in terms of the discretisation parameter h). Such specific conclusions cannot be drawn in the irregular case. Nevertheless, some important observations can be made. In terms of iteration counts, ELCG is the most efficient. This supports the remarks of §4 concerning the relative performance of the EBE method and diagonal scaling, namely, that although the h-dependence in the asymptotic estimates of their convergence rates is the same, that of the latter will contain a smaller constant [26]. Notice, however, the effect of vectorisation, which gives a dramatic improvement in the performance of DSCG. This means that despite taking more iterations, it is faster than ELCG in its vectorised form. In fact, for the largest of these particular problems, it is vectorised diagonal scaling which is the best technique to use in terms of total CPU time.

Problem 2. We look for a solution to Poisson's equation

$$\begin{aligned} \nabla^2 u &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \delta\Omega \end{aligned} \tag{5.3}$$

for domain Ω with boundary $\delta\Omega$. Using the Galerkin finite element method leads to a system with a symmetric positive definite coefficient matrix. This was solved using the PCG method with various preconditioners, this time terminating in each case when the Euclidean norm of the residual vector, $\mathbf{r}_n = \mathbf{b} - A\mathbf{u}_n$, was less than 10^{-6}. All calculations were again done in double precision FORTRAN on an ARDENT TITAN machine.

The results in Table 3 come from the irregular three-dimensional grid of serendipity bricks (with twenty unknowns per element) shown in Fig. 1. This grid was generated for a commercial simulation by the British company Nuclear Electric plc and some of its specifications are given in Table 4.

We compare the performance of the same three preconditioning methods described above, again quoting CPU times for the iterative solver only. The initial condition chosen was $\mathbf{u}_0 = 0$.

TABLE 4
Irregular grid specifications: Problem 2.

N	9422
E	2868
NZ	459516
$NZ_{\min}$	16
$NZ_{\max}$	119
$P_{\min}$	3
$P_{\max}$	12

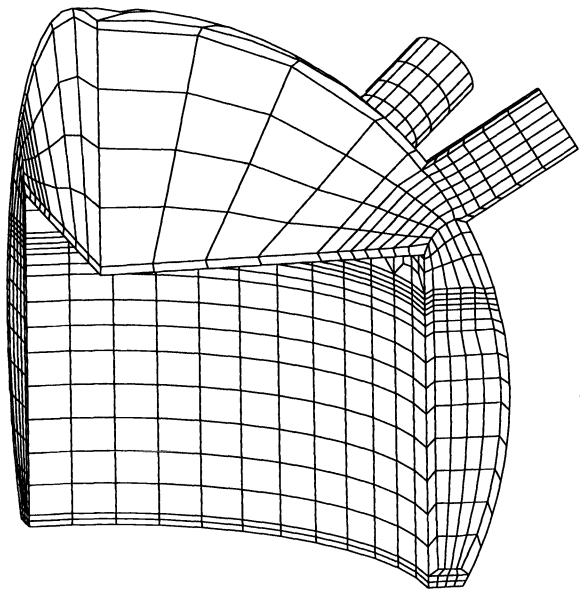


FIG. 1.

The major difference between these results and those from Problem 1 seems to be a deterioration in the performance of ELCG. The reason for this may be related to the fact that the grid contains elements of extreme aspect ratios: the relation between element aspect ratio and the EBE method was studied in [14]. For DSCG and SFCG, however, the same behaviour pattern is seen; that is, although SFCG performs well, the time taken to perform the actual factorisation is prohibitive. Vectorisation, while improving the performance of DSCG, does not have the same marked effect as in Problem 1. This is due to a difference in implementation. In this problem, the value of u is fixed at certain nodes by boundary conditions. This means that the number of unknowns in the matrix system is slightly less than the total number of nodes on the original grid, and so not all the elements have the same number of unknowns. As a result, a different storage strategy has been used for the coefficient matrix which makes some parts of the code less vectorisable.

The above experiments show that it is difficult to make generalisations when dealing with large irregular grids. In practice, the final choice of solution technique in any particular case will depend both on the type of problem to be solved and the hardware to be used in the solution process. It is however important to appreciate that using very large irregular three-dimensional grids may involve a different philosophical approach.

Appendix A. The finite element stiffness matrix of any linear triangle can be expressed entirely in terms of the coordinates of the three node points. To find the eigenvalues for an arbitrary linear triangle, it is sufficient to consider a triangle that has been translated to fix one node at the origin and rotated to fix one side along the y-axis, i.e., a triangle with coordinates $(x_1, y_1),(0, y_2),(0, 0)$.

In this case the stiffness matrix is

$$K_e = \frac{1}{4S_e}\begin{bmatrix} a & -b & b-a \\ -b & c & b-c \\ b-a & b-c & a-2b+c \end{bmatrix}, \tag{A.1}$$

where S_e is the element area, $a = y_2^2$, $b = y_1 y_2$, and $c = x_1^2 + y_1^2$. The eigenvalues of $D_e^{-1}K_e$ (where D_e is the diagonal of K_e) can be found algebraically and the results can be simplified by expressing a, b, and c in terms of the angles θ_i with opposite sides of length s_i ($i = 1, 2$, or 3). This gives precisely the eigenvalues in (2.6), namely,

$$\begin{cases} 0, \\ \dfrac{3}{2} \pm \dfrac{1}{2}\left\{9 - 8\left[\dfrac{S_e \sin 2\theta_i}{s_i^2} + \sin^2\theta_i\right]\right\}^{1/2}. \end{cases} \tag{A.2}$$

It is clear that the maximum and minimum values of (A.2) both occur at the minimum value of

$$\frac{S_e \sin 2\theta_i}{s_i^2} + \sin^2\theta_i. \tag{A.3}$$

As $S_e = \frac{1}{2}s_j s_k \sin\theta_i$ (where $i \neq j \neq k$), using the cosine rule

$$s_j s_k \cos\theta_i = \frac{1}{2}(s_j^2 + s_k^2 - s_i^2) \tag{A.4}$$

reduces (A.3) to

$$\left[\frac{\sin\theta_i}{s_i}\right]^2 (s_i^2 + s_j^2 + s_k^2). \tag{A.5}$$

Note that (from the sine rule) this is independent of the index.

By expressing the area S_e in terms of the semiperimeter $\frac{1}{2}(s_i + s_j + s_k)$, we can substitute for $\sin\theta$ in (A.5) to give

$$\frac{1}{8}(s_i + s_j + s_k)(s_j + s_k - s_i)(s_i + s_k - s_j)(s_i + s_j - s_k)\frac{(s_i^2 + s_j^2 + s_k^2)}{s_i^2 s_j^2 s_k^2}. \tag{A.6}$$

As this formula is symmetric, we arbitrarily choose s_k to be the longest side and write

$$\frac{s_i}{s_k} = \lambda, \qquad \frac{s_j}{s_k} = \mu, \tag{A.7}$$

so (A.6) becomes

$$\frac{1}{8}(\lambda + \mu + 1)(\mu + 1 - \lambda)(\lambda + 1 - \mu)(\lambda + \mu - 1)\frac{(\lambda^2 + \mu^2 + 1)}{\lambda^2\mu^2}. \tag{A.8}$$

From the obvious inequalities

$$\lambda + \mu \geq 1, \quad \mu + 1 \geq \lambda, \quad \lambda + 1 \geq \mu, \tag{A.9}$$

it is clear that (A.8) will have a minimum of zero precisely when one of these holds as an equality; that is, when the triangle involved is a straight line. Thus (A.3) will tend to a minimum as the angle θ_i tends towards π.

Acknowledgments. The authors would like to thank Dr. S. Ashby of Lawrence Livermore National Laboratory for useful comments concerning the proof in §4.

REFERENCES

[1] S. F. ASHBY, *Polynomial Preconditioning for Conjugate Gradient Methods*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1987.

[2] 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.

[3] O. AXELSSON AND V. A. BARKER, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Academic Press, New York, 1984.

[4] A. J. BAKER, P. NORONHA, AND E. WACHSPRESS, *Consistent Sparse Factorization of Elliptic Difference Equations*, Appl. Math. Lett., to appear.

[5] P. CONCUS, G. H. GOLUB, AND D. P. O'LEARY, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, Sparse Matrix Computations, J. Bunch and D.J. Rose, eds., Academic Press, New York, 1976.

[6] T. DUPONT, R. P. KENDALL, AND H. H. RACHFORD, JR., *An approximate factorization procedure for solving self-adjoint elliptic difference equations*, SIAM J. Numer. Anal., 5 (1968), pp. 559–573.

[7] G. E. FORSYTHE AND E. G. STRAUSS, *On best conditioned matrices*, Proc. Amer. Math. Soc., 6 (1955), pp. 340–345.

[8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., John Hopkins University Press, London, 1989.

[9] A. GREENBAUM, *Comparison of splittings used with the conjugate gradient algorithm*, Numer. Math., 33 (1979), pp. 181–193.

[10] A. GREENBAUM, C. LI, AND H. Z. CHAO, *Comparison of linear system solvers applied to diffusion-type finite element equations*, Numer. Math., 56 (1989), pp. 529–546.

[11] I. GUSTAFSSON, *A class of first order factorization methods*, BIT, 18 (1978), pp. 142–156.

[12] T. J. R. HUGHES, I. LEVIT, AND J. WINGET, *An element-by-element solution algorithm for problems of structural and solid mechanics*, Comput. Meth. Appl. Mech. Engrg., 36 (1983), pp. 241-254.

[13] C. J. HUNT, *Finite Element Meshes using Delaunay Triangles*, M.Sc. thesis, University of Reading, U.K., 1988.

[14] H.-C. LEE AND A. J. WATHEN, *On element-by-element preconditioning for general elliptic problems*, Comput. Meth. Appl. Mech. Engrg., 92 (1991), pp. 215–229.

[15] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[16] M. E. G. ONG, *Hierarchical Basis Preconditioners for Second Order Ellipitc Problems in Three Dimensions*, Ph.D. Thesis, University of Washington, Seattle, 1989.

[17] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[18] A. RAMAGE, *Preconditioned Conjugate Gradient Methods for Galerkin Finite Element Equations*, Ph.D. thesis, University of Bristol, U.K., 1990.

[19] H. L STONE, *Iterative solution of implicit approximations of multidimensional partial differential equations*, SIAM J. Numer. Anal., 5 (1968), pp. 530–558.

[20] G. STRANG AND G. J. FIX, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

[21] P. K. SWEBY, *FORTRAN tetrahedralisation code*, private communication, 1989.

[22] A. VAN DER SLUIS AND H. A. VAN DER VORST, *The rate of convergence of conjugate gradients*, Numer. Math., 48 (1986), pp. 543–560.

[23] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[24] A. J. WATHEN, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., 7 (1987), pp. 449–457.

[25] ———, *Spectral bounds and preconditioning methods using element-by-element analysis for Galerkin finite element equations*, The Mathematics of Finite Elements and Applications VI (MAFELAP 1987), J. R. Whiteman, ed., Academic Press, New York, 1988, pp. 157–168.

[26] ———, *An analysis of some element-by-element techniques*, Comput. Meth. Appl. Mech. Engrg., 74 (1989), pp. 271–287.

[27] H. YSERENTANT, *On the multi-level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 922–937, July 1994

014

THE REVERSE BORDERING METHOD*

C. BREZINSKI†, M. MORANDI CECCHI‡, AND M. REDIVO-ZAGLIA§

Abstract. The bordering method allows recursive computation of the solution of a system of linear equations by adding one new row and one new column at each step of the procedure. When some of the intermediate systems are nearly singular, it is possible, by the block bordering method, to add several new rows and columns simultaneously. However, in that case, the solutions of some of the intermediate systems are not computed. The reverse bordering method allows computation of the solutions of these systems afterwards. Such a procedure has many applications in numerical analysis, that include orthogonal polynomials, Padé approximation, and the progressive forms of extrapolation processes.

Key words. linear equations, bordering methods, extrapolation, Padé approximants, orthogonal polynomials

AMS subject classifications. 65F05, 65B05

1. Introduction.

The bordering method is a recursive method for computing the solution of a system of linear equations. It consists, at each step, of adding one new row and one new column to the previous matrix and using the previous solution to compute the new one. This method can only be used if some quantity is different from zero at each step. If, at some steps, this quantity is zero (or nearly zero), then it is possible to add several new rows and columns to the matrix simultaneously. However, if this situation occurs, the solutions of the intermediate systems that have been skipped are not computed. This is a drawback of the method since, in some applications, the solutions of all the intermediate systems must be known (if nonsingular). In this paper we propose the reverse bordering method for avoiding this case. The procedure is, after jumping over the near-singular intermediate systems and computing the solution of the first nonnear-singular system, to go back by decreasing the dimension of the matrix (that is, by deleting the last row and the last column) and using the solution of the previous larger system for computing the solutions of the smaller systems that have been skipped. Such a procedure has applications in the recursive computation of orthogonal polynomials, in Padé approximation, and in the implementation of the progressive forms of extrapolation algorithms.

2. Bordering method.

When we must solve a system of linear equations that is obtained by adding one new equation and one new unknown to a given system or, in other words, when the matrix of the system has been bordered by a new row

* Received by the editors March 17, 1992; accepted for publication (in revised form) January 26, 1993.

† Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA-M3, Université des Sciences et Technologies de Lille, F-59655 Villeneuve d'Ascq Cedex, France (brezinsk@omega.univ-lille1.fr).

‡ Dipartimento di Matematica Pura ed Applicata, Università degli Studi di Padova, via Belzoni 7, I-35131 Padova, Italy (mcecchi@pdmat1.unipd.it).The work of this author was partially supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (Italy) (National Project Analisi Numerica e Matematica Computazionale) and by Consiglio Nazionale delle Ricerche (Italy), Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo, Sottoprogetto Calcolo Scientifico per Grandi Sistemi.

§ Dipartimento di Elettronica e Informatica, Università degli Studi di Padova, via Gradenigo 6/a, I-35131 Padova, Italy (elen@elett1.dei.unipd.it).The work of this author was partially supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (Italy) (National Project Analisi Numerica e Matematica Computazionale) and by Consiglio Nazionale delle Ricerche (Italy), Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo, Sottoprogetto Calcolo Scientifico per Grandi Sistemi.

and a new column, the bordering method can be used to save computational time. This method is well known in numerical analysis and permits us to solve a system recursively by using the solution of the previous system (see, Faddeeva [7]). Let us first explain this method.

Let A_k be a regular square matrix of dimension k and d_k a vector of dimension k. Let z_k be the solution of the system

$$A_k z_k = d_k.$$

Now let u_k be a column vector of dimension k, v_k a row vector of dimension k, and a_k a scalar. We consider the bordered matrix A_{k+1} of dimension $k+1$ given by

$$A_{k+1} = \begin{pmatrix} A_k & u_k \\ v_k & a_k \end{pmatrix}.$$

We have

$$A_{k+1}^{-1} = \begin{pmatrix} A_k^{-1} + A_k^{-1} u_k v_k A_k^{-1}/\beta_k & -A_k^{-1} u_k/\beta_k \\ -v_k A_k^{-1}/\beta_k & 1/\beta_k \end{pmatrix}$$

with $\beta_k = a_k - v_k A_k^{-1} u_k$.

Let f_k be a scalar and z_{k+1} be the solution of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{pmatrix} d_k \\ f_k \end{pmatrix}.$$

Then we have

$$z_{k+1} = \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \frac{f_k - v_k z_k}{\beta_k} \begin{pmatrix} -A_k^{-1} u_k \\ 1 \end{pmatrix}.$$

This formula gives the solution of the bordered system in terms of the solution of the previous system.

To avoid computation and storage of A_k^{-1}, we can set $q_k = -A_k^{-1} u_k$ and compute it recursively by the same bordering method. In such a way, we obtain the following variant of the bordering method that needs the storage of A_k instead of that of A_k^{-1} for the original procedure.

Let $q_k^{(i)}$ be the solution of the system

$$A_i q_k^{(i)} = -u_k^{(i)},$$

where $u_k^{(i)}$ is the vector formed by the first i components of u_k. Thus $u_i^{(i)} = u_i$ and $q_i^{(i)} = q_i$ for all i. A_i is the matrix of dimension i formed by the first i rows and columns of A_k.

We have, since A_1 is a number,

$$q_k^{(1)} = -\frac{u_k^{(1)}}{A_1},$$

$$q_k^{(i+1)} = \begin{pmatrix} q_k^{(i)} \\ 0 \end{pmatrix} - \frac{u_{k,i+1} + v_i q_k^{(i)}}{a_i + v_i q_i^{(i)}} \begin{pmatrix} q_i^{(i)} \\ 1 \end{pmatrix}, \qquad i = 1, \ldots, k-1,$$

where $u_{k,i+1}$ is the $(i+1)$th component of u_k.

Then

$$q_k^{(k)} = q_k = -A_k^{-1} u_k,$$

thus allowing us to use the previous formula for computing z_{k+1} without knowledge of A_k^{-1}; [3] and [4] contain the subroutine BORDER performing this variant of the bordering method.

3. Block bordering method. The bordering method can be applied only if $\beta_k \neq 0$ for all k. When this is not the case, we can use a block bordering procedure (see Brezinski, Redivo-Zaglia, and Sadok [5]).

We now assume the following dimensions for the matrices involved in the process

$$\begin{aligned} A_k &\quad n_k \times n_k, \\ u_k &\quad n_k \times p_k, \\ v_k &\quad p_k \times n_k, \\ a_k &\quad p_k \times p_k, \end{aligned}$$

and finally

$$A_{k+1} \quad n_{k+1} \times n_{k+1}$$

with $n_{k+1} = n_k + p_k$.

We set

$$\beta_k = a_k - v_k A_k^{-1} u_k$$

and we have

$$A_{k+1}^{-1} = \begin{pmatrix} A_k^{-1} + A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1} & -A_k^{-1} u_k \beta_k^{-1} \\ -\beta_k^{-1} v_k A_k^{-1} & \beta_k^{-1} \end{pmatrix}.$$

f_k is now a vector with p_k components and we obtain

$$z_{k+1} = \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \begin{pmatrix} -A_k^{-1} u_k \\ I_k \end{pmatrix} \beta_k^{-1} \, (f_k - v_k z_k),$$

where I_k is the identity matrix of dimension p_k.

The subroutine BLBORD given in [4] performs this block bordering method.

Remark. We note that the subroutine BLBORD only works if $a_{11} = 1$, which can always be made true. It is also possible to add the instruction A(1,1)=1.0D0/A(1,1) after the instruction Z(1)=D(1)/A(1,1).

Again it is possible to avoid computation and storage of A_k^{-1} by setting $q_k = -A_k^{-1} u_k$ (whose dimension is $n_k \times p_k$) and computing it recursively by the bordering method in the following way.

Let $u_k^{(i)}$ be the $n_i \times p_k$ matrix formed by the first n_i rows of u_k for $i \leq k$, $n_i \leq n_k$. We have $u_i^{(i)} = u_i$. Let $q_k^{(i)}$ be the $n_i \times p_k$ matrix satisfying $A_i q_k^{(i)} = -u_k^{(i)}$ for $i \leq k$. We have $q_i^{(i)} = q_i$.

We set

$$q_k^{(1)} = -A_1^{-1} u_k^{(1)}$$

and then we have

$$q_k^{(i+1)} = \begin{pmatrix} q_k^{(i)} \\ 0 \end{pmatrix} - \begin{pmatrix} q_i^{(i)} \\ I_i \end{pmatrix} \beta_i^{-1} \left(u_{k,i+1} + v_i q_k^{(i)} \right), \qquad i = 1, \ldots, k-1$$

with $\beta_i = a_i + v_i q_i^{(i)}$ and $u_{k,i+1}$ the matrix formed by the rows $n_i + 1, \ldots, n_i + p_i$ of u_k.

Instead of using the block bordering method when β_k is zero for some k, it is possible to use a pivoting strategy. If, for some k, $\beta_k = 0$, then the last row of the matrix can be interchanged with the next one and so on until some $\beta_k \neq 0$ has been obtained. Such a procedure is not adapted when the solutions of the intermediate systems must be computed. It can be used if only the last solution is needed and if the solutions of the intermediate systems are not required.

Obviously, the block bordering method can also be used even if the matrix β_k is nonsingular and thus, at each step, an arbitrary number of new rows and columns can be added. In particular, when some of the intermediate systems are almost singular, such a strategy allows us to jump over them and thus improve the numerical stability and precision of the solutions of the subsequent systems. However, in such a case the solutions of the systems that have been skipped have not been computed. The reverse bordering method that we now present allows us to come back afterwards to these systems by deleting rows and columns one by one and obtain their solutions from the solution of the larger system.

4. Reverse bordering method. Let us now look at the possibility of finding A_k^{-1} from A_{k+1}^{-1}.

We write the inverse matrix A_{k+1}^{-1} of dimension n_{k+1} under the form

$$A_{k+1}^{-1} = \begin{array}{c} \begin{matrix} n'_k & p'_k \end{matrix} \\ \begin{pmatrix} A'_k & u'_k \\ v'_k & a'_k \end{pmatrix} \end{array} \begin{matrix} n'_k \\ p'_k \end{matrix}.$$

The matrix A_{k+1} will also be partitioned by blocks with the same corresponding dimensions. Thus A_k will be the square matrix of dimension $n'_k = n_{k+1} - p'_k$ obtained by suppressing the last p'_k rows and columns of A_{k+1}.

From the block bordering method we know that

$$\begin{aligned} A'_k &= A_k^{-1} + A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1} \\ u'_k &= -A_k^{-1} u_k \beta_k^{-1}, \qquad v'_k = -\beta_k^{-1} v_k A_k^{-1}, \\ a'_k &= \beta_k^{-1}, \qquad \beta_k = a_k - v_k A_k^{-1} u_k. \end{aligned}$$

Because

$$a_k'^{-1} = \beta_k$$

then

$$u'_k a_k'^{-1} = -A_k^{-1} u_k,$$

and thus

$$u'_k a_k'^{-1} v'_k = A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1}.$$

Thus using this in the expression of A'_k gives us

$$A_k^{-1} = A'_k - u'_k\, a_k'^{-1}\, v'_k. \tag{1}$$

This formula, which corresponds to the Schur complement, was already given by Duncan in 1944 [6]. The following relations also hold.

$$\det A_k^{-1} = \det A_{k+1}^{-1} / \det a'_k,$$
$$\det A_{k+1} = \det A_k \cdot \det \beta_k.$$

Moreover, from the Sherman–Morrison formula (see [8] for review), we have

$$A'_k = \left(A_k - u_k\, a_k^{-1}\, v_k\right)^{-1},$$

and

$$A_k = \left(A'_k - u'_k\, a_k'^{-1}\, v'_k\right)^{-1} = A_k'^{-1} + A_k'^{-1}\, u'_k\, \beta_k'^{-1}\, v'_k\, A_k'^{-1},$$

with

$$\beta'_k = a'_k - v'_k\, A_k'^{-1}\, u'_k.$$

This is another proof of (1).

From these formulæ, we obtain

$$A_k = A_k'^{-1} + u_k\, a_k^{-1}\, v_k = \left(A'_k - u'_k\, a_k'^{-1}\, v'_k\right)^{-1},$$
$$A'_k = A_k^{-1} + u'_k\, a_k'^{-1}\, v'_k = \left(A_k - u_k\, a_k^{-1}\, v_k\right)^{-1}.$$

Now we want to compute the solution z_k of the previous system

$$A_k z_k = d_k$$

starting from the solution z_{k+1} of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{pmatrix} d_k \\ f_k \end{pmatrix} \begin{matrix} n'_k \\ p'_k \end{matrix}.$$

As previously stated

$$z_{k+1} = \begin{pmatrix} z'_k \\ c_k \end{pmatrix} = \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \begin{pmatrix} -A_k^{-1} u_k \\ I_k \end{pmatrix} \beta_k^{-1}\, (f_k - v_k z_k)$$
$$= \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \begin{pmatrix} u'_k \\ a'_k \end{pmatrix} (f_k - v_k z_k).$$

Thus

$$c_k = a'_k f_k - a'_k v_k z_k,$$

that is,

$$a_k'^{-1} c_k = f_k - v_k z_k \quad \text{or} \quad v_k z_k = f_k - a_k'^{-1} c_k$$

and

$$z'_k = z_k + u'_k\, (f_k - v_k z_k) = z_k + u'_k\, a_k'^{-1}\, c_k.$$

Finally, it holds that

$$z_k = z'_k - u'_k\, a'^{-1}_k\, c_k.$$

Another way of finding z_k is as follows. From (1), we have

$$A_k^{-1} d_k = A'_k d_k - u'_k\, a'^{-1}_k\, v'_k\, d_k.$$

But

$$\begin{pmatrix} z'_k \\ c_k \end{pmatrix} = \begin{pmatrix} A'_k & u'_k \\ v'_k & a'_k \end{pmatrix} \begin{pmatrix} d_k \\ f_k \end{pmatrix} = \begin{pmatrix} A'_k d_k + u'_k f_k \\ v'_k d_k + a'_k f_k \end{pmatrix}.$$

Thus

$$A'_k d_k = z'_k - u'_k f_k \quad \text{and} \quad v'_k d_k = c_k - a'_k f_k$$

and we have

$$\begin{aligned} z_k = A_k^{-1} d_k &= z'_k - u'_k f_k - u'_k a'^{-1}_k\, (c_k - a'_k f_k) \\ &= z'_k - u'_k f_k - u'_k\, a'^{-1}_k\, c_k + u'_k\, a'^{-1}_k\, a'_k\, f_k \\ &= z'_k - u'_k a'^{-1}_k c_k. \end{aligned}$$

5. Variants and particular cases. Instead of bordering the matrix A_k as we did, we can also add the new rows and columns on the top and on the left according to the scheme

$$A_{k+1} = \begin{pmatrix} a_k & v_k \\ u_k & A_k \end{pmatrix}.$$

Thus the inverse matrix becomes

$$A_{k+1}^{-1} = \begin{pmatrix} \beta_k^{-1} & -\beta_k^{-1} v_k A_k^{-1} \\ -A_k^{-1} u_k \beta_k^{-1} & A_k^{-1} + A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1} \end{pmatrix}.$$

The solution z_{k+1} of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{pmatrix} f_k \\ d_k \end{pmatrix}$$

can be computed by

$$z_{k+1} = \begin{pmatrix} 0 \\ z_k \end{pmatrix} + \begin{pmatrix} I_k \\ -A_k^{-1} u_k \end{pmatrix} \beta_k^{-1}\, (f_k - v_k z_k).$$

Similarly for the reverse bordering method, starting from

$$A_{k+1}^{-1} = \begin{pmatrix} & p'_k & n'_k \\ & a'_k & v'_k \\ & u'_k & A'_k \end{pmatrix} \begin{matrix} \\ p'_k \\ n'_k \end{matrix}$$

we have

$$A_k^{-1} = A'_k - u'_k\, a'^{-1}_k\, v'_k.$$

The solution z_k of the system

$$A_k z_k = d_k$$

can be obtained from the solution z_{k+1} of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{pmatrix} f_k \\ d_k \end{pmatrix} \begin{matrix} p'_k \\ n'_k \end{matrix}.$$

We set

$$z_{k+1} = \begin{pmatrix} c_k \\ z'_k \end{pmatrix} \begin{matrix} p'_k \\ n'_k \end{matrix}$$

and we have

$$z_k = z'_k - u'_k a'^{-1}_k c_k.$$

Thus the block bordering method and the reverse bordering method can be applied in the following two cases.

Case 1.

$$\left(\begin{array}{c|c} A_k & u_k \\ \hline v_k & a_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} d_k \\ \hline f_k \end{array}\right).$$

Case 2.

$$\left(\begin{array}{c|c} a_k & v_k \\ \hline u_k & A_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} f_k \\ \hline d_k \end{array}\right).$$

There are also two other possibilities of bordering that could be investigated.

Case 3.

$$\left(\begin{array}{c|c} u_k & A_k \\ \hline a_k & v_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} d_k \\ \hline f_k \end{array}\right).$$

This case can be treated the same as Case 2 because we can put the last p_k rows of the matrix and the right-hand side on the top and all the formulæ for the methods are the same.

Case 4.

$$\left(\begin{array}{c|c} v_k & a_k \\ \hline A_k & u_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} f_k \\ \hline d_k \end{array}\right).$$

This case can be treated the same as Case 1 for the reason explained in Case 3.

Two particular cases can be interesting since they have many applications.

Let us first consider the case where A_k is a Hankel matrix; that is, when its elements a_{ij} are such that $a_{ij} = c_{i+j}$ where the c_i are given complex numbers. In this case, the reverse bordering method must be applied in its normal formulation because the structure of the inverse matrix does not permit any simplification.

Let us now consider the case of Toeplitz matrices. Let $A_k = (a_{ij})$ be the Hermitian positive definite Toeplitz matrix, built from a sequence of complex numbers $c_0, c_1, c_2, \ldots$. Thus we have $a_{ij} = c_{i-j}$ (for $i, j = 0, \ldots, n_k - 1$), $c_l = \overline{c}_{-l}$, and

$$A_k = T^{(0)}_{n_k} = \begin{pmatrix} c_0 & \overline{c}_1 & \overline{c}_2 & \cdots & \overline{c}_{n_k-1} \\ c_1 & c_0 & \overline{c}_1 & \cdots & \overline{c}_{n_k-2} \\ \vdots & \vdots & \vdots & & \vdots \\ c_{n_k-1} & c_{n_k-2} & c_{n_k-3} & \cdots & c_0 \end{pmatrix}.$$

In this case, A_{k+1} can be obtained from A_k by bordering either at the bottom and on the right or at the top and on the left. In both cases, due to the particular structure of the matrix, we have $v_k = \overline{u}_k^{\mathrm{T}}$.

Thus we can choose between two bordered matrices of dimension $n_{k+1} = n_k + p_k$

$$A_{k+1} = \begin{pmatrix} A_k & u_k \\ \overline{u}_k^{\mathrm{T}} & a_k \end{pmatrix} \quad \text{or} \quad A'_{k+1} = \begin{pmatrix} a_k & \overline{w}_k^{\mathrm{T}} \\ w_k & A_k \end{pmatrix}.$$

Obviously the two possibilities are not equivalent and the systems to be solved are different. However, in both cases, the bordering method can be applied after the simplification due to the special structure of the matrices.

If we consider the reverse unit matrix J (i.e., the unit matrix with its columns in the reverse order) of order n_k and the reverse unit matrix J' of order p_k, we have

$$w_k = J\overline{u}_k J'.$$

6. Numerical examples. When solving a system of linear equations by the bordering method some intermediate systems can be nearly singular. In that case, the block bordering method described in [5] allows us to jump over these near-singularities and the numerical stability of the process is thus improved.

Before giving a numerical example, let us discuss our strategy for deciding when and how far to jump. This strategy is based on the relation

$$\det A_{k+1} = \det A_k \cdot \det \beta_k.$$

Assuming that A_k^{-1} has already been obtained, we first add one new row and one new column to the matrix A_k; that is, we use the formulæ of §2 (or, equivalently, those of §3 with $p_k = 1$). If

$$|\beta_k| \leq \varepsilon$$

for some given $\varepsilon > 0$, we will add one more new row and one more new column to A_k and use the formulæ of §3 with $p_k = 2$. If

$$|\det \beta_k| \leq \varepsilon$$

we again add one new row and one new column to A_k and repeat the process until, after having added p_k new rows and columns, we obtain a matrix β_k such that

$$|\det \beta_k| > \varepsilon.$$

Then A_{k+1}^{-1} and z_{k+1} can be computed by the formulæ of §3. Let us also mention that the determinant of β_k is computed as the product of the pivots in a Gaussian elimination process. Such a strategy avoids the inversion of nearly singular matrices β_k, thus improving the numerical stability of the bordering method as shown by the following examples.

We first consider the system

$$\begin{pmatrix} 1 & 1 & 1 & 1 & -1 & 0 & -1 \\ 1 & 1 & 2 & 0 & 1 & 1 & -1 \\ 1 & 1 & -1 & 0 & 2 & -2 & 0 \\ -1 & 1 & 2 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} = \begin{pmatrix} -2 \\ 13 \\ -2 \\ 22 \\ 25 \\ 18 \\ 9 \end{pmatrix}.$$

TABLE 1
Solutions with the bordering method ($\eta = 10^{-14}$).

n							
1	−0.2000(1)						
2	−0.1501(16)	0.1501(16)					
3	−0.1001(16)	0.1001(16)	0.5000(1)				
4	−0.4500(1)	0.7500(1)	0.4992(1)	−0.1001(2)			
5	−0.2950(2)	−0.9297(1)	0.1333(2)	0.4833(2)	0.2500(2)		
6	−0.7006(15)	−0.1171(16)	0.9348(15)	0.1635(16)	0.7006(15)	−0.7006(15)	
7	0.9222	0.1859(1)	0.3006(1)	0.3570(1)	0.5078(1)	0.5922(1)	0.7000(1)

TABLE 2
Solutions with the block bordering method ($\eta = 10^{-14}, \varepsilon = 10^{-14}$).

n							
1	−0.2000(1)						
2	———	———					
3	———	———	———				
4	−0.4500(1)	0.7500(1)	0.5000(1)	−0.1000(2)			
5	−0.2950(2)	−0.9167(1)	0.1333(2)	0.4833(2)	0.2500(2)		
6	———	———	———	———	———	———	
7	0.1000(1)	0.2000(1)	0.3000(1)	0.4000(1)	0.5000(1)	0.6000(1)	0.7000(1)

In this system, the subsystems of dimensions 2, 3, and 6 are singular. Thus we add a perturbation η to a_{11} and a_{55}. So that the solution of the system remains the same, we also add η to the first component of the right-hand side and 5η to its fifth component.

Using the bordering and the block bordering methods for solving this system, we obtain the results in Tables 1 and 2 (ε denotes the threshold under which the block bordering method jumps and the numbers in parentheses denote the powers of 10).

However, in some applications, it is necessary to compute the solutions of all the intermediate systems. For example, this is the case in the computation of orthogonal polynomials [9], the Padé approximation, and the implementation of the progressive forms of extrapolation processes where the first step consists of the computation of the first descending diagonal of the triangular array [4]. In such cases, the block bordering method allows us to jump over the near-singular systems and then the reverse bordering method allows us to compute afterwards their solution with an improved numerical stability.

Let us first discuss the strategy used in the reverse bordering method. We assume that A_k^{-1} and A_{k+1}^{-1} are known and we want to compute the solutions of the intermediate systems of dimensions $n_{k+1} - 1, \ldots, n_k + 1$, which were skipped in the block bordering method when climbing to higher dimensions. We begin by deleting the last row and the last column of A_{k+1}^{-1}, that is, we use the formulæ of §4 with $p'_k = 1$ and $n'_k = n_{k+1} - 1$. For that, we must compute $a_k'^{-1}$. If a'_k is nearly singular, we delete the last two rows and the last two columns of A_{k+1}^{-1}; that is, we use the formulæ of §4 with $p'_k = 2$ and $n'_k = n_{k+1} - 2$ and so on until a nonnearly singular matrix a'_k has been obtained. The near singularity of a'_k is tested by computing its determinant (again by Gaussian elimination) and checking to see if $|\det a'_k| \leq \varepsilon'$ or not. However, if, in this test, we take $\varepsilon' \geq \varepsilon$ (where ε is the threshold used in the block bordering method) then a jump will occur from n_{k+1} to n_k and the intermediate systems that were not solved when climbing to higher dimensions will again be skipped. Thus we must choose $\varepsilon' < \varepsilon$.

TABLE 3
Solutions with the block bordering method and the reverse bordering method ($\eta = 10^{-14}, \varepsilon = 10^{-14}, \varepsilon' = 10^{-20}$).

n							
1	−0.2000(1)						
2	−0.1501(16)	0.1501(16)					
3	−0.1001(16)	0.1001(16)	0.5000(1)				
4	−0.4500(1)	0.7500(1)	0.5000(1)	−0.1000(2)			
5	−0.2950(2)	−0.9167(1)	0.1333(2)	0.4833(2)	0.2500(2)		
6	−0.7006(15)	−0.1168(16)	0.9341(15)	0.1635(16)	0.7006(15)	−0.7006(15)	
7	0.1000(1)	0.2000(1)	0.3000(1)	0.4000(1)	0.5000(1)	0.6000(1)	0.7000(1)

Let us now return to our previous example. Table 3 shows this improvement (ε' is the threshold for jumping in the reverse bordering method).

Let us now give an example with the ε-algorithm. This algorithm is an extrapolation process whose theory can be found in [4]. It can be interpreted as solving the system

$$\begin{cases} a_0 S_0 + a_1 S_1 + \cdots + a_k S_k = 1 \\ a_0 S_1 + a_1 S_2 + \cdots + a_k S_{k+1} = 1 \\ \vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ a_0 S_k + a_1 S_{k+1} + \cdots + a_k S_{2k} = 1 \end{cases}$$

and then computing [1]

$$\varepsilon_{2k}^{(0)} = 1 / \sum_{i=0}^{k} a_i.$$

Let us apply the ε-algorithm to the partial sums of the series expansion of

$$f(x) = \frac{1 + b_1 x + \cdots + b_{m-1} x^{m-1} + x^m}{1 + x^m}.$$

Thanks to the theory of the ε-algorithm and its connection with Padé approximants (see the next section), we should have

$$\varepsilon_{2m}^{(0)} = f(x).$$

With $\eta = 0.25, m = 10, x = 2$, and $b_i = i \cdot \eta$, we have $f(x) = 2.998536585365854$. We set $\varepsilon = 10^{-6}$ and $\varepsilon' = 10^{-30}$ for the block bordering and reverse bordering methods and we obtain the following results for $\varepsilon_{2k}^{(0)}$. R means that the corresponding value was obtained by the reverse bordering method.

k	Bordering method	Block and reverse	
0	1.000000000000000	1.000000000000000	
1	0.833333333333333	0.833333333333333	
2	1.500000000000007	1.500000000000007	
3	1.500000000000006	1.500000000000007	R
4	1.500000000000004	1.499999999999987	R
5	1.500000000000004	1.499999999999984	R
6	1.500000000000006	1.500000000000006	R
7	1.500000000000007	1.500000000000006	
8	1.057370161706715	1.061205132114060	
9	5.442384375014805	5.530461077969034	
10	2.965829933964836	2.998536585365856	

Thus, only two exact digits are obtained for $\varepsilon_{10}^{(0)}$ with the bordering method and fifteen exact digits with the block bordering and reverse bordering methods.

Let us now take $\eta = 0.1, m = 10, x = 1, b_i = \eta/i$, and $\varepsilon = 10^{-3}$.

We have $f(x) = 1.141448412698413$ and we obtain

k	Bordering method	Block and reverse	
0	1.000000000000000	1.000000000000000	
1	1.200000000000000	1.200000000000000	
2	1.299999999999898	1.299999999999898	
3	1.366666666630687	1.366666666630683	R
4	1.416666663166007	1.416666665980086	R
5	1.416669182535733	1.416669185348150	R
6	1.370133070676631	1.370133070822927	R
7	1.363779438853716	1.363779438821982	
8	1.299241827763746	1.299241827748625	
9	1.221626694740750	1.221626694709201	
10	1.141448412733146	1.141448412698013	

Again the precision has been improved by the reverse bordering method.

In both examples, the subsystems of dimensions 3, 4, 5, and 6 were nearly singular and their solutions were obtained by the reverse bordering method from the solution of the system of dimension 7.

7. Application to Padé approximants. An important application of the bordering and the reverse bordering methods is the recursive computation of Padé approximants. We now recall the necessary definitions (see [2]).

A Padé approximant is a rational function whose series expansion in ascending powers of the variable agrees with a given power series f up to the term whose degree is the sum of the degrees of its numerator and its denominator. Such a Padé approximant is denoted by $[p/q]_f(x)$, where p is the degree of the numerator and q the degree of the denominator. By definition we have

$$[p/q]_f(x) - f(x) = O(x^{p+q+1}).$$

Let us define the linear functional c on the vector space of polynomials by

$$c\left(x^i\right) = c_i \quad \text{for } i = 0, 1, \ldots.$$

We consider the polynomial P_k of degree k belonging to the family of orthogonal polynomials with respect to c; that is,

$$P_k(x) = b_0 + b_1 x + \cdots + b_k x^k$$

such that

$$c\left(x^i P_k(x)\right) = \sum_{j=0}^{k} c_{i+j}\, b_j = 0 \quad \text{for } i = 0, 1, \ldots, k-1.$$

One of the b_i's is arbitrary and we choose the normalization $b_0 = 1$.

The coefficients $b_1, \ldots, b_k$ are obtained as the solution of a linear system

$$\begin{pmatrix} c_1 & c_2 & \cdots & c_k \\ c_2 & c_3 & \cdots & c_{k+1} \\ \vdots & \vdots & & \vdots \\ c_k & c_{k+1} & \cdots & c_{2k-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix} = - \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{pmatrix}.$$

If we denote by A_k the Hankel matrix of the preceding system, by d_k its right-hand side, and by $z_k = (b_1, \ldots, b_k)^{\mathrm{T}}$ its solution, we can solve this system by the block bordering and the reverse bordering methods.

We consider the formal power series

$$f(x) = c_0 + c_1 x + c_2 x^2 + \cdots$$

and the polynomial $\widetilde{P}_k(x)$ given by

$$\widetilde{P}_k(x) = x^k P_k\left(x^{-1}\right) = b_k + b_{k-1} x + \cdots + b_1 x^{k-1} + x^k.$$

From the connections between Padé approximants and orthogonal polynomials, we know that $\widetilde{P}_k(x)$ is the denominator of the Padé approximant $[k-1/k]_f(x)$. If we want to have the normalization $b_k = 1$ one can simply consider the polynomial

$$P_k^*(x) = b_k^{-1} \widetilde{P}_k(x).$$

Thus the block bordering and the reverse bordering methods allow us to compute recursively the coefficients of the denominators of the Padé approximants [0/1], [1/2], [2/3], ...

To control the accuracy and numerical stability of the reverse bordering method, we take f as the power series expansion of a rational function with numerator of degree $k-1$ and denominator of degree k. In that case, the Padé approximant $[k-1/k]_f$ must be identical to f. We set

$$f(x) = \frac{1 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_{k-1} x^{k-1}}{1 + \beta_0 x^k} = c_0 + c_1 x + c_2 x^2 + \cdots.$$

Giving some values to the α_i's and to the β_i's, we compute the c_i's so that $[k-1/k]_f = f$; that is, in order to have

$$P_k^* = 1 + \beta_0 x^k = 1 + \frac{b_0}{b_k} x^k,$$

where $b_0 = 1$, b_k is the coefficient of x^k in the orthogonal polynomial P_k, and all the $b_i, i = 1, \ldots, k-1$ are zero. The α_i's depend on a parameter η and we give to it different values.

We set ε as the threshold under which the block bordering method jumps, and ε' as the threshold under which the reverse bordering method jumps.

In the following examples, we give the residual $r_k = |A_k z_k - d_k|$, where z_k is the vector of the coefficients $b_i, i = 1, \ldots, k$ of the polynomial P_k (the numbers in parenthesis denote again the powers of 10). The coefficients of the polynomial P_k^*, which is the denominator of $[k-1/k] = f$, are also given.

7.1. Example 1. We consider the function

$$f(x) = \frac{1 + \eta x + \eta x^2}{1 - x^3} = 1 + \eta x + \eta x^2 + x^3 + \eta x^4 + \eta x^5 + x^6 + \cdots.$$

We should have $[2/3]_f = f$, that is, $P_3^* = 1 - x^3$.

$\eta = 10^{-5}$							
k	Bordering method no jump			Block and reverse $\varepsilon = 10^{-4}, \varepsilon' = 10^{-10}$			
1	.11(−15)			.0			R
2	.11(−15)	.46(−16)		.0	.16(−15)		R
3	.0	.21(−16)	.50(−11)	.22(−15)	.17(−20)	.0	
	$P_3^* = 1 - .28 \cdot 10^{-16}x - .49 \cdot 10^{-11}x^2 - x^3$			$P_3^* = 1 + .25 \cdot 10^{-21}x - .23 \cdot 10^{-20}x^2 - x^3$			

$\eta = 10^{-10}$							
k	Bordering method no jump			Block and reverse $\varepsilon = 10^{-9}, \varepsilon' = 10^{-15}$			
1	.0			.0			R
2	.0	.83(−17)		.0	.83(−17)		R
3	.0	.25(−16)	.0	.0	.0	.0	
	$P_3^* = 1 - .25 \cdot 10^{-16}x - x^3$			$P_3^* = 1 + .61 \cdot 10^{-26}x - x^3$			

7.2. Example 2. We consider the function

$$f(x) = \frac{1 + \eta x + 2\eta x^2 + 3\eta x^3}{1 - x^4} = 1 + \eta x + 2\eta x^2 + 3\eta x^3 + x^4 + \eta x^5 + 2\eta x^6 + 3\eta x^7 + \cdot$$

We should have $[3/4]_f = f$, that is, in particular, $P_4^* = 1 - x^4$.
The system to be solved is

$$\begin{pmatrix} \eta & 2\eta & 3\eta & 1 \\ 2\eta & 3\eta & 1 & \eta \\ 3\eta & 1 & \eta & 2\eta \\ 1 & \eta & 2\eta & 3\eta \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = - \begin{pmatrix} 1 \\ \eta \\ 2\eta \\ 3\eta \end{pmatrix}.$$

$\eta = 10^{-4}$					
k	Bordering method no jump				
1	.0				
2	.22(−15)	.51(−15)			
3	.0	.66(−15)	.47(−11)		
4	.44(−15)	.55(−15)	.25(−12)	.61(−12)	

$P_4^* = 1 + .75 \cdot 10^{-15}x - .24 \cdot 10^{-12}x^2 - .61 \cdot 10^{-12}x^3 - x^4$

$\eta = 10^{-4}$					
k	Block and reverse $\varepsilon = 10^{-4}$, $\varepsilon' = 10^{-10}$				
1	.0				
2	.22(−15)	.51(−15)			R
3	.0	.23(−15)	.76(−16)		
4	.11(−15)	.20(−15)	.53(−15)	.61(−12)	

$P_4^* = 1 - .74 \cdot 10^{-16}x - .34 \cdot 10^{-15}x^2 - .61 \cdot 10^{-12}x^3 - x^4$

$\eta = 10^{-4}$					
k	Block and reverse $\varepsilon = 1.1 \cdot 10^{-4},\ \varepsilon' = 10^{-10}$				
1	.67(−15)				R
2	.20(−12)	.17(−12)			R
3	.11(−15)	.23(−15)	.52(−15)		
4	.11(−15)	.12(−15)	.92(−16)	.61(−12)	

$P_4^* = 1 + .6 \cdot 10^{-17}x + .91 \cdot 10^{-16}x^2 - .61 \cdot 10^{-12}x^3 - x^4$

$\eta = 10^{-4}$					
k	Block and reverse $\varepsilon = 10^{-3},\ \varepsilon' = 10^{-10}$				
1	.67(−15)				R
2	.67(−15)	.24(−15)			R
3	.11(−15)	.23(−15)	.76(−16)		R
4	.0	.0	.0	.0	

$P_4^* = 1 + .94 \cdot 10^{-21}x + .11 \cdot 10^{-19}x^2 - .24 \cdot 10^{-19}x^3 - x^4$

$\eta = 10^{-10}$				
k	Bordering method no jump			
1	.0			
2	.11(−15)	.26(−15)		
3	.0	.45(−15)	.35(−5)	
4	.22(−15)	.83(−16)	.36(−6)	.36(−16)

$P_4^* = 1 + .19 \cdot 10^{-15}x - .36 \cdot 10^{-6}x^2 - x^4$

$\eta = 10^{-10}$					
k	Block and reverse $\varepsilon = 10^{-9},\ \varepsilon' = 10^{-15}$				
1	.0				R
2	.0	.37(−15)			R
3	.0	.83(−17)	.41(−15)		R
4	.0	.0	.0	.52(−25)	

$P_4^* = 1 - .37 \cdot 10^{-26}x - .93 \cdot 10^{-26}x^2 - .51 \cdot 10^{-25}x^3 - x^4$

$\eta = 10^{-15}$				
k	Bordering method no jump			
1	.0			
2	.0	.16(−15)		
3	.44(−15)	.12(−14)	.96(−2)	
4	.22(−15)	.11(−14)	.96(−2)	.96(−17)

$P_4^* = 1 + .1 \cdot 10^{-14}x + .96 \cdot 10^{-2}x^2 - x^4$

$\eta = 10^{-15}$					
k	Block and reverse $\varepsilon = 10^{-14}$, $\varepsilon' = 10^{-20}$				
1	.89(−15)				R
2	.11(−14)	.47(−15)			R
3	.22(−15)	.33(−15)	.32(−17)		R
4	.0	.20(−30)	.0	.0	

$$P_4^* = 1 - .16 \cdot 10^{-30}x + .71 \cdot 10^{-31}x^2 - x^4$$

7.3. Example 3. We consider the function

$$f(x) = \frac{1 + x + \eta x^2 + 2\eta x^3 + 3\eta x^4}{1 - x^5}$$
$$= 1 + x + \eta x^2 + 2\eta x^3 + 3\eta x^4 + x^5 + x^6 + \eta x^7 + 2\eta x^8 + 3\eta x^9 + \cdot$$

We should have $[4/5]_f = f$, that is, $P_5^* = 1 - x^5$.

$\eta = 10^{-5}$					
k	Bordering method no jump				
1	.0				
2	.0	.0			
3	.22(−15)	.0	.29(−15)		
4	.22(−15)	.22(−15)	.51(−15)	.62(−11)	
5	.22(−15)	.22(−15)	.47(−15)	.84(−11)	.11(−15)

$$P_5^* = 1 - .41 \cdot 10^{-11}x + .41 \cdot 10^{-11}x^2 + .41 \cdot 10^{-11}x^3 - .41 \cdot 10^{-11}x^4 - x^5$$

$\eta = 10^{-5}$						
k	Block and reverse $\varepsilon = 10^{-4}$, $\varepsilon' = 10^{-10}$					
1	.0					
2	.0	.0				R
3	.11(−15)	.0	.16(−15)			R
4	.0	.0	.46(−16)	.73(−16)		
5	.22(−15)	.0	.61(−16)	.26(−15)	.30(−15)	

$$P_5^* = 1 - .15 \cdot 10^{-16}x + .76 \cdot 10^{-16}x^2 + .18 \cdot 10^{-15}x^3 + .12 \cdot 10^{-15}x^4 - x^5$$

$\eta = 10^{-15}$					
k	Bordering method no jump				
1	.0				
2	.0	.0			
3	.0	.22(−15)	.11(−15)		
4	.22(−15)	.11(−15)	.11(−15)	.55(−1)	
5	.22(−15)	.11(−15)	.15(−17)	.11	.59(−15)

$$P_5^* = 1 + .6 \cdot 10^{-1}x - .6 \cdot 10^{-1}x^2 - .6 \cdot 10^{-1}x^3 + .6 \cdot 10^{-1}x^4 - 1.06x^5$$

$\eta = 10^{-15}$						
k	Block and reverse $\varepsilon = 10^{-14}$, $\varepsilon' = 10^{-20}$					
1	.0					
2	.0	.22(−15)				R
3	.11(−15)	.22(−15)	.11(−15)			R
4	.0	.11(−15)	.33(−15)	.33(−15)		
5	.0	.0	.11(−15)	.44(−15)	.22(−15)	

$$P_5^* = 1 + .11 \cdot 10^{-15}x - .22 \cdot 10^{-15}x^2 - .22 \cdot 10^{-15}x^3 - x^5$$

8. Conclusions. In Gaussian elimination, pivotal strategies are often necessary to ensure a better numerical stability. In particular, they avoid division by numbers close to zero (which are possibly due to cancellation errors in the previous steps), thus preventing possible catastrophic errors. The block bordering method provides a similar strategy in a different context for the same drawback. However, with such a strategy, the solutions of the intermediate systems that were skipped when climbing to higher and higher dimensions are not computed. It was the purpose of this paper to propose an algorithm (the reverse bordering method) for obtaining these solutions.

REFERENCES

[1] C. BREZINSKI, *Résultats sur les procédés de sommation et sur l'ε-algorithme*, RIRO, R3 (1970), pp. 147–153.

[2] ———, *Padé-type Approximation and General Orthogonal Polynomials*, Birkhäuser-Verlag, Basel, 1980.

[3] ———, *Bordering methods and progressive forms for sequence transformations*, Zastos. Mat., 20 (1990), pp. 435–443.

[4] C. BREZINSKI AND M. REDIVO-ZAGLIA, *Extrapolation Methods. Theory and Practice*, North-Holland, Amsterdam, 1991.

[5] C. BREZINSKI, M. REDIVO-ZAGLIA, AND H. SADOK, *A breakdown-free Lanczos type algorithm for solving linear systems*, Numer. Math., 63 (1992), pp. 29–38.

[6] W. J. DUNCAN, *Some devices for the solution of large sets of simultaneous linear equations*, Philos. Mag. Ser. 7, 35 (1944), pp. 660–670.

[7] V. N. FADDEEVA, *Computational Methods of Linear Algebra*, Dover, New York, 1959.

[8] W. W. HAGER, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239.

[9] M. M. CECCHI AND M. REDIVO-ZAGLIA, *A new recursive algorithm for a Gaussian quadrature formula via orthogonal polynomials*, in Orthogonal Polynomials and Their Applications, C. Brezinski, L. Gori, and A. Ronveaux, eds., Baltzer, Basel, 1991, pp. 353–358.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 938–942, July 1994

015

SOME SPECTRAL PROPERTIES OF HERMITIAN TOEPLITZ MATRICES*

WILLIAM F. TRENCH†

Abstract. Necessary conditions are given for the Hermitian Toeplitz matrix $T_n = (t_{r-s})_{r,s=1}^n$ to have a repeated eigenvalue λ with multiplicity $m > 1$ and for an eigenpolynomial of T_n associated with λ to have a given number of zeros off the unit circle $|z| = 1$. It is assumed that $t_r = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)e^{-ir\theta}\,d\theta$ $(0 \le r \le n-1)$, where f is real-valued and in $L(-\pi,\pi)$. The conditions are given in terms of the number of changes in sign of $f(\theta) - \lambda$.

Key words. Toeplitz, Hermitian, eigenvalue, eigenvector, eigenpolynomial

AMS subject classifications. 15A18, 15A42

1. Introduction.

We consider the Hermitian Toeplitz matrix

$$T_n = (t_{r-s})_{r,s=1}^n,$$

where

$$t_r = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)e^{-ir\theta}\,d\theta, \qquad r = 0, 1, \ldots, n-1, \tag{1}$$

and f is real-valued and Lebesgue integrable on $(-\pi,\pi)$ and not constant on a set of measure 2π.

Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the eigenvalues of T_n, with associated orthonormal eigenvectors $x_1, x_2, \ldots, x_n$. Our first main result (Theorem 2.1) presents a necessary condition on f for λ_r to have multiplicity $m > 1$. To describe our second main result we first recall some well-known properties of eigenvectors of Hermitian Toeplitz matrices. If J is the $n \times n$ matrix with ones on the secondary diagonal and zeros elsewhere, then $JT_nJ = \overline{T}_n$. This implies that a vector x_r is a λ_r-eigenvector of T_n if and only if $J\overline{x}_r$ is. It follows that if λ_r has multiplicity one then

$$J\overline{x}_r = \xi x_r, \tag{2}$$

where ξ is a complex constant with modulus one. A stronger result holds if T_n is real and symmetric: Cantoni and Butler [1] have shown that in this case (even if T_n has repeated eigenvalues) R^n has an orthonormal basis consisting of $\lceil n/2 \rceil$ eigenvectors of T_n for which (2) holds with $\xi = 1$ and $\lfloor n/2 \rfloor$ for which (2) holds with $\xi = -1$.

The polynomial

$$X_r(z) = [1, z, \ldots, z^{n-1}]x_r \tag{3}$$

is said to be an *eigenpolynomial of* T_n *associated with* λ_r. The location of the zeros of the eigenpolynomials of Hermitian Toeplitz matrices is of interest in signal processing applications [2]–[5], [7]. If x_r satisfies (2) then

$$X_r(z) = \overline{\xi} z^{n-1}\overline{X_r(1/\overline{z})};$$

* Received by the editors October 19, 1992; accepted for publication (in revised form) February 16, 1993. This work was partially supported by National Science Foundation grants DMS-8907939 and DMS-9108254.

† Trinity University, 715 Stadium Drive, San Antonio, Texas 78212 (wtrench@trinity.edu).

hence, zeros of $X_r(z)$ that are not on the unit circle must occur in pairs ζ and $1/\overline{\zeta}$.

Gueguen proved the following theorem in [5]. (See also [2] and [4].)

THEOREM 1.1. *Let λ_r be an eigenvalue of T_n, but not of T_{n-1}. Then its associated eigenpolynomial $X_r(z)$ has at least $|n-2r+1|$ zeros on the unit circle $|z|=1$.*

Delsarte, Genin, and Kamp proved the following theorem in [3]. (See also [4].)

THEOREM 1.2. *Suppose that the eigenvalue λ_r of T_n has multiplicity m and let s be the largest integer $< n$ such that λ_r is not an eigenvalue of T_s. Then any eigenpolynomial $X(z)$ of T_n corresponding to λ_r has at least $|n-m-2r+2|$ and at most $m+s-1$ zeros on the unit circle $|z|=1$.*

Our second main result (Theorem 3.1) gives a necessary condition on f for an eigenpolynomial of T_n satisfying (2) to have a given number of zeros that are not on the unit circle.

2. A necessary condition for repeated eigenvalues.. Let α and β be the essential upper and lower bounds of f; that is, α is the largest number and β the smallest such that $\alpha \le f(\theta) \le \beta$ almost everywhere on $(-\pi,\pi)$. It is known [6, p. 65] that all the eigenvalues of T are in (α,β). A proof of this is included naturally in the proof of the following theorem.

THEOREM 2.1. *If λ_r is an eigenvalue of T_n with multiplicity m, then $f(\theta)-\lambda_r$ must change sign at least $2m-1$ times in $(-\pi,\pi)$.*

Proof. Associate with each vector $v=[v_1,v_2,\dots,v_n]^t$ in C^n the polynomial

$$V(z)=[1,z,\dots,z^{n-1}]v=\sum_{j=1}^{n} v_j z^{j-1}.$$

If u and v are in C^n then

$$(u,v)=\frac{1}{2\pi}\int_{-\pi}^{\pi} U(z)\overline{V(z)}\,d\theta, \tag{4}$$

where $z=e^{i\theta}$ whenever z appears in an integral. Moreover, (1) implies that

$$(T_n u,v)=\frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)U(z)\overline{V(z)}\,d\theta. \tag{5}$$

Now let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the eigenvalues of T_n, with corresponding orthonormal eigenvectors $x_1,x_2,\dots,x_n$, and let

$$X_i(z)=[1,z,\dots,z^{n-1}]x_i, \qquad 1\le i\le n,$$

be the corresponding eigenpolynomials. From (4),

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} X_i(z)\overline{X_j(z)}\,d\theta=\delta_{ij}, \qquad 1\le i,j\le n \tag{6}$$

and from (5),

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)X_i(z)\overline{X_j(z)}\,d\theta=\delta_{ij}\lambda_j, \qquad 1\le i,j\le n. \tag{7}$$

The last two equations with $i=j$ show that the eigenvalues of T_n are in (α,β). Therefore, $f(\theta)-\lambda_r$ must change sign at some point in $(-\pi,\pi)$. This completes the proof if $m=1$.

Now suppose that $m > 1$ and $f(\theta) - \lambda_r$ changes sign only at the points $\theta_1 < \theta_2 < \cdots < \theta_k$ in $(-\pi, \pi)$, where $k \le 2m-2$. We will show that this assumption leads to a contradiction.

Define

$$g(\theta) = \frac{1}{2\pi}\left(f(\theta) - \lambda_r\right). \tag{8}$$

For reference below note that if $k = 2p$ then the function

$$g(\theta)\prod_{j=1}^{2p}\sin\left(\frac{\theta-\theta_j}{2}\right) \tag{9}$$

does not change sign in $(-\pi, \pi)$. This remains true if $k = 2p-1$, if we define $\theta_{2p} = \pi$. Now suppose that λ_r has multiplicity m; that is,

$$\lambda_r = \lambda_{r+1} = \cdots = \lambda_{r+m-1}. \tag{10}$$

From (6), (7), and (10),

$$\int_{-\pi}^{\pi} g(\theta)X_i(z)\overline{X_j(z)}\,d\theta = 0 \qquad (r \le i \le r+m-1,\ 1 \le j \le n).$$

Therefore

$$\int_{-\pi}^{\pi} g(\theta)\left(\sum_{\ell=0}^{m-1} c_\ell X_{r+\ell}(z)\right)\overline{X_j(z)}\,d\theta = 0, \qquad 1 \le j \le n,$$

if $c_0, \ldots, c_{m-1}$ are constants. This implies that

$$\int_{-\pi}^{\pi} g(\theta)\left(\sum_{\ell=0}^{m-1} c_\ell X_{r+\ell}(z)\right)\overline{Q(z)}\,d\theta = 0 \tag{11}$$

if Q is any polynomial of degree $\le n-1$, since any such polynomial can be written as a linear combination of $X_1(z), \ldots, X_n(z)$. In particular, choose $c_0, \ldots, c_{m-1}$—not all zero—so that

$$\sum_{\ell=0}^{m-1} c_\ell X_{r+\ell}(e^{i\theta_j}) = 0, \qquad 1 \le j \le p$$

(this is possible, since $p < m$), and let

$$Q(z) = \left(\sum_{\ell=0}^{m-1} c_\ell X_{r+\ell}(z)\right)\prod_{j=1}^{p}\frac{z - e^{i\theta_{p+j}}}{z - e^{i\theta_j}}.$$

Substituting this into (11) yields

$$\int_{-\pi}^{\pi} g(\theta)\left|\sum_{\ell=0}^{m-1} c_\ell X_{r+\ell}(z)\right|^2 \prod_{j=1}^{p}\frac{\overline{z} - e^{-i\theta_{p+j}}}{\overline{z} - e^{-i\theta_j}}\,d\theta = 0,$$

or, equivalently,

$$\int_{-\pi}^{\pi} g_1(\theta) \prod_{j=1}^{p} (z - e^{i\theta_j})(\overline{z} - e^{-i\theta_{p+j}})\, d\theta = 0, \tag{12}$$

where

$$g_1(\theta) = g(\theta) \left| \frac{\sum_{\ell=0}^{m-1} c_\ell X_{r+\ell}(z)}{\prod_{j=1}^{p} (z - e^{i\theta_j})} \right|^2.$$

If $z = e^{i\theta}$ then

$$(z - e^{i\theta_j})(\overline{z} - e^{-i\theta_{p+j}}) = 4e^{i(\theta_j - \theta_{p+j})/2} \sin\left(\frac{\theta - \theta_j}{2}\right) \sin\left(\frac{\theta - \theta_{p+j}}{2}\right);$$

hence, (12) implies that

$$\int_{-\pi}^{\pi} g_1(\theta) \prod_{j=1}^{2p} \sin\left(\frac{\theta - \theta_j}{2}\right) d\theta = 0,$$

which is impossible because of (8) and our observation that the function in (9) is sign constant on $(-\pi, \pi)$. □

Theorem 2.1 immediately implies the following theorems. Theorem 2.4 was proved in [8].

THEOREM 2.2. *If f is monotonic on $(-\pi, \pi)$ or there is a number ϕ in $(-\pi, \pi)$ such that f is monotonic on $(-\pi, \phi)$ and (ϕ, π), then all eigenvalues of T_n have multiplicity one.*

THEOREM 2.3. *Suppose that $f(-\theta) = f(\theta)$, so that T_n is a real symmetric Toeplitz matrix. If λ_r is an eigenvalue of T_n with multiplicity m, then $f(\theta) - \lambda_r$ must change sign at least m times in $(0, \pi)$*

THEOREM 2.4. *Suppose that $f(-\theta) = f(\theta)$ and f is monotonic on $(0, \pi)$. Then all the eigenvalues of T_n have multiplicity one.*

3. Location of the zeros of eigenpolynomials. The following theorem is the main result of this section.

THEOREM 3.1. *Suppose that the eigenvalue λ_r has an associated eigenvector x_r such that $J\overline{x}_r = \xi x_r$, where ξ is a constant, and the eigenpolynomial $X_r(z)$ defined in (3) has $2m$ zeros ($m \geq 1$) that are not on the unit circle. Then $f(\theta) - \lambda_r$ must change sign at least $2m + 1$ times in $(-\pi, \pi)$.*

Proof. The proof is by contradiction. Suppose $f(\theta) - \lambda_r$ changes sign only at the points $\theta_1 < \cdots < \theta_k$ in $(-\pi, \pi)$, where $1 \leq k \leq 2m$. Then, as in the proof of Theorem 2.1, the function (9) does not change sign in $(-\pi, \pi)$. (Again, $k = 2p$ if k is even, and we define $\theta_{2p} = \pi$ if $k = 2p - 1$.) From among the $2m$ zeros of $X_r(z)$ not on the unit circle choose $2p$ distinct zeros $\zeta_1, \ldots, \zeta_p, 1/\overline{\zeta}_1, \ldots, 1/\overline{\zeta}_p$, and define g as in (8).

From (6) and (7),

$$\int_{-\pi}^{\pi} g(\theta) X_r(z) \overline{X_s(z)}\, d\theta = 0 \, (1 \leq s \leq n),$$

which implies that

$$\int_{-\pi}^{\pi} g(\theta)X_r(z)\overline{Q(z)}\,d\theta = 0 \tag{13}$$

if Q is any polynomial of degree $\le n-1$.

Now define

$$q_j(z) = \frac{(z-e^{i\theta_j})(1-e^{-i\theta_{p+j}}z)}{(z-\zeta_j)(1-\overline{\zeta}_j z)}, \qquad 1 \le j \le p,$$

and let

$$Q(z) = X_r(z)q_1(z)\cdots q_p(z).$$

Then (13) implies that

$$\int_{-\pi}^{\pi} g(\theta)|X_r(z)|^2\overline{q_1(z)}\cdots\overline{q_p(z)}\,d\theta = 0. \tag{14}$$

However, if $z = e^{i\theta}$ then

$$q_j(z) = \frac{4e^{i(\theta_j-\theta_{p+j})/2}}{|1-\overline{\zeta}_j e^{i\theta}|^2}\sin\left(\frac{\theta-\theta_j}{2}\right)\sin\left(\frac{\theta-\theta_{p+j}}{2}\right).$$

This and (14) imply that

$$\int_{-\pi}^{\pi} \frac{g(\theta)|X_r(z)|^2}{\prod_{j=1}^{p}|1-\overline{\zeta}_j e^{i\theta}|^2}\prod_{j=1}^{2p}\sin\left(\frac{\theta-\theta_j}{2}\right)\,d\theta = 0, \tag{15}$$

which is impossible, since the function (9) is sign constant in $(-\pi,\pi)$. □

Theorem 3.1 immediately implies the following theorem.

THEOREM 3.2. *If f satisfies the hypotheses of either Theorem 2.2 or Theorem 2.4, then all zeros of the eigenpolynomials of T_n are on the unit circle $|z| = 1$.*

REFERENCES

[1] A. CANTONI AND F. BUTLER, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl., 13 (1976), pp. 275–288.

[2] P. DELSARTE AND Y. GENIN, *Spectral properties of finite Toeplitz matrices*, in Mathematical Theory of Networks and Systems, Proc. MTNS–83 International Symposium, Beer Sheva, Israel, 1983, pp. 194–213.

[3] P. DELSARTE, Y. GENIN, AND Y. KAMP, *Parametric Toeplitz systems*, Circuits, Systems, Signal Processing, 3 (1984), pp. 207–223.

[4] Y. GENIN, *A survey of the eigenstructure properties of finite Hermitian Toeplitz matrices*, Integral Equations Operator Theory, 10 (1987), pp. 621–639.

[5] C. GUEGUEN, *Linear prediction in the singular case and the stability of singular models*, Proc. Internat. Conf. Acoustics, Speech, Signal Processing, Atlanta, 1981, pp. 881–885.

[6] U. GRENANDER AND G. SZEGÖ, *Toeplitz Forms and Their Applications*, University of California Press, Berkeley, Los Angeles, CA, 1958.

[7] J. MAKHOUL, *On the eigenvectors of symmetric Toeplitz matrices*, IEEE Trans. Acoustics Speech, Signal Proc., ASSP-29 (1981), pp. 868–872.

[8] W. F. TRENCH, *Interlacement of the even and odd spectra of real symmetric Toeplitz matrices*, Linear Algebra Appl., 195 (1993), pp. 59–69.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 943–967, July 1994

016

THEORY OF DECOMPOSITION AND BULGE-CHASING ALGORITHMS FOR THE GENERALIZED EIGENVALUE PROBLEM*

DAVID WATKINS† AND LUDWIG ELSNER‡

Abstract. A generic GZ algorithm for the generalized eigenvalue problem $Ax = \lambda Bx$ is presented. This is actually a large class of algorithms that includes multiple-step QZ and LZ algorithms, as well as QZ-LZ hybrids, as special cases. First the convergence properties of the GZ algorithm are discussed, then a study of implementations is undertaken. The notion of an elimination rule is introduced as a device for studying the QZ, LZ and other algorithms simultaneously. To each elimination rule there corresponds an explicit GZ algorithm. Through a careful study of the steps involved in executing the explicit algorithm, it is discovered how to implement the algorithm implicitly by bulge chasing. The approach taken here was introduced by Miminis and Paige in the context of the QR algorithm for the ordinary eigenvalue problem. It is more involved than the standard approach, but it yields a much clearer picture of the relationship between the implicit and explicit versions of the algorithm. Furthermore, it is more general than the standard approach, as it does not require the use of a theorem of "Implicit-Q" type. Finally a generalization of the implicit GZ algorithm, the generic bulge-chasing algorithm, is introduced. It is proved that the generic bulge-chasing algorithm implicitly performs iterations of the generic GZ algorithm. Thus the convergence theorems that are proved for the generic GZ algorithm hold for the generic bulge-chasing algorithm as well.

Key words. generalized eigenvalue problem, QZ algorithm, GZ algorithm, chasing the bulge

AMS subject classifications. 65F15, 15A18

1. Introduction.

The standard algorithm for finding the eigenvalues of a dense, indefinite, matrix pencil $A - \lambda B$ with B nonsingular is the QZ algorithm of Moler and Stewart [11]. Related methods are the LZ algorithm of Kaufman [8] and the combination-shift QZ algorithm of Ward [14]. In this paper we introduce and study a generic GZ algorithm, which is actually a large class of algorithms that contains these and many other algorithms as special cases. For example, QZ-LZ hybrids are also included. Our coverage is not restricted to single- or double-step algorithms; we allow multiple steps of arbitrary multiplicity.

The QZ algorithm is an extension of the QR algorithm, which is one of the most widely used algorithms for the standard eigenvalue problem. The QR algorithm has both explicit and implicit versions. The explicit version is useful for introducing the algorithm and discussing theoretical aspects such as convergence theory, but it is usually the implicit version that is actually implemented. The standard approach to the QZ algorithm, as presented in contemporary textbooks [6], [12], mentions only an implicit version, which is interpreted as a way of applying the QR algorithm to the matrix AB^{-1} without actually forming AB^{-1} or even B^{-1}. Earlier approaches [11], [8] started from an explicit version and derived the implicit version therefrom. In every instance the focus was on the matrix AB^{-1}. Our approach also starts with an explicit version, but our explicit QZ algorithm differs from earlier formulations in that it effectively applies the QR algorithm to both AB^{-1} and $B^{-1}A$. The advantage of this approach is that it reveals symmetries in the algorithm that are obscured by the usual approaches. In particular, it puts the "Q" and "Z" transformations on an

* Received by the editors December 16, 1991; accepted for publication (in revised form) February 12, 1993.

† Department of Pure and Applied Mathematics, Washington State University, Pullman, Washington 99164-3113 (na.watkins@na-net.ornl.gov).

‡ Fakultät für Mathematik, Universität Bielefeld, Postfach 8640, W-4800 Bielefeld 1, Germany (elsner@math1.mathematik.uni-bielefeld.de).

equal footing. Of course our discussion is couched in more general terms. We consider a generic GZ algorithm that amounts to the generic GR algorithm [17] applied to the matrices $p(AB^{-1})$ and $p(B^{-1}A)$ simultaneously, where p is a polynomial whose degree is the multiplicity of the step. Our generic GZ algorithm is quite similar to the FGZ algorithm of [16].

The explicit algorithm is not a practical algorithm, because it would be too costly to implement and quite likely unstable as well. However, it is a useful vehicle for both the study of convergence and the introduction of implicit versions of the algorithm. We introduce the generic GZ algorithm in §2. In §3 we study the convergence properties of the algorithm and in §4 we consider questions of implementation. Sections 3 and 4 can be read independently of one another.

The convergence theorem that we prove in §3 is a generalization of a theorem on the convergence of the GR algorithm that we proved in [17]. The theorem says roughly that if the eigenvalues can be separated, and the shifts converge, and the condition numbers of the accumulated transforming matrices remain bounded, then the algorithm converges. We also introduce the generalized Rayleigh quotient shift strategy and discuss its asymptotic convergence properties without proof. Usually the convergence rate is quadratic.

In §4 we consider how to implement the GZ algorithm. Our approach is inspired by Miminis and Paige [10]. They showed that by taking a detailed look at how one would carry out an iteration of the QR algorithm in its explicit form, one can discover how it can be done implicitly. As Miminis and Paige pointed out, this approach is more involved than the usual approach, which invokes the Implicit-Q Theorem [6, p. 367], but it gives a much clearer picture of the relationship between the explicit and implicit versions of the algorithm. Miminis and Paige also stated that their approach is quite general. Our vehicle for introducing the desired generality is the idea of an elimination rule, which allows us to discuss the QZ, LZ, and all related algorithms simultaneously. Each elimination rule gives rise to a specific implementation of the GZ algorithm. Following Miminis and Paige, we take a close look at the steps involved in implementing the GZ algorithm explicitly. By studying the form of the intermediate matrices so produced, we discover how the algorithm can be implemented implicitly, that is, without forming or operating on the matrices $p(AB^{-1})$ or $p(B^{-1}A)$.

Once we have derived the implicit GZ algorithm, we introduce a generalization called the generic bulge-chasing algorithm and prove that each iteration of the generic bulge-chasing algorithm amounts to an iteration of the generic GZ algorithm. The purpose of making this last generalization is to allow additional flexibility in implementing the algorithm. This flexibility can be exploited to build more efficient and stable algorithms. In particular it allows the introduction of variants that do not break down when B happens to be singular. (For the originators of the QZ and LZ algorithms this was an important point.) When it comes to implementing the algorithm in practice, these are the variants that should be used.

In [18] we introduced a generic bulge-chasing algorithm for the standard eigenvalue problem. Our aim there was to lay common foundations for implicit versions of GR algorithms of all types (e.g., QR, LR with or without pivoting, SR, hybrids, etc.). To achieve the desired level of generality, we devised an approach that, like the Miminis–Paige approach, avoids using a theorem of the Implicit-Q type. However, unlike Miminis and Paige, we did not establish a close correspondence between the operations in the implicit and explicit versions of the algorithms. The results of this paper generalize those of [18].

2. The generic GZ algorithm. We consider the generalized eigenvalue problem

$$(A - \lambda B)x = 0,$$

where A and B are square matrices whose entries are complex numbers. Recall that the pencil $A - \lambda B$ is said to be *singular* if its determinant is zero for all λ and *regular* otherwise. We focus here on the regular case. If the given pencil is singular (or not known a priori to be regular), the staircase algorithm of Van Dooren [13] can be used to remove the singular part. (See also Demmel and Kågström [3], [4].) This algorithm also removes the infinite eigenvalue and its associated structure (which may be present if B is singular) and the zero eigenvalue and its associated structure (which may be present if A is singular). What is left is a regular pencil for which both A and B are nonsingular. We assume throughout (with few exceptions, when we explicitly state otherwise) that our pencil has a nonsingular B; we do not need to assume that A is nonsingular.[1]

Recall that the pencils $A - \lambda B$ and $\hat{A} - \lambda \hat{B}$ are said to be *strictly equivalent* if there exist nonsingular matrices G and Z such that

$$\hat{A} = G^{-1}AZ \quad \text{and} \quad \hat{B} = G^{-1}BZ.$$

Strictly equivalent pencils have the same eigenvalues, and the eigenvectors are related in a simple way through the transforming matrices G and Z. The generic GZ algorithm generates a sequence of strictly equivalent pencils $(A_i - \lambda B_i)$ that converges (we hope) to upper triangular or block triangular form, thus exposing the eigenvalues of the pencil. The eigenvectors can be found by a back-substitution process that utilizes the final upper-triangular matrices and the accumulated transforming matrices.

We assume that before we start our iterations of the GZ algorithm, we transform the pencil to some initial form

$$A_0 = G_0^{-1}AZ_0, \qquad B_0 = G_0^{-1}BZ_0.$$

For example, it is possible to make A_0 upper Hessenberg and B_0 upper triangular, as described in [6] and elsewhere. Later on we assume that A_0 and B_0 have this form, but for now we allow them to have any form; for example, we could take $G_0 = Z_0 = I$.

The ith iteration of the GZ algorithm transforms $A_{i-1} - \lambda B_{i-1}$ to $A_i - \lambda B_i$ by transformations obtained from GR decompositions. By a GR decomposition of a square matrix M, we mean any decomposition

$$M = GR$$

in which G is nonsingular and R is upper triangular. Every matrix has many different GR decompositions. To obtain A_i and B_i we first take GR decompositions of $p_i(A_{i-1}B_{i-1}^{-1})$ and $p_i(B_{i-1}^{-1}A_{i-1})$, where p_i is a polynomial. Thus we find nonsingular G_i and Z_i and upper triangular R_i and S_i such that

$$p_i(A_{i-1}B_{i-1}^{-1}) = G_iR_i \quad \text{and} \quad p_i(B_{i-1}^{-1}A_{i-1}) = Z_iS_i.$$

[1] However, if A is known to be nonsingular, one has the possibility of reversing the roles of A and B and considering the pencil $B - \mu A$, where $\mu = 1/\lambda$. If neither A nor B is known to be nonsingular, a prudent course of action is to run the staircase algorithm to determine the fine structure of the pencil.

We then let

$$A_i = G_i^{-1} A_{i-1} Z_i \quad \text{and} \quad B_i = G_i^{-1} B_{i-1} Z_i.$$

In the special case $B_{i-1} = I$, $Z_i = G_i$, $S_i = R_i$, this algorithm reduces to the generic GR algorithm for the standard eigenvalue problem.

The GZ algorithm is really a large class of algorithms. Specific instances are obtained by specifying the exact form of each GR decomposition and how the p_i are to be chosen. For example, variants of the QZ and LZ algorithms are obtained by specifying that each decomposition be a QR or LR decomposition, respectively. The p_i are chosen so that their roots, which we call the *shifts* for the ith iteration, are estimates of eigenvalues. The degree of p_i is called the *multiplicity* of the iteration.

We will see that it is possible to carry out the GZ iterations implicitly without even calculating matrices of the form AB^{-1} or $B^{-1}A$, much less $p(AB^{-1})$ or $p(B^{-1}A)$. Were this not the case, there would be no point in discussing these algorithms at all. First we look at convergence.

3. Convergence of GZ algorithms. An easy computation shows that

$$A_i B_i^{-1} = G_i^{-1} (A_{i-1} B_{i-1}^{-1}) G_i.$$

Since G_i was obtained from the decomposition $p_i(A_{i-1}B_{i-1}^{-1}) = G_i R_i$, we see that the transformation $A_{i-1}B_{i-1}^{-1} \to A_i B_i^{-1}$ is an iteration of the GR algorithm [17]. At the same time we have

$$B_i^{-1} A_i = Z_i^{-1} (B_{i-1}^{-1} A_{i-1}) Z_i,$$

where

$$p_i(B_{i-1}^{-1} A_{i-1}) = Z_i S_i,$$

so the transformation $B_{i-1}^{-1}A_{i-1} \to B_i^{-1}A_i$ is also a GR iteration. It follows from the theorems in [17] that both of the sequences $(A_i B_i^{-1})$ and $(B_i^{-1}A_i)$ converge to (block) upper triangular form, provided that the condition numbers of the accumulated transforming matrices $\hat{G}_i = G_1 \cdots G_i$ and $\hat{Z}_i = Z_1 \cdots Z_i$ remain bounded and the shifts converge, as $i \to \infty$. Preferably the shifts should converge to eigenvalues of the pencil, in which case the convergence of $(A_i B_i^{-1})$ and $(B_i^{-1}A_i)$ is superlinear.

We would like to be able to say something about the convergence of the sequences (A_i) and (B_i) separately, since these are the matrices with which we actually work. To do this we recall some nomenclature. Let $\mathcal{T}_d$ and $\mathcal{T}_r$ be subspaces of $\mathbb{C}^n$ of equal dimension. The pair $(\mathcal{T}_d, \mathcal{T}_r)$ is called a *deflating pair* for the regular pencil $A - \lambda B$ if and only if

$$A\mathcal{T}_d \subseteq \mathcal{T}_r \quad \text{and} \quad B\mathcal{T}_d \subseteq \mathcal{T}_r.$$

The subscripts d and r are mnemonics for domain and range, respectively. Since we are assuming that B is nonsingular, the condition $B\mathcal{T}_d \subseteq \mathcal{T}_r$ implies $B\mathcal{T}_d = \mathcal{T}_r$. Clearly $(\mathcal{T}_d, \mathcal{T}_r)$ is a deflating pair for $A - \lambda B$ if and only if $\mathcal{T}_d$ is invariant under $B^{-1}A$, $\mathcal{T}_r$ is invariant under AB^{-1}, and $B\mathcal{T}_d = \mathcal{T}_r$. The following lemma generalizes Lemma 6.1 of [17]. Here $d(\mathcal{S}, \mathcal{T})$ denotes the usual distance (or gap) between two subspaces and $\kappa_2(G)$ denotes the condition number of G with respect to the spectral norm.

LEMMA 3.1. *Let $A, B \in \mathbb{C}^{n\times n}$ and let $(\mathcal{T}_d, \mathcal{T}_r)$ be a deflating pair of k-dimensional subspaces for the pencil $A-\lambda B$. Let Z, $G \in \mathbb{C}^{n\times n}$ be nonsingular matrices, and let $\mathcal{S}_d$ and $\mathcal{S}_r$ be the spaces spanned by the first k columns of Z and G, respectively. (Think of $\mathcal{S}_d$ and $\mathcal{S}_r$ as approximations to $\mathcal{T}_d$ and $\mathcal{T}_r$, respectively.) Let C denote either A or B, and let $\hat{C} = G^{-1}CZ$. Consider the partition*

$$\hat{C} = \begin{bmatrix} \hat{C}_{11} & \hat{C}_{12} \\ \hat{C}_{21} & \hat{C}_{22} \end{bmatrix},$$

where $\hat{C}_{11} \in \mathbb{C}^{k\times k}$. Then

$$\frac{\|\hat{C}_{21}\|_2}{\|\hat{C}\|_2} \le \sqrt{2}\kappa_2(G)\kappa_2(Z)\left[d(\mathcal{S}_d, \mathcal{T}_d) + d(\mathcal{S}_r, \mathcal{T}_r)\right].$$

Proof. Let $Z = PU$, $G = QR$ be the QR decompositions of Z and G, respectively. Thus P and Q are unitary, and U and R are upper triangular. Partition these decompositions as

$$\begin{bmatrix} Z_1 & Z_2 \end{bmatrix} = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix},$$

where Z_1 and P_1 are $n \times k$, and similarly for the decomposition $G = QR$. Since $\hat{C} = G^{-1}CZ = R^{-1}Q^*CPU$, we have $\hat{C}_{21} = R_{22}^{-1}Q_2^*CP_1U_{11}$, from which

$$\|\hat{C}_{21}\|_2 \le \|R_{22}^{-1}\|_2 \|Q_2^*CP_1\|_2 \|U_{11}\|_2.$$

Since $\|R_{22}^{-1}\|_2 \le \|R^{-1}\|_2 = \|G^{-1}\|_2$, $\|U_{11}\|_2 \le \|U\|_2 = \|Z\|_2$, and $\|C\|_2 \le \|G\|_2\|\hat{C}\|_2\|Z^{-1}\|_2$, we see that

$$\frac{\|\hat{C}_{21}\|_2}{\|\hat{C}\|_2} \le \kappa_2(G)\kappa_2(Z)\frac{\|Q_2^*CP_1\|_2}{\|C\|_2}. \tag{1}$$

Since $Z_1 = P_1U_{11}$ and $G_1 = Q_1R_{11}$, we have $\mathcal{S}_d = \mathcal{R}(Z_1) = \mathcal{R}(P_1)$ and $\mathcal{S}_r^\perp = \mathcal{R}(G_1)^\perp = \mathcal{R}(Q_1)^\perp = \mathcal{R}(Q_2)$. Therefore, by Lemma 4.1 of [17], there exist $T_1 \in \mathbb{C}^{n\times k}$ and $T_2 \in \mathbb{C}^{n\times n-k}$ with orthonormal columns, such that $\mathcal{T}_d = \mathcal{R}(T_1)$, $\mathcal{T}_r^\perp = \mathcal{R}(T_2)$,

$$\|P_1 - T_1\|_2 \le \sqrt{2}d(\mathcal{S}_d, \mathcal{T}_d)$$

and

$$\|Q_2 - T_2\|_2 \le \sqrt{2}d(\mathcal{S}_r, \mathcal{T}_r).$$

We use here the fact that $d(\mathcal{S}_r, \mathcal{T}_r) = d(\mathcal{S}_r^\perp, \mathcal{T}_r^\perp)$. Now

$$\|Q_2^*CP_1\|_2 \le \|(Q_2 - T_2)^*CP_1\|_2 + \|T_2^*C(P_1 - T_1)\|_2 + \|T_2^*CT_1\|_2.$$

Since $\mathcal{R}(T_1) = \mathcal{T}_d$, $C\mathcal{T}_d \subseteq \mathcal{T}_r$, and $\mathcal{R}(T_2) = \mathcal{T}_r^\perp$, the product $T_2^*CT_1$ is zero. Thus

$$\begin{aligned}\|Q_2^*CP_1\|_2 &\le \|Q_2 - T_2\|_2\|C\|_2\|P_1\|_2 + \|T_2\|_2\|C\|_2\|P_1 - T_1\|_2 \\ &\le \sqrt{2}\,\|C\|_2\left[d(\mathcal{S}_d, \mathcal{T}_d) + d(\mathcal{S}_r, \mathcal{T}_r).\right]\end{aligned}$$

Combining this inequality with (1), we obtain the desired result. ∎

Define the cumulative transforming matrices by

$$\hat{G}_i = G_1 \cdots G_i, \qquad \hat{R}_i = R_i \cdots R_1,$$

$$\hat{Z}_i = Z_1 \cdots Z_i, \qquad \hat{S}_i = S_i \cdots S_1.$$

Then

$$C_i = \hat{G}_i^{-1} C_0 \hat{Z}_i,$$

where C can stand for either A or B. In the following theorem we prove the convergence of the GZ algorithm by applying Lemma 3.1 with the roles of C, G, Z, and $\hat{C}$ played by C_0, $\hat{G}_i$, $\hat{Z}_i$, and C_i, respectively. The symbol $\langle e_1, \ldots, e_k \rangle$ denotes the space spanned by the vectors $e_1, \ldots, e_k$.

THEOREM 3.2. *Let A_0, $B_0 \in \mathbb{C}^{n \times n}$ with B_0 nonsingular, and let p be a polynomial of degree $\leq n$. Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of the pencil $A_0 - \lambda B_0$, ordered so that $|p(\lambda_1)| \geq |p(\lambda_2)| \geq \cdots \geq |p(\lambda_n)|$. Suppose k is a positive integer less than n such that $|p(\lambda_k)| > |p(\lambda_{k+1})|$, and let $\rho = |p(\lambda_{k+1})|/|p(\lambda_k)|$. Let (p_i) be a sequence of polynomials of degree $\leq n$ such that $\lim_{i \to \infty} p_i = p$ and $p_i(\lambda_j) \neq 0$ for $j = 1, \ldots, k$ and all i. Let $(\mathcal{T}_d, \mathcal{T}_r)$ and $(\mathcal{U}_d, \mathcal{U}_r)$ be the deflating subspaces of $A_0 - \lambda B_0$ associated with $\lambda_1, \ldots, \lambda_k$ and $\lambda_{k+1}, \ldots, \lambda_n$, respectively. Suppose $\langle e_1, \ldots, e_k \rangle \cap \mathcal{U}_d = \{0\}$ and $\langle e_1, \ldots, e_k \rangle \cap \mathcal{U}_r = \{0\}$. Let $(A_i - \lambda B_i)$ be the sequence of iterates of the GZ algorithm using the given (p_i), starting from $A_0 - \lambda B_0$. If there is a constant $\hat{\kappa}$ such that the cumulative transforming matrices $\hat{G}_i$ and $\hat{Z}_i$ all satisfy $\kappa_2(\hat{G}_i) \leq \hat{\kappa}$ and $\kappa_2(\hat{Z}_i) \leq \hat{\kappa}$, then $(A_i - \lambda B_i)$ tends to block triangular form in the following sense. Let C_i denote either A_i or B_i, and partition C_i as*

$$C_i = \begin{bmatrix} C_{11}^{(i)} & C_{12}^{(i)} \\ C_{21}^{(i)} & C_{22}^{(i)} \end{bmatrix},$$

where $C_{21}^{(i)} \in \mathbb{C}^{(n-k) \times k}$. Then for every $\hat{\rho}$ satisfying $\rho < \hat{\rho} < 1$ there exists a constant M such that

$$\frac{\| C_{21}^{(i)} \|_2}{\| C_i \|_2} \leq M \hat{\rho}^i \quad \text{for all } i. \tag{2}$$

Remark 1. If A_0 is upper Hessenberg with no zeros on the subdiagonal and B_0 is upper triangular, then the subspace conditions $\langle e_1, \ldots, e_k \rangle \cap \mathcal{U}_d = \{0\}$ and $\langle e_1, \ldots, e_k \rangle \cap \mathcal{U}_r = \{0\}$ are satisfied for all k, as is explained in [17] for the standard eigenvalue problem. The reasoning is no different for the generalized eigenvalue problem.

Remark 2. The conditions $p_i(\lambda_j) \neq 0$ for $j = 1, \ldots, k$ may occasionally be violated, but this is not undesirable. If $p_i(\lambda_j) = 0$, then $p_i(A_i B_i^{-1})$ is singular. Theorem 4.3 shows that in this case the eigenvalue λ_j can be deflated from the problem after the ith iteration.

Remark 3. The conclusion of the theorem implies that the eigenvalues of $A_{11}^{(i)} - \lambda B_{11}^{(i)}$ and $A_{22}^{(i)} - \lambda B_{22}^{(i)}$ converge to $\lambda_1, \ldots, \lambda_k$ and $\lambda_{k+1}, \ldots, \lambda_n$, respectively, as can be shown by standard techniques.

Remark 4. If p has $\lambda_{k+1}, \ldots, \lambda_n$ among its roots, then $\rho = 0$, so (2) holds for all $\hat{\rho} > 0$. Thus the convergence is superlinear.

Remark 5. The hypotheses of the theorem usually hold for many values of k simultaneously, thereby giving a limiting form that is block triangular with many small blocks on the main diagonal. If the conditions hold for all k $(1 \le k \le n-1)$, the limiting form is upper triangular.

Proof. Let $\hat{p}_i = p_i \cdots p_1$, let $\mathcal{S} = \langle e_1, \ldots, e_k \rangle$, $\mathcal{S}_{ri} = \hat{p}_i(A_0 B_0^{-1})\mathcal{S}$, and $\mathcal{S}_{di} = \hat{p}_i(B_0^{-1}A_0)\mathcal{S}$. All of the hypotheses of Theorem 5.4 of [17] are satisfied, with the role of A in that theorem played by either $A_0 B_0^{-1}$ or $B_0^{-1}A_0$. Consequently there exists $\hat{M}$ such that $d(\mathcal{S}_{ri}, \mathcal{T}_r) \le \hat{M}\hat{\rho}^i$ and $d(\mathcal{S}_{di}, \mathcal{T}_d) \le \hat{M}\hat{\rho}^i$. Recall that (as shown in [17] and elsewhere) $\hat{p}_i(A_0 B_0^{-1}) = \hat{G}_i \hat{R}_i$ and $\hat{p}_i(B_0^{-1}A_0) = \hat{Z}_i \hat{S}_i$. Consider the partition $\hat{G}_i = [\hat{G}_1^{(i)} \ \hat{G}_2^{(i)}]$, $\hat{Z}_i = [\hat{Z}_1^{(i)} \ \hat{Z}_2^{(i)}]$, where $\hat{G}_1^{(i)}$, $\hat{Z}_1^{(i)} \in \mathbb{C}^{n \times k}$. Since $\hat{R}_i$ and $\hat{S}_i$ are upper triangular, $\mathcal{S}_{ri} = \mathcal{R}(\hat{G}_1^{(i)})$ and $\mathcal{S}_{di} = \mathcal{R}(\hat{Z}_1^{(i)})$. This is true even if $\hat{p}_i(A_0 B_0^{-1})$ and $\hat{p}_i(B_0^{-1}A_0)$ are singular, as the assumptions $\mathcal{S} \cap \mathcal{U}_d = \{0\}$, $\mathcal{S} \cap \mathcal{U}_r = \{0\}$, and $|p_i(\lambda_j)| > 0$, $j = 1, \ldots, k$ guarantee that $\mathcal{S}$ contains no nontrivial null vectors of $\hat{p}_i(A_0 B_0^{-1})$ or $\hat{p}_i(B_0^{-1}A_0)$. Therefore the spaces $\mathcal{S}_{di}$ and $\mathcal{S}_{ri}$ have dimension k for all i. Applying Lemma 3.1 with the roles of C, G, Z, and $\hat{C}$ played by C_0, $\hat{G}_i$, $\hat{Z}_i$, and C_i, respectively, we conclude that

$$\frac{\| C_{21}^{(i)} \|_2}{\| C_i \|_2} \le 2\sqrt{2}\kappa_2(\hat{G}_i)\kappa_2(\hat{Z}_i)\hat{M}\hat{\rho}^i \le M\hat{\rho}^i,$$

where $M = 2\sqrt{2}\hat{\kappa}^2\hat{M}$. □

3.1. The generalized Rayleigh quotient shift. Suppose we plan to perform GZ iterations of multiplicity m, where $m \ll n$. A natural way of choosing the shift polynomials is to let p_i be the characteristic polynomial of the $m \times m$ lower right-hand corner pencil $A_{22}^{(i)} - \lambda B_{22}^{(i)}$. We call this the *generalized Rayleigh quotient shift* strategy.

In [17] we proved that for the standard eigenvalue problem, the asymptotic convergence rate of the GR algorithm with the generalized Rayleigh quotient shift strategy is quadratic, provided that the eigenvalues of the given matrix are simple. This result also holds for the generalized eigenvalue problem. Specifically, if the GZ algorithm converges under the conditions of Theorem 3.2, and generalized Rayleigh quotient shifts with $m = n - k$ are used, the asymptotic convergence rate will be quadratic, provided $A_0 B_0^{-1}$ is simple. We omit the proof. The details are more tedious than they are for the standard problem, but the ideas are the same.

4. Implementation of GZ algorithms. We assume from now on that the initial transformation

$$A_0 = G_0^{-1} A Z_0, \qquad B_0 = G_0^{-1} B Z_0$$

makes A_0 upper Hessenberg and B_0 upper triangular. We even assume that A_0 is a *proper* upper Hessenberg matrix; that is, all of its subdiagonal entries $a_{i+1,i}^{(0)}$ are nonzero. This implies no loss of generality, for if some of the entries $a_{i+1,i}^{(0)}$ are zero, we can reduce the problem to two or more subproblems, each of which has a properly upper Hessenberg coefficient matrix. Since we are now concerned with the problem of implementing one iteration of the GZ algorithm, we drop the subscripts and consider the single iteration

$$\hat{A} = \hat{G}^{-1} A \hat{Z}, \qquad \hat{B} = \hat{G}^{-1} B \hat{Z}, \tag{3}$$

where

$$p(AB^{-1}) = \hat{G}\hat{R}, \qquad p(B^{-1}A) = \hat{Z}\hat{S}. \tag{4}$$

Here all matrices are in $\mathbb{C}^{n\times n}$. The degree of the polynomial p is m, which is assumed to be less than n. Normally $m \ll n$. Dropping the subscripts allows us to reintroduce subscripts later for a different purpose.

An important relationship that follows directly from (3) and (4) is given in the following lemma, which plays a key role in determining the structure of $\hat{A}$, $\hat{B}$ and intermediate matrices that arise during the execution of a GZ iteration. (It is a generalization of [10, (3.5)].) Although the lemma is used to study structure, it is not itself dependent on any special structure of the matrices involved, except that it is crucial that $\hat{G}$ and $\hat{Z}$ be nonsingular.

LEMMA 4.1. *Suppose A, $\hat{A}$, B, $\hat{B}$, $\hat{R}$, and $\hat{S}$ are any $n \times n$ matrices related by* (3) *and* (4), *where $\hat{G}$ and $\hat{Z}$ are nonsingular matrices, and p is a polynomial. Then*

$$\hat{A}\hat{S} = \hat{R}A \quad \textit{and} \quad \hat{B}\hat{S} = \hat{R}B.$$

Proof. $\hat{A}\hat{S} = \hat{A}\hat{Z}^{-1}p(B^{-1}A) = \hat{G}^{-1}Ap(B^{-1}A) = \hat{G}^{-1}p(AB^{-1})A = \hat{R}A$. The same argument shows that $\hat{B}\hat{S} = \hat{R}B$, since the equation $Bp(B^{-1}A) = p(AB^{-1})B$ also holds. □

As a first application of Lemma 4.1, consider a GZ iteration in which the matrices $p(AB^{-1})$ and $p(B^{-1}A)$ are nonsingular, as is usually the case. Then $\hat{R}$ and $\hat{S}$ are also nonsingular, so the equations in Lemma 4.1 can be rewritten in the form

$$\hat{A} = \hat{R}A\hat{S}^{-1}, \qquad \hat{B} = \hat{R}B\hat{S}^{-1}.$$

Since A is properly upper Hessenberg and B, $\hat{R}$, and $\hat{S}^{-1}$ are all upper triangular, we see immediately that $\hat{A}$ is properly upper Hessenberg and $\hat{B}$ is upper triangular. Thus the special form is preserved from one iteration to the next.

4.1. The singular case. When $p(AB^{-1})$ and $p(B^{-1}A)$ are singular, the upper Hessenberg-triangular form is not preserved, but something even better happens. A small subpencil at the lower right-hand corner of the matrix can be deflated from the pencil after the iteration. The part of the pencil that remains after deflation remains in Hessenberg-triangular form. We consider this case in detail.

The matrices AB^{-1} and $B^{-1}A$ are both properly upper Hessenberg. The proper upper Hessenberg matrix W that appears in the following lemma is taken to be AB^{-1} or $B^{-1}A$ in the application.

If $p(W)$ is singular, then at least one of the shifts (roots of p) is an eigenvalue of W, and conversely. Any shift that is an eigenvalue is called a *perfect* shift.

LEMMA 4.2. *Let W be a proper upper Hessenberg matrix, and let p be a polynomial that has ν roots that are perfect shifts for W. Then*

$$\operatorname{rank}(p(W)) = n - \nu.$$

Furthermore, the leading $n - \nu$ columns of $p(W)$ are linearly independent.

Remark. When we count perfect shifts, we allow repeated shifts, but we count a repeated shift no more times than it appears as a root of the characteristic polynomial of W.

This result is also proved in [10] as part of Theorem 4.1.

Proof. The statement about the rank is just Lemma 4.4 of [18]. To get the other assertion, let $x = p(W)e_1$. Let $K(W,x)$ denote the Krylov matrix

$$K(W,x) = [x, Wx, W^2x, \ldots, W^{n-1}x].$$

Then

$$K(W,x) = p(W)T, \tag{5}$$

where $T = K(W,e_1)$. Since W is properly upper Hessenberg, T is upper triangular and nonsingular. Thus, for any k, the span of the first k columns of $K(W,x)$ is the same as the span of the first k columns of $p(W)$. In particular, $\operatorname{rank}(K(W,x)) = n-\nu$. The form of a Krylov matrix implies that if a given column is a linear combination of previous columns, all subsequent columns will also be linear combinations of the previous columns. Thus the first $n-\nu$ columns of $K(W,x)$, and hence also of $p(W)$, must be linearly independent. □

THEOREM 4.3. *Consider the GZ iteration* (3), (4), *in which ν of the shifts are eigenvalues of $A - \lambda B$. Then*

$$\hat{R} = \begin{bmatrix} \hat{R}_{11} & \hat{R}_{12} \\ 0 & 0 \end{bmatrix}, \qquad \hat{S} = \begin{bmatrix} \hat{S}_{11} & \hat{S}_{12} \\ 0 & 0 \end{bmatrix},$$

$$\hat{A} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ 0 & \hat{A}_{22} \end{bmatrix}, \quad \textit{and} \quad \hat{B} = \begin{bmatrix} \hat{B}_{11} & \hat{B}_{12} \\ 0 & \hat{B}_{22} \end{bmatrix},$$

where $\hat{R}_{11}$, $\hat{S}_{11}$, $\hat{A}_{11}$, $\hat{B}_{11} \in \mathbb{C}^{(n-\nu)\times(n-\nu)}$, $\hat{R}_{11}$ and $\hat{S}_{11}$ are nonsingular, $\hat{A}_{11}$ is properly upper Hessenberg, and $\hat{B}_{11}$ is upper triangular. The eigenvalues of the subpencil $\hat{A}_{22} - \lambda\hat{B}_{22}$ are exactly the ν perfect shifts.

Proof. In light of (4) and Lemma 4.2, the upper triangular matrices $\hat{R}$ and $\hat{S}$ both have rank $n-\nu$ and their first $n-\nu$ columns are linearly independent. This proves that they have the stated form.

Writing the equations of Lemma 4.1 in partitioned form, we have

$$\begin{bmatrix} \hat{C}_{11} & \hat{C}_{12} \\ \hat{C}_{21} & \hat{C}_{22} \end{bmatrix} \begin{bmatrix} \hat{S}_{11} & \hat{S}_{12} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \hat{R}_{11} & \hat{R}_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},$$

where C can denote either A or B. Equating the $(2,1)$ blocks of the partitioned equation, we find that $\hat{C}_{21}\hat{S}_{11} = 0$. Since $\hat{S}_{11}$ is nonsingular, we have $\hat{C}_{21} = 0$. Equating the $(1,1)$ blocks and multiplying on the right by $\hat{S}_{11}^{-1}$, we get

$$\hat{C}_{11} = \hat{R}_{11}C_{11}\hat{S}_{11}^{-1} + \hat{R}_{12}C_{21}\hat{S}_{11}^{-1}.$$

In the case $C = B$, we have $B_{21} = 0$, so $\hat{B}_{11} = \hat{R}_{11}B_{11}\hat{S}_{11}^{-1}$, which shows that $\hat{B}_{11}$ is upper triangular. Now consider the case $C = A$. Since A is upper Hessenberg, $A_{21} = \alpha e_1 e_{n-\nu}^T$, where $\alpha = a_{n-\nu+1,n-\nu}$. Since $\hat{S}_{11}^{-1}$ is upper triangular, we have $\alpha e_{n-\nu}^T\hat{S}_{11}^{-1} = \beta e_{n-\nu}^T$ for some β. Let $x = \hat{R}_{12}e_1$. Then

$$\hat{A}_{11} = \hat{R}_{11}A_{11}\hat{S}_{11}^{-1} + \beta x e_{n-\nu}^T.$$

The first term on the right-hand side is properly upper Hessenberg and the second term has nonzero entries only in the last column. Thus $\hat{A}_{11}$ is a proper upper Hessenberg matrix.

The fact that the eigenvalues of $\hat{A}_{22} - \lambda\hat{B}_{22}$ are just the ν perfect shifts can be deduced in the same way as in the standard eigenvalue problem by considering the form of $\hat{A}\hat{B}^{-1}$. See Theorem 4.5 of [18]. □

Remarks. We have opted for a brief algebraic proof using Lemma 4.1. Alternatively one could prove the form of $\hat{A}$ and $\hat{B}$ geometrically, using the Hessenberg form of A, the relationships between the underlying subspaces (spanned by the leading columns of $\hat{G}$ and $\hat{Z}$), and the fact that the $n-\nu$ dimensional spaces $\mathcal{T}_d = \mathcal{R}(p(B^{-1}A))$ and $\mathcal{T}_r = \mathcal{R}(p(AB^{-1}))$ form a deflating pair for $A - \lambda B$. Such a proof would be lengthier but perhaps more revealing.

Theorem 4.3 generalizes Theorem 4.5 of [18] and some aspects of Theorem 4.1 of Miminis and Paige [10]. However, the Miminis–Paige result addresses certain details that we have chosen to ignore.

Theorem 4.3 shows that if ν of the shifts are perfect, then a $\nu \times \nu$ subpencil can be deflated from the problem after the iteration. The pencil $\hat{A}_{22} - \lambda\hat{B}_{22}$ may not have Hessenberg-triangular form, but it is normally small enough that it can easily be returned to that form and its eigenvalues found. Subsequent iterations can focus on the pencil $\hat{A}_{11} - \lambda\hat{B}_{11}$, which does have Hessenberg-triangular form. Of course this is only a theoretical result. In a GZ step with roundoff errors, $\hat{A}_{21}$ will be not quite zero. Usually it will be far enough from zero to prevent deflation. In that case a subsequent GZ step with the same p will often produce the deflation.

4.2. *GR* decompositions and elimination rules. To introduce specific versions of the GZ algorithm, we need to consider how GR decompositions are carried out in practice. The standard way to perform a GR decomposition of any type is to "reduce the matrix to triangular form" by introducing zeros into the matrix one column at a time. Each column of zeros is obtained by multiplying on the left by a nonsingular matrix of a specified form. Algorithms of this type have the following general structure. A matrix $M \in \mathbb{C}^{n\times n}$ is reduced to upper triangular form in $n-1$ steps. After $i-1$ steps, M has been transformed to a matrix $\hat{R}_{i-1}$ whose first $i-1$ columns have been reduced to upper triangular form. That is,

$$\hat{R}_{i-1} = \begin{bmatrix} T & E \\ 0 & F \end{bmatrix},$$

where $T \in \mathbb{C}^{i-1\times i-1}$ is upper triangular. The ith step transforms $\hat{R}_{i-1}$ to $\hat{R}_i = G_i^{-1}\hat{R}_{i-1}$, where G_i has the form

$$G_i = \begin{bmatrix} I & 0 \\ 0 & \tilde{G}_i \end{bmatrix},$$

and $\tilde{G}_i \in \mathbb{C}^{(n-i+1)\times(n-i+1)}$ is chosen so that $\tilde{G}_i^{-1}x = \alpha e_1$, where x is the first column of F, and α is a scalar. After $n-1$ such steps, M will have been transformed to the upper triangular matrix $\hat{R} = \hat{R}_{n-1}$. Clearly $\hat{R} = G_{n-1}^{-1}\cdots G_1^{-1}M$, or

$$M = \hat{G}\hat{R},$$

where $\hat{G} = G_1 \cdots G_{n-1}$.

Given any vector $x \in \mathbb{C}^m$ (for any $m \geq 2$) we say that a matrix $\tilde{G} \in \mathbb{C}^{m \times m}$ is an *elimination matrix* for x if $\tilde{G}$ is nonsingular and $\tilde{G}^{-1}x = \alpha e_1$ for some scalar α. If $\tilde{G}$ is an elimination matrix for x, then $\tilde{G}$ is an elimination matrix for all nonzero multiples of x.

Usually an elimination matrix $\tilde{G}$ is embedded in a larger matrix G. We also refer to the larger matrix as an elimination matrix.

An *elimination rule* is a map $x \mapsto \tilde{G}$ having the following properties. (i) The domain of the map is a subset of $\bigcup_{i=2}^{\infty} \mathbb{C}^i$. (ii) Each vector x in the domain is mapped to a matrix $\tilde{G}$ that is an elimination matrix for x. (iii) The map is homogeneous, that is, if x is in the domain, so are all nonzero multiples of x, and they are all mapped to the same elimination matrix. (iv) Zero vectors are in the domain, and each is mapped to the identity matrix of the same size. (v) If $x = \begin{bmatrix} y \\ 0 \end{bmatrix} \in \mathbb{C}^n$, where $y \in \mathbb{C}^k$ with $k < n$, then x is in the domain if and only if y is, and x is mapped to the matrix $\tilde{G} = \operatorname{diag}\{\tilde{H}, I_{n-k}\}$, where $\tilde{H}$ is the elimination matrix assigned to y by the map.

A *complete* elimination rule is one whose domain is all of $\bigcup_{i=2}^{\infty} \mathbb{C}^i$. A *partial* elimination rule is one whose domain is a proper subset of $\bigcup_{i=2}^{\infty} \mathbb{C}^i$.

Probably the simplest elimination rule is *Gaussian elimination without pivoting.* It is a partial elimination rule, as it is undefined on those nonzero x that satisfy $x_1 = 0$. An example of a complete elimination rule is Gaussian elimination with pivoting, which interchanges x_1 with the entry in x of largest magnitude before performing the elimination. Another complete elimination rule is *elimination by reflector* (*Householder transformation*). All of these types of elimination are discussed in [6] and [15], for example. One can also build hybrid elimination rules from other rules. For example, one can pick a tolerance τ satisfying $0 < \tau < 1$ and specify that x should be eliminated by Gaussian elimination (without pivoting) if $\max_{2 \leq i \leq k} |x_i| \leq \tau |x_1|$ and by a reflector otherwise. This type of strategy has been used successfully in some of the algorithms in [7]. There are also more exotic types of elimination rules. For example, a symplectic (partial) elimination rule, which gives rise to the SR algorithm, is described in [1].

Every elimination rule induces a rule for carrying out GR decompositions; namely, carry out the "reduction to triangular form" described above using the specified elimination rule. Hence each elimination rule, together with a mechanism for choosing p, induces a GZ algorithm. If the elimination rule is not complete, the algorithm will break down (fail) if at some point it needs to perform an elimination on a vector that is not in the domain of the rule.

4.3. The explicit GZ algorithm. We now assume that we have chosen an elimination rule and will perform all of our eliminations with that rule.[2] Let us examine closely the steps involved in performing a GZ iteration explicitly. First the matrices $p(AB^{-1})$ and $p(B^{-1}A)$ are calculated. Then GR decompositions of both matrices are performed, using our chosen elimination rule. As above, we assume that ν of the shifts are perfect. If $\nu > 0$, then by Theorem 4.3, the resulting upper triangular matrices $\hat{R}$ and $\hat{S}$ have rank $n - \nu$, and their bottom ν rows zero. This

[2] However, everything we do could be cast in greater generality. For example, we could allow a different rule to be used at each step of the decomposition. Another possibility is to prescribe different rules for the two different GR decompositions on which the GZ iteration is based. Such an algorithm was once proposed by Kaufman [9]. In this algorithm all of the "G" transformations are unitary and all of the "Z" transformations are stabilized elementary (i.e., Gaussian elimination) transformations. This is a GZ algorithm in which the decomposition of $p(AB^{-1})$ is a QR decomposition and that of $p(B^{-1}A)$ is an LR decomposition with partial pivoting.

implies that the reductions to triangular form will be completed after $n - \nu$ steps. Thus, letting $\rho = \min\{n - \nu, n - 1\}$, the reductions have the form

$$\hat{R} = G_\rho^{-1} \cdots G_2^{-1} G_1^{-1} p(AB^{-1}),$$

$$\hat{S} = Z_\rho^{-1} \cdots Z_2^{-1} Z_1^{-1} p(B^{-1}A).$$

Therefore

$$p(AB^{-1}) = \hat{G}\hat{R} \quad \text{and} \quad p(B^{-1}A) = \hat{Z}\hat{S},$$

where

$$\hat{G} = G_1 \cdots G_\rho \quad \text{and} \quad \hat{Z} = Z_1 \cdots Z_\rho.$$

We then complete the iteration by performing the equivalence transformations

$$\hat{A} = \hat{G}^{-1} A \hat{Z} \quad \text{and} \quad \hat{B} = \hat{G}^{-1} B \hat{Z}. \tag{6}$$

Remark. In the case of the standard eigenvalue problem ($B = I$), we have $p(AB^{-1}) = p(B^{-1}A) = p(A)$, so $G_i = Z_i$, $i = 1, \ldots, \rho$, $\hat{G} = \hat{Z}$, $\hat{R} = \hat{S}$, $\hat{A} = \hat{G}^{-1} A \hat{G}$, and $\hat{B} = I$. This is one iteration of a GR algorithm [17].

To determine how to do these operations implicitly, we break the transformations (6) down into small steps and study the intermediate results. Let C denote A or B, as before; define $\hat{C}_0 = C$, and

$$\left.\begin{aligned} \hat{C}_{i-1/2} &= G_i^{-1} \hat{C}_{i-1} \\ \hat{C}_i &= \hat{C}_{i-1/2} Z_i \end{aligned}\right\} \quad i = 1, \ldots, \rho. \tag{7}$$

Then $\hat{C} = \hat{C}_\rho$.

We also give names to the intermediate matrices in the GR decompositions. Let $\hat{R}_0 = p(AB^{-1})$, $\hat{S}_0 = p(B^{-1}A)$, and

$$\left.\begin{aligned} \hat{R}_i &= G_i^{-1} \hat{R}_{i-1} \\ \hat{S}_i &= Z_i^{-1} \hat{S}_{i-1} \end{aligned}\right\} \quad i = 1, \ldots, \rho.$$

Then $\hat{R} = \hat{R}_\rho$ and $\hat{S} = \hat{S}_\rho$.

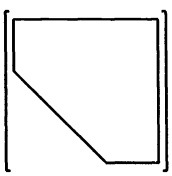
Since AB^{-1} and $B^{-1}A$ are proper upper Hessenberg matrices, $\hat{R}_0$ and $\hat{S}_0$ satisfy

$$\hat{r}^{(0)}_{j+m,j} \neq 0, \quad \hat{s}^{(0)}_{j+m,j} \neq 0, \quad j = 1, \ldots, n - m$$

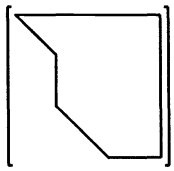
and

$$\hat{r}^{(0)}_{ij} = \hat{s}^{(0)}_{ij} = 0 \quad \text{when } i > j + m,$$

where m is the degree of p. Thus they have the form



where the entries outside of the outlined area are all zero. Since G_1 is the elimination matrix for the first column of $\hat{R}_0$, $G_1 = \text{diag}\{\tilde{G}_1, I\}$, where $\tilde{G}_1 \in \mathbb{C}^{(m+1)\times(m+1)}$ is the elimination matrix of $[\hat{r}_{11}^{(0)}, \ldots, \hat{r}_{m+1,1}^{(0)}]^T$. The form of Z_1 is similar. In general $\hat{R}_{i-1}$ and $\hat{S}_{i-1}$ have the form



for $i = 2, \ldots, n-m$. The first $i-1$ columns are in upper triangular form. The ith column has m nonzeros below the main diagonal. Thus $G_i = \text{diag}\{I_{i-1}, \tilde{G}_i, I\}$ and $Z_i = \text{diag}\{I_{i-1}, \tilde{Z}_i, I\}$, where $\tilde{G}_i$ and $\tilde{Z}_i$ are the elimination matrices of

$$[\hat{r}_{ii}^{(i-1)}, \ldots, \hat{r}_{m+i,i}^{(i-1)}]^T \quad \text{and} \quad [\hat{s}_{ii}^{(i-1)}, \ldots, \hat{s}_{m+i,i}^{(i-1)}]^T,$$

respectively. For $i > n-m$ the transformations have the same form, except that the vectors to be eliminated are shorter because we have reached the bottom of the matrix. We then have $G_i = \text{diag}\{I_{i-1}, \tilde{G}_i\}$, where $\tilde{G}_i$ is the elimination matrix of $[\hat{r}_{ii}^{(i-1)}, \ldots, \hat{r}_{n,i}^{(i-1)}]^T$, and similarly for Z_i.

Let $\hat{G}_i = G_1 \cdots G_i$ and $\hat{Z}_i = Z_1 \cdots Z_i$, $i = 1, \ldots, \rho$. Then for $i < n-m$, $\hat{G}_i$ has the form

$$\hat{G}_i = \begin{bmatrix} \hat{G}_{11}^{(i)} & 0 \\ 0 & I \end{bmatrix},$$

where $\hat{G}_{11}^{(i)}$ is $(m+i) \times (m+i)$. This is clear from the form of the factors. The form of $\hat{Z}_i$ is the same.

The initial pencil $A - \lambda B = \hat{A}_0 - \lambda \hat{B}_0$ is in Hessenberg-triangular form, and so is the final pencil $\hat{A} - \lambda \hat{B} = \hat{A}_\rho - \lambda \hat{B}_\rho$, except possibly for a small subpencil that can be removed by deflation. The intermediate pencils $\hat{A}_{i-1/2} - \lambda \hat{B}_{i-1/2}$ and $\hat{A}_i - \lambda \hat{B}_i$ are not in Hessenberg-triangular form, but, as we shall see, they do not deviate from it by too much. First note that

$$\hat{A}_i = \hat{G}_i^{-1} A \hat{Z}_i \quad \text{and} \quad \hat{B}_i = \hat{G}_i^{-1} B \hat{Z}_i, \tag{8}$$

for $i = 1, \ldots, \rho$. Since also

$$p(AB^{-1}) = \hat{G}_i \hat{R}_i \quad \text{and} \quad p(B^{-1}A) = \hat{Z}_i \hat{S}_i, \tag{9}$$

we see that the pencil $\hat{A}_i - \lambda \hat{B}_i$ is the result of a partial GZ iteration driven by the partial GR decompositions (9). Applying Lemma 4.1 to (8) and (9), we find that also

$$\hat{A}_i \hat{S}_i = \hat{R}_i A \quad \text{and} \quad \hat{B}_i \hat{S}_i = \hat{R}_i B. \tag{10}$$

We use these two equations in Lemmas 4.8 and 4.5, respectively, to help determine the shape of $\hat{A}_i$ and $\hat{B}_i$.

Similarly, we have

$$\hat{A}_{i-1/2} = \hat{G}_i^{-1} A \hat{Z}_{i-1} \quad \text{and} \quad \hat{B}_{i-1/2} = \hat{G}_i^{-1} B \hat{Z}_{i-1}, \tag{11}$$

so the pencil $\hat{A}_{i-1/2} - \lambda\hat{B}_{i-1/2}$ can be viewed as the result of a partial GZ iteration driven by the partial GR decompositions

$$p(AB^{-1}) = \hat{G}_i\hat{R}_i \quad \text{and} \quad p(B^{-1}A) = \hat{Z}_{i-1}\hat{S}_{i-1}. \tag{12}$$

Applying Lemma 4.1 to (11) and (12), we obtain

$$\hat{A}_{i-1/2}\hat{S}_{i-1} = \hat{R}_iA \quad \text{and} \quad \hat{B}_{i-1/2}\hat{S}_{i-1} = \hat{R}_iB. \tag{13}$$

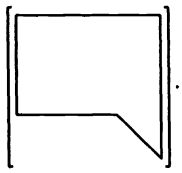
These equations are used to help to determine the shape of $\hat{A}_{i-1/2}$ and $\hat{B}_{i-1/2}$.

We study first the shape of the "B" matrices.

LEMMA 4.4. *For $i = 1, \ldots, n-m-1$, the last $n-m-i$ rows of $\hat{B}_{i-1/2}$ and $\hat{B}_i$ are in upper triangular form. That is,*

$$\hat{b}_{jk}^{(i-1/2)} = \hat{b}_{jk}^{(i)} = 0 \quad \text{if } j > k \quad \text{and} \quad j > i+m.$$

Pictorially, $\hat{B}_{i-1/2}$ and $\hat{B}_i$ have the form



Proof. Writing the transformation $\hat{B}_i = \hat{G}_i{}^{-1}B\hat{Z}_i$ in partitioned form, we have

$$\begin{bmatrix} \hat{B}_{11}^{(i)} & \hat{B}_{12}^{(i)} \\ \hat{B}_{21}^{(i)} & \hat{B}_{22}^{(i)} \end{bmatrix} = \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix} \begin{bmatrix} Y & 0 \\ 0 & I \end{bmatrix},$$

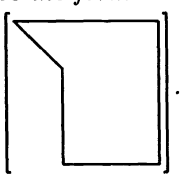
where the $(1,1)$ blocks all have dimension $(m+i) \times (m+i)$. Clearly $\hat{B}_{21}^{(i)} = 0$, and $\hat{B}_{22}^{(i)} = B_{22}$, which is upper triangular. Thus $\hat{B}_i$ has the stated form. To prove that $\hat{B}_{i-1/2}$ also has this form, apply the same partition to the equation $\hat{B}_{i-1/2} = \hat{G}_i^{-1}B\hat{Z}_{i-1}$. □

Lemma 4.4 suggests that $\hat{B}_{n-m-1/2}$ and $\hat{B}_{n-m}$ should be completely filled in. Fortunately the transformations do not only destroy zeros, they create zeros as well, as we see in Lemma 4.5. For the purpose of avoiding distracting complications in the statement of this lemma, we define $\hat{B}_{\rho+1/2} = \hat{B}$.

LEMMA 4.5. *For $i = 1, \ldots, \rho$, the first i columns of $\hat{B}_i$ and $\hat{B}_{i+1/2}$ are in upper triangular form. That is,*

$$\hat{b}_{jk}^{(i)} = \hat{b}_{jk}^{(i+1/2)} = 0 \quad \text{if } j > k \quad \text{and} \quad k \leq i.$$

Pictorially, both $\hat{B}_i$ and $\hat{B}_{i+1/2}$ have the form



Proof. By (10) $\hat{B}_i \hat{S}_i = \hat{R}_i B$. We write this equation in partitioned form as

$$\begin{bmatrix} \hat{B}_{11}^{(i)} & \hat{B}_{12}^{(i)} \\ \hat{B}_{21}^{(i)} & \hat{B}_{22}^{(i)} \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix},$$

where the $(1,1)$ blocks are all $i \times i$. The matrices S_{11}, R_{11}, and B_{11} are upper triangular. We know that the first $n-\nu$ columns of $\hat{S}_i$ are linearly independent, and since $i \leq \rho \leq n-\nu$, S_{11} must be nonsingular. Therefore

$$\begin{bmatrix} \hat{B}_{11}^{(i)} \\ \hat{B}_{21}^{(i)} \end{bmatrix} = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix} B_{11} S_{11}^{-1}.$$

Consequently, $\hat{B}_{11}^{(i)} = R_{11} B_{11} S_{11}^{-1}$ is upper triangular and $\hat{B}_{21}^{(i)} = 0$. This proves that $\hat{B}_i$ has the stated form. To obtain the same result for $\hat{B}_{i+1/2}$, partition the equation $\hat{B}_{i+1/2} \hat{S}_i = \hat{R}_{i+1} B$ (from (13)) exactly as above. □

Remark. In the nonsingular case, or even in the case $\nu = 1$, Lemma 4.5 shows that $\hat{B} = \hat{B}_{n-1}$ is upper triangular. If $\nu \geq 2$, we can conclude that $\hat{B} = \hat{B}_\rho$ has only its first $\rho = n-\nu$ columns upper triangular; it is not guaranteed that the final ν columns get reduced to upper triangular form. But we already know from Theorem 4.3 that this portion of the matrix will be removed by deflation at the end of the iteration.

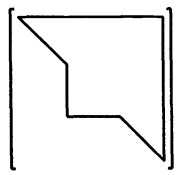
Combining Lemmas 4.4 and 4.5, we have the following result.

THEOREM 4.6. *For* $i = 1, \ldots, \rho$,

$$\hat{b}_{jk}^{(i-1/2)} = 0 \quad \text{if } j > k \text{ and either } j > i+m \text{ or } k \leq i-1,$$

$$\hat{b}_{jk}^{(i)} = 0 \quad \text{if } j > k \text{ and either } j > i+m \text{ or } k \leq i.$$

Thus we see that when $i < n-m$, $\hat{B}_{i-1/2}$ is upper triangular in its first $i-1$ columns and its last $n-i-m$ rows. Its nonzero pattern is



which would be upper triangular, except that it has a bulge. The tip of the bulge is at the $(i+m, i)$ position. We call this an *m-bulge* or a *bulge of order m* because it protrudes m diagonals below the upper triangular part of the matrix. The form of $\hat{B}_i$ is similar, but the tip of its bulge is at the $(i+m, i+1)$ position. This is a bulge of order $m-1$. We see thus that the transformation $\hat{B}_{i-1/2} \to \hat{B}_i$ shrinks the bulge by deleting one column from the left side. On the other hand, the transformation $\hat{B}_i \to \hat{B}_{i+1/2}$ enlarges the bulge by adding one row to the bottom. Thus the bulge is chased downward and to the right as the GZ iteration proceeds. When $i = n-m$, the bulge has reached the bottom of the matrix and begins to be pushed off the edge. If $\nu \leq 1$, the bulge is eventually eliminated completely. If $\nu > 1$, the iteration ends with the last ν columns uncleared.

We now turn our attention to the "A" matrices.

LEMMA 4.7. *For $i = 0, \ldots, n-m-2$, the last $n-i-m-1$ rows of $\hat{A}_i$ and $\hat{A}_{i+1/2}$ have upper Hessenberg form. That is,*

$$\hat{a}_{jk}^{(i)} = \hat{a}_{jk}^{(i+1/2)} = 0 \quad \text{if } j > k+1 \quad \text{and} \quad j > i+m+1.$$

Proof. Write the transformation $\hat{A}_{i+1/2} = \hat{G}_{i+1}^{-1} A \hat{Z}_i$ in the unsymmetric partitioned form

$$\begin{bmatrix} \hat{A}_{11}^{(i+1/2)} & \hat{A}_{12}^{(i+1/2)} \\ \hat{A}_{21}^{(i+1/2)} & \hat{A}_{22}^{(i+1/2)} \end{bmatrix} = \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} Y & 0 \\ 0 & I \end{bmatrix},$$

where $\hat{A}_{11}^{(i+1/2)}$, $A_{11} \in \mathbb{C}^{(i+m+1)\times(i+m)}$, $X \in \mathbb{C}^{(i+m+1)\times(i+m+1)}$, $Y \in \mathbb{C}^{(i+m)\times(i+m)}$. Clearly $\hat{A}_{21}^{(i+1/2)} = 0$, and $\hat{A}_{22}^{(i+1/2)} = A_{22}$, which is the bottom right-hand corner of an upper Hessenberg matrix. Thus $\hat{A}_{i+1/2}$ has the stated form. To prove that $\hat{A}_i$ also has this form, apply the same partition to the equation $\hat{A}_i = \hat{G}_i^{-1} A \hat{Z}_i$. □

Thus it appears that $\hat{A}_{n-m-1}$ is completely filled in. But again it turns out that the transformations are not only destroying zeros, they are creating zeros as well.

LEMMA 4.8. *For $2 \le i \le \rho$, the first $i-1$ columns of $\hat{A}_{i-1/2}$ and $\hat{A}_i$ are in upper Hessenberg form. That is,*

$$\hat{a}_{jk}^{(i-1/2)} = \hat{a}_{jk}^{(i)} = 0 \quad \text{if } j > k+1 \quad \text{and} \quad k \le i-1.$$

Proof. By (13) we know that $\hat{A}_{i-1/2}\hat{S}_{i-1} = \hat{R}_i A$. Consider the unsymmetric partition

$$\begin{bmatrix} \hat{A}_{11}^{(i-1/2)} & \hat{A}_{12}^{(i-1/2)} \\ \hat{A}_{21}^{(i-1/2)} & \hat{A}_{22}^{(i-1/2)} \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

where $\hat{A}_{11}^{(i-1/2)}$, $A_{11} \in \mathbb{C}^{i\times(i-1)}$, $S_{11} \in \mathbb{C}^{(i-1)\times(i-1)}$, and $R_{11} \in \mathbb{C}^{i\times i}$. Both S_{11} and R_{11} are upper triangular and nonsingular. Thus

$$\begin{bmatrix} \hat{A}_{11}^{(i-1/2)} \\ \hat{A}_{21}^{(i-1/2)} \end{bmatrix} = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix} A_{11} S_{11}^{-1}.$$

Therefore $\hat{A}_{21}^{(i-1/2)} = 0$ and $\hat{A}_{11}^{(i-1/2)} = R_{11} A_{11} S_{11}^{-1}$. Since R_{11} and S_{11}^{-1} are upper triangular, and A_{11} satisfies $a_{jk} = 0$ if $j > k+1$, $\hat{A}_{11}^{(i-1/2)}$ must also have this zero pattern. This proves that $\hat{A}_{i-1/2}$ has the stated form. We obtain the same result for $\hat{A}_i$ by partitioning the equation $\hat{A}_i \hat{S}_i = \hat{R}_i A$ in exactly the same way. □

Remark. As long as $\nu \le 1$, Lemma 4.8 shows that both $\hat{A}_{\rho-1/2}$ and $\hat{A}_\rho$ are upper Hessenberg. If $\nu > 1$, Lemma 4.8 states that the first $\rho - 1$ columns of $\hat{A} = \hat{A}_\rho$ are upper Hessenberg. In fact the situation is better than that. From Theorem 4.3 we know that $\hat{A}$ is block triangular; all of the entries in column ρ below the main diagonal are automatically zero. Of course this is a theoretical result that is valid only in the absence of roundoff error.

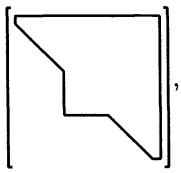
Combining the results of Lemmas 4.7 and 4.8, we have the following theorem.

THEOREM 4.9. *For* $1 \le i \le \rho$,

$$\hat{a}_{jk}^{(i-1/2)} = 0 \quad \text{if } j > k+1 \quad \text{and either } j > i+m \text{ or } k \le i-1,$$

$$\hat{a}_{jk}^{(i)} = 0 \quad \text{if } j > k+1 \text{ and either } j > i+m+1 \text{ or } k \le i-1.$$

For $i = 1, \ldots, n-m-1$, $\hat{A}_{i-1/2}$ has its first $i-1$ columns and its last $n-m-i$ rows in upper Hessenberg form. Thus it has the form



It has a bulge that has its tip at position $(i+m, i)$. We call this a bulge of *order* $m-1$ because it protrudes $m-1$ diagonals below the Hessenberg part of the matrix. $\hat{A}_i$ has its first $i-1$ columns and last $n-m-i-1$ rows upper Hessenberg. That is, it has a bulge whose tip is at $(i+m+1, i)$. This is an m-bulge.

The transformation $\hat{A}_{i-1/2} \to \hat{A}_i$ enlarges the bulge by adding one row to the bottom. On the other hand, the transformation $\hat{A}_i \to \hat{A}_{i+1/2}$ shrinks the bulge by removing one column from the left side. Thus the bulge in $\hat{A}_i$ moves downward and to the right, just as it does in $\hat{B}_i$. When $i = n-m-1$, the bulge has reached the bottom of the matrix and begins to be pushed off the edge.

Now let us consider the effects of the transformations on the A and B matrices together. The transformation $\hat{C}_{i-1/2} \to \hat{C}_i$ enlarges the "A"-bulge while shrinking the "B"-bulge. On the other hand, the transformation $\hat{C}_i \to \hat{C}_{i+1/2}$ shrinks the "A"-bulge while expanding the "B"-bulge. The relative positions of the bulges in $\hat{A}_i$ and $\hat{B}_i$ can be seen by superimposing them on one array.

$$\begin{bmatrix}
b & b & b & b & b & b & b & b \\
a & b & & & & & & b \\
 & a & b & & & & & b \\
 & a & b & & & & & b \\
 & a & b & b & b & & & b \\
 & a & a & a & a & b & & b \\
 & & & & & a & b & b \\
 & & & & & & a & b
\end{bmatrix}.$$

Here we have pictured the case $n = 8$, $m = 3$, $i = 2$. The nonzero part of $\hat{B}_2$ is the area outlined by the letter b. The nonzero part of $\hat{A}_2$ includes the nonzero part of $\hat{B}_2$ and in addition the entries marked with the letter a. This is an m-bulge for $\hat{A}_2$ and an $(m-1)$-bulge for $\hat{B}_2$. After the transformation $\hat{C}_2 \to \hat{C}_{5/2}$, the situation is

as follows.

$$\begin{bmatrix} b & b & b & b & b & b & b & b \\ a & b & & & & & & b \\ & a & b & & & & & b \\ & 0 & b & & & & & b \\ & 0 & b & b & b & & & b \\ & 0 & b & b & b & b & & b \\ & & & & & a & b & b \\ & & & & & & a & b \end{bmatrix}.$$

The "A"-bulge has been shrunk by the elimination of one column from the left (marked by zeros), and the "B"-bulge has been enlarged by the addition of one row at the bottom. Now the bulges coincide perfectly. This is an $m-1$ bulge in $\hat{A}_{5/2}$ and an m bulge in $\hat{B}_{5/2}$.

4.4. The implicit GZ algorithm. Now we are ready to use the information amassed in the previous section to see how to carry out an iteration of the GZ algorithm without actually forming the matrices $\hat{R}_0 = p(AB^{-1})$, $\hat{S}_0 = p(B^{-1}A)$, or any of the derived matrices $\hat{R}_i$, $\hat{S}_i$.

The first step is to find G_1, which is the elimination matrix for the first column of $\hat{R}_0 = p(AB^{-1})$. Thus we need to find $x = p(AB^{-1})e_1$. This can be computed relatively inexpensively if $m \ll n$. Indeed, p is given in the factored form $p(\lambda) = (\lambda - \sigma_1)\cdots(\lambda - \sigma_m)$. Thus x can be calculated by the recursion

$$x^{(j)} = (AB^{-1} - \sigma_j I)x^{(j-1)}, \qquad j = 1, \ldots, m, \tag{14}$$

with $x^{(0)} = e_1$. Then $x = x^{(m)}$. Since it does not matter whether we get x or a multiple of x, in practice we would also rescale at each step to avoid over/underflow. This is inexpensive. Since AB^{-1} is upper Hessenberg, only the first j components of $x^{(j-1)}$ are nonzero. Now consider the jth step. If we let $y^{(j)} = AB^{-1}x^{(j-1)}$, then $x^{(j)} = y^{(j)} - \sigma_j x^{(j-1)}$. We can find $y^{(j)}$ by solving $Bz^{(j)} = x^{(j-1)}$ for $z^{(j)}$ and then calculating $y^{(j)} = Az^{(j)}$. Because only the first j components of $x^{(j-1)}$ are nonzero and B is upper triangular, the system $Bz^{(j)} = x^{(j-1)}$ is in fact only a $j \times j$ upper triangular system, whose solution requires only $O(j^2)$ operations. Only the first j entries of $z^{(j)}$ are nonzero. Thus the product $Az^{(j)}$ involves only the first j columns of A. The nonzero entries in these columns are confined to the first $j+1$ rows, so $y^{(j)}$ can be obtained in $O(j^2)$ operations. The computation $x^{(j)} = y^{(j)} - \sigma_j x^{(j-1)}$ requires only $O(j)$ operations. Thus the total operation count for the jth step is $O(j^2)$. This must be done for $j = 1, \ldots, m$, so the total cost of computing x is $O(m^3)$, which is small if $m \ll n$.[3]

Once we have x, we can let $G_1 = \operatorname{diag}\{\tilde{G}_1, I\}$ be the elimination matrix for x given by the chosen elimination rule and calculate

$$\hat{A}_{1/2} = G_1^{-1}A \quad \text{and} \quad \hat{B}_{1/2} = G_1^{-1}B.$$

Logically the next question would be how to find Z_1. However, we postpone that and ask instead how one finds G_i, $i = 2, 3, \ldots, \rho$, to make the transformations

$$\hat{A}_{i-1/2} = G_i^{-1}\hat{A}_{i-1} \quad \text{and} \quad \hat{B}_{i-1/2} = G_i^{-1}\hat{B}_{i-1},$$

[3] This procedure can be modified in various ways. For example, a more accurate formula for the case $m = 2$ is given in [11]. Also, although we are assuming throughout this paper that B is nonsingular, one might reasonably ask whether this process can be salvaged in case one of $b_{11}, \ldots, b_{mm}$ happens to be zero. Deflation strategies for singular B are discussed in [11], [9], and [6, p. 400].

(see (7)) given that $\hat{A}_{i-1}$ and $\hat{B}_{i-1}$ are available. To deal with the two cases $i \leq n-m$ and $i > n-m$ simultaneously, let $k = \min\{i+m, n\}$. Then G_i has the form $\operatorname{diag}\{I_{i-1}, \tilde{G}_i, I_{n-k}\}$, where $\tilde{G}_i \in \mathbb{C}^{(k-i+1)\times(k-i+1)}$ is the elimination matrix for $[\hat{r}_{i,i}^{(i-1)}, \ldots, \hat{r}_{k,i}^{(i-1)}]^T$. Call this (column) vector y. We need to find y or a multiple of y and we need to do it without knowledge of $\hat{R}_{i-1}$. We know that $\tilde{G}^{-1}y = \alpha e_1$ for some scalar α. The fact that the first ρ columns of $\hat{R}_i$ are linearly independent guarantees that $\alpha \neq 0$.

We know from Theorem 4.9 that the operation

$$G_i^{-1}\hat{A}_{i-1} = \hat{A}_{i-1/2}$$

shrinks the bulge in $\hat{A}_{i-1}$ by removing column $i-1$ from the bulge. All this means is that the entries in positions $(i+1, i-1), \ldots, (k, i-1)$ get set to zero. Let us focus on this column. In transforming $\hat{A}_{i-1}$ to $\hat{A}_{i-1/2}$, the submatrix $\tilde{G}_i^{-1}$ acts only on rows i through k. As far as column $i-1$ is concerned, the action is

$$\tilde{G}_i^{-1}\begin{bmatrix} \hat{a}_{i,i-1}^{(i-1)} \\ \hat{a}_{i+1,i-1}^{(i-1)} \\ \vdots \\ \hat{a}_{k,i-1}^{(i-1)} \end{bmatrix} = \begin{bmatrix} \hat{a}_{i,i-1}^{(i-1/2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

In other words, defining $x \in \mathbb{C}^{k-i+1}$ by $x = [\hat{a}_{i,i-1}^{(i-1)}, \hat{a}_{i+1,i-1}^{(i-1)}, \ldots, \hat{a}_{k,i-1}^{(i-1)}]^T$, we have $\tilde{G}_i^{-1}x = \beta e_1$, where $\beta = \hat{a}_{i,i-1}^{(i-1/2)}$. One easily checks that $\beta \neq 0$. Indeed, subsequent transformations do not alter the entry in position $(i, i-1)$. Thus $\beta = \hat{a}_{i,i-1}$, and this is nonzero by Theorem 4.3. The relationship $\tilde{G}_i^{-1}x = \beta e_1$ says that $\tilde{G}_i$ is an elimination matrix for x. Since it is also an elimination matrix for y, x must be a multiple of y. Indeed, $x = \beta\tilde{G}_i e_1 = \beta\alpha^{-1}y$.

This relationship can also be inferred directly from the equation $\hat{A}_{i-1}\hat{S}_{i-1} = \hat{R}_{i-1}A$ by comparing column $i-1$ of the left-hand side product with the same column of the right-hand side, which is analogous to the approach used in [10]. Focusing on rows i through k, and taking into account the zero structure of the various matrices, we find that

$$\begin{bmatrix} \hat{a}_{i,i-1}^{(i-1)} \\ \vdots \\ \hat{a}_{k,i-1}^{(i-1)} \end{bmatrix} \hat{s}_{i-1,i-1}^{(i-1)} = \begin{bmatrix} \hat{r}_{i,i}^{(i-1)} \\ \vdots \\ \hat{r}_{k,i}^{(i-1)} \end{bmatrix} a_{i,i-1}. \tag{15}$$

We know that $a_{i,i-1} \neq 0$ for all i, and $\hat{s}_{i-1,i-1}^{(i-1)} \neq 0$ as long as $i \leq \rho + 1$. Thus x is a multiple of y.

This solves the problem of how to find G_i. We see that we do not need $\hat{R}_{i-1}$, as the required information for building G_i is also present in $\hat{A}_{i-1}$. We summarize our findings as a theorem.

THEOREM 4.10. *Let $2 \leq i \leq \rho$, and let $k = \min\{i+m, n\}$. Then $G_i = \operatorname{diag}\{I_{i-1}, \tilde{G}_i, I_{n-k}\}$, where $\tilde{G}_i$ is the elimination matrix of*

$$[\hat{a}_{i,i-1}^{(i-1)}, \hat{a}_{i+1,i-1}^{(i-1)}, \ldots, \hat{a}_{k,i-1}^{(i-1)}]^T.$$

We now turn to the question of how to compute the transformations Z_i. Given $1 \leq i \leq \rho$, let $k = \max\{i+m, n\}$. We know that $Z_i = \operatorname{diag}\{I_{i-1}, \tilde{Z}_i, I_{n-k}\}$, where $\tilde{Z}_i$

is the elimination matrix for the vector

$$x = [\hat{s}_{i,i}^{(i-1)}, \hat{s}_{i+1,i}^{(i-1)}, \ldots, \hat{s}_{k,i}^{(i-1)}]^T,$$

which is part of the ith column of $\hat{S}_i$. We wish to determine this vector (or a multiple thereof) without computing $\hat{S}_i$. We have $\tilde{Z}_i^{-1}x = \alpha e_1$, where $\alpha \neq 0$.

We know from Theorem 4.6 that the transformation $\hat{B}_{i-1/2}Z_i = \hat{B}_i$ (from (7)) shrinks the bulge in $\hat{B}_{i-1/2}$ by setting the entries in positions $(i+1,i),\ldots,(k,i)$ to zero. To see how Z_i^{-1} acts, we consider the inverse equation $Z_i^{-1}\hat{B}_{i-1/2}^{-1} = \hat{B}_i^{-1}$. We know that $\hat{B}_{i-1/2}$ has the form

$$\hat{B}_{i-1/2} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & B_{33} \end{bmatrix},$$

where $B_{11} \in \mathbb{C}^{(i-1)\times(i-1)}$, $B_{22} \in \mathbb{C}^{(k-i+1)\times(k-i+1)}$, and $B_{33} \in \mathbb{C}^{(n-k)\times(n-k)}$. B_{11} and B_{33} are upper triangular, but B_{22} is full, as it holds the bulge. Clearly $\hat{B}_{i-1/2}^{-1}$ has exactly the same zero structure; it is upper triangular with a bulge of order $k-i$. $\hat{B}_i$ has the same form as $\hat{B}_{i-1/2}$, except that the first column of B_{22} has been set to upper triangular form. It is upper triangular with a bulge of order $k-i-1$. Its inverse has exactly the same structure. Thus the transformation $Z_i^{-1}\hat{B}_{i-1/2}^{-1} = \hat{B}_i^{-1}$ shrinks the bulge in $\hat{B}_{i-1/2}^{-1}$ by setting the entries in positions $(i+1,i),\ldots,(k,i)$ to zero. The submatrix $\tilde{Z}_i^{-1}$ acts on rows i through k. Within these rows, our interest focuses on column i, the first column of B_{22}^{-1}, as this is where the elimination takes place. Call this column y; that is, $y = B_{22}^{-1}e_1$. Then $\tilde{Z}_i^{-1}y = \beta e_1$, where $\beta \neq 0$ because $\hat{B}_i^{-1}$ is nonsingular. Thus $\tilde{Z}_i$ is an elimination matrix of y. Since it is also an elimination matrix of x, y must be a multiple of x. Indeed $y = \beta\tilde{Z}_i e_1 = \beta\alpha^{-1}x$. Since $y = B_{22}^{-1}e_1$, we can obtain it by solving the small system $B_{22}y = e_1$. We have now proved the following theorem.

THEOREM 4.11. *For $1 \leq i \leq \rho$ let $k = \min\{i+m, n\}$. Let B_{22} denote the principal submatrix of $\hat{B}_{i-1/2}$ consisting of rows and columns i through k. Let y be the unique solution of $B_{22}y = e_1$. Then $Z_i = \operatorname{diag}\{I_{i-1}, \tilde{Z}_i, I_{n-k}\}$, where $\tilde{Z}_i$ is the elimination matrix of y.*

Theorem 4.11 can also be inferred from the equation

$$\hat{B}_{i-1/2}\hat{S}_{i-1} = \hat{R}_i B, \tag{16}$$

which holds by (13). The vector x lies in the ith column of $\hat{S}_{i-1}$, so consider the ith column of (16), partitioned as

$$\begin{bmatrix} B_{11} & B_{12} & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & B_{33} \end{bmatrix} \begin{bmatrix} z \\ x \\ 0 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \\ 0 & R_{32} \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix},$$

where $\hat{B}_{i-1/2}$ is partitioned as before, $z \in \mathbb{C}^{i-1}$, $x \in \mathbb{C}^{k-i+1}$, $R_{11} \in \mathbb{C}^{(i-1)\times i}$, $R_{21} \in \mathbb{C}^{(k-i+1)\times i}$, and $b \in \mathbb{C}^i$. Since the first i columns of $\hat{R}_i$ are upper triangular, R_{21} has only one nonzero entry, $\hat{r}_{ii}^{(i)}$, which lies in the upper right-hand corner. Thus $R_{21} = \gamma e_1 e_i^T$, where $\gamma = \hat{r}_{ii}^{(i)} \neq 0$. We seek x. But clearly $B_{22}x = R_{21}b = \gamma e_1 e_i^T b = \delta e_1$, where $\delta = \gamma e_i^T b = \gamma b_{ii} \neq 0$. Thus we can find $y = \delta^{-1}x$ by solving $B_{22}y = e_1$.

Theorems 4.10 and 4.11 justify the implicit GZ algorithm, which is summarized in (17). Notice that the algorithm takes $n-1$ steps, rather than stopping after ρ steps. In practice ρ is usually unknown because ν is unknown. Even if ν is known in principle, it is not well determined in the presence of roundoff error. When it happens that $\nu > 1$, the implicit algorithm differs from the explicit algorithm only in one way. In this case a small subpencil can be deflated from the bottom of the matrix. The implicit algorithm operates on this subpencil, reducing it to Hessenberg-triangular form, whereas the explicit algorithm does not. This further reduction is useful because we need to calculate the eigenvalues of the subpencil anyway.

IMPLICIT GZ ALGORITHM

(17)

```
for i = 1, ..., n - 1
    k ← min{i + m, n}
    if (i = 1) then
        x ← p(AB^{-1})e_1        (See the discussion following (14).)
        y ← first k entries of x
    else
        y ← [a_{i,i-1}, ..., a_{k,i-1}]^T
    end if
    G̃ ← elimination matrix of y                (*)
    G ← diag{I_{i-1}, G̃, I_{n-k}}
    A ← G^{-1}A,   B ← G^{-1}B
    B_22 ← rows, columns i through k of B
    z ← B_22^{-1} e_1
    Z̃ ← elimination matrix of z                (**)
    Z ← diag{I_{i-1}, Z̃, I_{n-k}}
    A ← AZ,   B ← BZ
```

Remarks. For standard eigenvalue problems ($B = I$), we have already noted that $Z_i = G_i$ for $i = 1, \ldots, n-1$. Thus we have $Z = G$ at each step of (17). This special case is the implicit GR algorithm.

In [2] Bunse-Gerstner and Elsner developed a new version of the QZ algorithm for unitary pencils, which are of interest in certain signal-processing applications. Instead of using the Hessenberg-triangular form, they introduced a more condensed block-diagonal form. Since our development is built upon the Hessenberg-triangular form, it does not encompass the algorithm of [2] as a special case. However, the methodology used here can be adapted to that situation and used to derive that algorithm and its generalizations to higher multiplicity. In particular, if A and B are unitary, the equations in Lemma 4.1 are joined by the two related equations

$$\hat{A}^* \hat{R} = \hat{S} A^* \quad \text{and} \quad \hat{B}^* \hat{R} = \hat{S} B^*.$$

Whereas the equations of Lemma 4.1 yield information about zeros below the main diagonal (e.g., upper Hessenberg form is preserved from one iteration to the next), these new equations give information about zeros above the main diagonal. In particular, block diagonal forms are preserved from one iteration to the next. Of course, we must now insist that the G and Z transformations be unitary, so that the pencil stays unitary from one iteration to the next.

4.5. The generic bulge-chasing algorithm. If one compares the implicit GZ algorithm (17) with the standard double-step QZ algorithm, as presented in, for

example, EISPACK [5], one notices an important difference. In (17) the transformation Z_i is designed to clear out one column of the bulge in B. In contrast, the corresponding transformation in the standard code eliminates the entire B bulge before proceeding to the computation of G_{i+1}. The bulge is annihilated row by row, using one elimination matrix for each row. In the case $m = 2$ (as in the EISPACK code) the added cost of doing this is small. However, as m is made larger, the cost difference becomes significant. On the other hand, the standard procedure may be substantially more stable. The reason for this is that the Z_i calculated in (17) is designed so that the transformation $Z_i^{-1}\hat{B}_{i-1/2}^{-1} = \hat{B}_i^{-1}$ will clear out one column in the bulge in the inverse matrix. As a consequence, the transformation $\hat{B}_{i-1/2}Z_i = \hat{B}_i$ will also, in principle, remove one column from the bulge in $\hat{B}_{i-1/2}$. However, these zeros are introduced only incidentally; they are not enforced by the transformation. Therefore, in the presence of roundoff errors these numbers will not be exactly zero. They may sometimes be far enough from zero that they cannot be set to zero without compromising the stability of the algorithm. After all, if the submatrix B_{22} should be ill conditioned at some point, then the solution of $B_{22}z = e_1$ and the resulting Z_i may not be well determined. In contrast, the standard procedure introduces the desired zeros explicitly through the mechanism of eliminating the entire bulge from B. Therefore there is no question of having to set to zero some numbers that should be, but are not quite, zero. We note finally that in the extreme case of singular B, the implicit GZ algorithm (17) breaks down, whereas the standard procedure does not.

It is therefore desirable to broaden our class of algorithms to include procedures of this type. To this end we introduce a *generic bulge-chasing algorithm*, which is exactly algorithm (17), except that at the two steps labelled $(*)$ and $(**)$, where $\tilde{G}$ and $\tilde{Z}$ are chosen, we do not require that they be determined by a specific elimination rule. At $(*)$ we allow any nonsingular $\tilde{G}$ for which $\tilde{G}^{-1}y = \alpha e_1$ for some α, and likewise for $\tilde{Z}$ at $(**)$. That $\tilde{G}^{-1}y = \alpha e_1$ means exactly that (at the ith step) the premultiplication by G^{-1} causes entries $(i+1, i-1), \ldots, (i+k, i-1)$ of the A matrix to be transformed to zero. Similarly, that $\tilde{Z}^{-1}z = \beta e_1$, where $z = B_{22}^{-1}e_1$, means neither more nor less than that (at the ith step) the postmultiplication by Z causes entries $(i+1, i), \ldots, (i+k, i)$ of the B matrix to be transformed to zero. We do not require that the vector z be calculated explicitly, only that the transformations produce the desired zeros. (Indeed this can be done even if B_{22} is singular, in which case z is not well defined.) Thus algorithms that annihilate the entire bulge in B at each step fit into this structure.

It is worth mentioning a class of algorithms that avoids the high cost of eliminating the entire bulge in B at each step by never allowing that bulge to build up in the first place [9, p. 75]. Suppose that after $i-1$ steps of a generic bulge-chasing algorithm we have $\hat{A}_{i-1} - \lambda\hat{B}_{i-1}$, where $\hat{B}_{i-1}$ has no bulge. The next step is to build an elimination matrix G_i such that G_i^{-1} annihilates the entries in positions $(i+1, i-1), \ldots, (i+k, i-1)$ of $\hat{A}_{i-1}$. One can build such a G_i^{-1} as a product of simpler matrices $G_{i,i+1}^{-1} \cdots G_{i,i+k}^{-1}$, where each $G_{i,j}^{-1}$ acts only on rows $j-1$ and j (that is, its nontrivial part is a 2×2 matrix) and annihilates the $(j, i-1)$ entry. One easily checks that the matrix $\hat{B}_{i-1/2} = G_i^{-1}\hat{B}_{i-1}$ fails to be upper triangular only in that the entries in positions $(i+1, i)$, $(i+2, i+1), \ldots, (i+k, i+k-1)$ are nonzero. These can be annihilated by a transformation $\hat{B}_i = \hat{B}_{i-1/2}Z_i$, where $Z_i = Z_{i,i+k} \cdots Z_{i,i+1}$, and each $Z_{i,j}$ acts only on columns $j-1$ and j and annihilates the entry in position $(j, j-1)$. Thus $\hat{B}_i$ is upper triangular. Algorithms of this type conform to the framework of the generic bulge-chasing algorithm.

Our analysis of the generic bulge-chasing algorithm will show that each iteration amounts to an iteration of a generic GZ algorithm. Thus the generic bulge-chasing algorithm lies within the class of algorithms whose convergence properties we studied in §3. To this end we introduce some notation. In fact the notation is identical to notation that we used earlier, but the symbols now carry slightly different meanings. Let G_i and Z_i denote the transformations produced at the ith step of the generic chasing algorithm (as we have already done in the previous paragraph), let $\hat{G}_i = G_1 \cdots G_i$, $\hat{Z}_i = Z_1 \cdots Z_i$, $\hat{A}_i = \hat{G}_i^{-1} A \hat{Z}_i$, $\hat{B}_i = \hat{G}_i^{-1} B \hat{Z}_i$, $\hat{A}_{i-1/2} = \hat{G}_i^{-1} A \hat{Z}_{i-1}$, and $\hat{B}_{i-1/2} = \hat{G}_i^{-1} B \hat{Z}_{i-1}$, $i = 1, \ldots, n-1$. These matrices may be different from those featured in Theorems 4.6 and 4.9, but they have the same bulge structure; it is exactly the function of the generic bulge-chasing algorithm to enforce this structure. Let $\hat{A} = \hat{A}_{n-1}$ and $\hat{B} = \hat{B}_{n-1}$. These are the final products of (one iteration of) the generic bulge-chasing algorithm. $\hat{A}$ is upper Hessenberg, and $\hat{B}$ is upper triangular. Assuming once again that B is nonsingular, let $\hat{R}_0 = p(AB^{-1})$ and $\hat{S}_0 = p(B^{-1}A)$, as before, and let $\hat{R}_i = G_i^{-1} \hat{R}_{i-1}$ and $\hat{S}_i = Z_i^{-1} \hat{S}_{i-1}$ for $i = 1, \ldots, n-1$. Now we are using the matrices G_i and Z_i to define $\hat{R}_i$ and $\hat{S}_i$, whereas in the development of the explicit GZ algorithm we used $\hat{R}_i$ and $\hat{S}_i$ to define G_i and Z_i. It is an immediate consequence of the new definitions that

$$p(AB^{-1}) = \hat{G}_i \hat{R}_i \quad \text{and} \quad p(B^{-1}A) = \hat{Z}_i \hat{S}_i$$

for $i = 1, \ldots, n-1$. The matrices $\hat{R}_i$ and $\hat{S}_i$ defined in connection with the explicit algorithm were partially upper triangular. Whether or not the new $\hat{R}_i$ and $\hat{S}_i$ have that property is not immediately clear from the definition. In fact they do, as the following theorem shows.

THEOREM 4.12. *For $i = 1, \ldots, n-1$, the matrices $\hat{R}_i$ and $\hat{S}_i$ defined in the previous paragraph both have the form*

$$\begin{bmatrix} X_{11} & X_{12} \\ 0 & X_{22} \end{bmatrix}, \tag{18}$$

where $X_{11} \in \mathbb{C}^{i \times i}$ is upper triangular.

Proof. The proof is by induction on i. First let $i = 1$. The transformation G_1 is designed to annihilate $p(AB^{-1})e_1$, the first column of $p(AB^{-1}) = \hat{R}_0$. Since $\hat{R}_1 = G_1^{-1} \hat{R}_0$, $\hat{R}_1$ must have its first column in upper triangular form, as claimed.

Now we show that, for $i = 1, \ldots, n-1$, if $\hat{R}_i$ has the stated form, then so does $\hat{S}_i$. Since $\hat{B}_i = \hat{G}_i^{-1} B \hat{Z}_i$, $p(AB^{-1}) = \hat{G}_i \hat{R}_i$ and $p(B^{-1}A) = \hat{Z}_i \hat{S}_i$, we can apply Lemma 4.1 to obtain $\hat{B}_i \hat{S}_i = \hat{R}_i B$ or, equivalently,

$$\hat{S}_i = \hat{B}_i^{-1} \hat{R}_i B.$$

Each of the three matrices on the right-hand side has the form (18), where $X_{11} \in \mathbb{C}^{i \times i}$, so $\hat{S}_i$ must also have this form.

We complete the induction by showing that for $i = 2, \ldots, n-1$, if $\hat{R}_{i-1}$ and $\hat{S}_{i-1}$ have the stated form, then so does $\hat{R}_i$. Certainly the first $i-1$ columns of $\hat{R}_i$ are in upper triangular form, for this is true of $\hat{R}_{i-1}$, and the transformation $\hat{R}_i = G_i^{-1} \hat{R}_{i-1}$ does not alter these columns, as one easily checks. Thus we can focus on the ith column of $\hat{R}_i$. Since $\hat{A}_{i-1/2} = \hat{G}_i^{-1} A \hat{Z}_{i-1}$, $p(AB^{-1}) = \hat{G}_i \hat{R}_i$ and $p(B^{-1}A) = \hat{Z}_{i-1} \hat{S}_{i-1}$, we can apply Lemma 4.1 to obtain

$$\hat{A}_{i-1/2} \hat{S}_{i-1} = \hat{R}_i A. \tag{19}$$

We wish to pick out the ith column of $\hat{R}_i$. Noting that $a_{i,i-1} \neq 0$, we examine column $i-1$ of (19), partitioned unsymmetrically as

$$\begin{bmatrix} \hat{A}_{11}^{(i-1/2)} & \hat{A}_{12}^{(i-1/2)} \\ 0 & \hat{A}_{22}^{(i-1/2)} \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} w \\ 0 \end{bmatrix}, \tag{20}$$

where $\hat{A}_{11}^{(i-1/2)} \in \mathbb{C}^{i \times i-1}$, $R_{11} \in \mathbb{C}^{i \times i}$, $v \in \mathbb{C}^{i-1}$, and $w \in \mathbb{C}^i$. The last entry in w is $a_{i,i-1}$. We already know that the first $i-1$ columns of $\hat{R}_i$ are upper triangular. This implies that R_{11} is upper triangular, and all but the last column of R_{12} is zero. Thus $R_{12} = xe_i^T$ for some x. If we can show that $x = 0$, we will be finished. Equating second components of (20), we have $0 = R_{21}w = xe_i^T w = xa_{i,i-1}$. Since $a_{i,i-1} \neq 0$, we have $x = 0$. □

Let $\hat{G} = \hat{G}_{n-1}$, $\hat{Z} = \hat{Z}_{n-1}$, $\hat{R} = \hat{R}_{n-1}$, and $\hat{S} = \hat{S}_{n-1}$. Then $\hat{R}$ and $\hat{S}$ are upper triangular by Theorem 4.12 with $i = n-1$, and

$$\hat{A} = \hat{G}^{-1} A \hat{Z} \quad \text{and} \quad \hat{B} = \hat{G}^{-1} B \hat{Z}, \tag{21}$$

where

$$p(AB^{-1}) = \hat{G}\hat{R} \quad \text{and} \quad p(B^{-1}A) = \hat{Z}\hat{S}. \tag{22}$$

We conclude that one iteration of the generic bulge-chasing algorithm amounts to one iteration of the generic GZ algorithm.

REFERENCES

[1] W. Bunse and A. Bunse-Gerstner, *Numerische lineare Algebra,* Teubner, Stuttgart, 1985.

[2] A. Bunse-Gerstner and L. Elsner, *Schur parameter pencils for the solution of the unitary eigenvalue problem,* Linear Algebra Appl., 154–156 (1991), pp. 741–778.

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

[4] J. Demmel and B. Kågström, *GUPTRI,* NETLIB, 1991.

[5] B. S. Garbow, J. M. Doyle, J. J. Dongarra, and C. B. Moler, *Matrix Eigensystem Routines–EISPACK Guide Extension,* Springer-Verlag, New York, 1977.

[6] G. Golub and C. Van Loan, *Matrix Computations,* Second Edition, The Johns Hopkins University Press, Baltimore, MD, 1989.

[7] J. B. Haag and D. S. Watkins, *QR-like algorithms for the nonsymmetric eigenvalue problem,* ACM Trans. Math. Software, 19 (1993), pp. 407–418.

[8] L. Kaufman, *The LZ algorithm to solve the generalized eigenvalue problem,* SIAM J. Numer. Anal., 11 (1974), pp. 997–1024.

[9] ———, *Some thoughts on the QZ algorithm for solving the generalized eigenvalue problem,* ACM Trans. Math. Software, 3 (1977), pp. 65–75.

[10] G. S. Miminis and C. C. Paige, *Implicit shifting in the QR algorithm and related algorithms,* SIAM J. Matrix Anal. Appl., 12 (1991), pp. 385–400.

[11] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems,* SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[12] G. W. Stewart, *Introduction to Matrix Computations,* Academic Press, New York, 1973.

[13] P. Van Dooren, *The computation of Kronecker's canonical form of a singular pencil,* Linear Algebra Appl., 27 (1979), pp. 103–140.

[14] R. C. Ward, *The combination shift QZ algorithm,* SIAM J. Numer. Anal., 12 (1975), pp. 835–853.

[15] D. S. WATKINS, *Fundamentals of Matrix Computations,* John Wiley and Sons, New York, 1991.
[16] D. S. WATKINS AND L. ELSNER, *Self-equivalent flows associated with the generalized eigenvalue problem,* Linear Algebra Appl., 118 (1989), pp. 107–127.
[17] ———, *Convergence of algorithms of decomposition type for the eigenvalue problem,* Linear Algebra Appl., 143 (1991), pp. 19–47.
[18] ———, *Chasing algorithms for the eigenvalue problem,* SIAM J. Matrix Anal. Appl., 12 (1991), pp. 374–384.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 968–973, July 1994

017

THE DIAGONAL TORUS OF A MATRIX UNDER SPECIAL UNITARY EQUIVALENCE*

ROBERT C. THOMPSON†

Abstract. Given a matrix A, a sufficient condition is given for the vector of diagonal elements of UAV to cover a torus with specified base circle radii as U and V run over the special unitary group.

Key words. singular values, subtracted terms, spectral inequalities

AMS subject classification. 15A18

1. Introduction. Let A be an $n \times n$ matrix with complex elements and singular values $s_1 \geqq \cdots \geqq s_n$. Define $\mathcal{U}_n$ to be the $n \times n$ unitary group. Some years ago, the author [3] and Sing [2] (independently) characterized the principal diagonals of the matrices in the $\mathcal{U}_n \otimes \mathcal{U}_n$ orbit of A, under the group action for which $U \otimes V^*$ takes A to UAV, for U and V in $\mathcal{U}_n$. (V^* is the conjugate transpose of V.) The present paper should be viewed as a supplement to [2] and [3], especially [3].

To explain the principal diagonal characterization, let $d_1, \ldots, d_n$ be complex numbers enumerated such that $|d_1| \geqq \cdots \geqq |d_n|$. Then [3], [2] unitary matrices U and V exist such that the principal diagonal of UAV consists of $d_1, \ldots, d_n$ if and only if

$$\sum_{t=1}^{k} |d_t| \leqq \sum_{t=1}^{k} s_t, \qquad k = 1, \ldots, n,$$

$$\sum_{t=1}^{n-1} |d_t| - |d_n| \leqq \sum_{t=1}^{n-1} s_t - s_n.$$

This theorem solved a question posed by Mirsky [1] in 1964.

An unexpected facet of this theorem is the last inequality, each side of which has a subtracted term.

The author, in [3], also investigated the proper orthogonal counterpart of this theorem in which A is a real matrix with given singular values and nonnegative determinant, and the unitary group $\mathcal{U}_n$ is replaced by the real proper orthogonal group $\mathcal{O}_n$. The result was a somewhat similar theorem, except that under certain circumstances the minus term on the left side of the last inequality is replaced with a plus term, with the minus term on the right still being present.

Since the publication of [3], it has been realized that singular values frequently satisfy inequalities with subtracted terms; see [4] and [5].

Motivated by the real proper orthogonal theorem just mentioned, the objective of this paper is to investigate the vectors of principal diagonal elements in the $\mathcal{SU}_n \otimes \mathcal{SU}_n$ orbit of a given matrix A, where $\mathcal{SU}_n$ is the $n \times n$ special unitary group, comprising the unitary matrices with determinant one. While the diagonal vectors may satisfy intricate conditions (see the lemma in §3) a tidy theorem is obtained if attention is

* Received by the editors February 19, 1992; accepted for publication (in revised form) February 8, 1993. The work of this author was supported in part by a National Science Foundation grant.

† Department of Mathematics, University of California, Santa Barbara, California 93106-3080 (thompson@math.ucsb.edu).

focused on diagonals covering a torus (definition below) in n space. Our theorem furnishes a sufficient condition on the radii of the base circles in an n-torus in order that the diagonal vectors of the matrices in the orbit of A will cover it.

We say that a vector $[d_1, \dots, d_n]$ of diagonal elements of a matrix UAV in the $\mathcal{SU}_n \otimes \mathcal{SU}_n$ orbit of A belongs to a torus of diagonal vectors if $[\zeta_1 d_1, \cdots, \zeta_n d_n]$ are diagonal vectors of matrices in the orbit of A for every choice of $\zeta_1, \dots, \zeta_n$ with $|\zeta_1| = \cdots = |\zeta_n| = 1$. The radii of the base circles in the torus are $|d_1|, \dots, |d_n|$.

2. The torus theorem. When studying the diagonal elements $d_1, \dots, d_n$ of a matrix UAV in the orbit of A, no generality is lost if we assume that $|d_1| \geqq \cdots \geqq |d_n|$ since arbitrary rearrangements of the principal diagonal are obtained by passing to $P(UAV)P^*$ for a suitable generalized permutation matrix P with determinant one.

THEOREM. *Let A be a matrix with singular values $s_1 \geqq \cdots \geqq s_n$, and let $d_1, \dots, d_n$ be complex numbers with $|d_1| \geqq \cdots \geqq |d_n|$. If*

$$\sum_{t=1}^{k} |d_t| \leqq \sum_{t=1}^{k} s_t, \qquad k = 1, \dots, n-1, \tag{1}$$

$$\sum_{t=1}^{n} |d_t| \leqq \sum_{t=1}^{n-1} s_t - s_n, \tag{2}$$

then the diagonal vectors of the matrices in the $\mathcal{SU}_n \otimes \mathcal{SU}_n$ orbit of A cover the torus with $|d_1|, \dots, |d_n|$ as the base circle radii.

In the last condition, there is no subtracted term on the left even though there is one on the right.

We speculate that if the diagonal vectors of the matrices in the $\mathcal{SU}_n \otimes \mathcal{SU}_n$ orbit of A cover a torus with base circle radii $|d_1| \geqq \cdots \geqq |d_n|$, then the $|d_i|$ satisfy the just displayed inequalities, of which only the last is conjectural.

In passing, we note that the $\mathcal{SU}_n \otimes \mathcal{SU}_n$ orbit of a matrix A with singular values $s_1 \geqq \cdots \geqq s_n$ contains a unique matrix $\operatorname{diag}(s_1, \dots, s_{n-1}, \zeta s_n)$ where $|\zeta| = 1$. This is the special unitary version of the singular value decomposition.

3. The 2×2 case. The following lemma establishes the theorem and its converse in the 2×2 case.

LEMMA. *Let A be a 2×2 matrix with singular values $s_1 \geqq s_2$. Then the diagonal vectors $[d_1, d_2]$ of the matrices in the $\mathcal{SU}_2 \otimes \mathcal{SU}_2$ orbit of A cover a torus with fixed base circle radii $|d_1|, |d_2|$ if and only if $|d_1| + |d_2| \leqq s_1 - s_2$.*

Proof. Let $\det A = \zeta s_1 s_2$, where $|\zeta| = 1$. We wish to construct a matrix

$$\begin{bmatrix} d_1 & z_1 \\ z_2 & d_2 \end{bmatrix}$$

in the $\mathcal{SU}_2 \otimes \mathcal{SU}_2$ orbit of A, where z_1, z_2 are complex numbers to be determined. This matrix belongs to the orbit if and only if

$$|d_1|^2 + |d_2|^2 + |z_1|^2 + |z_2|^2 = s_1^2 + s_2^2,$$
$$d_1 d_2 - z_1 z_2 = \zeta s_1 s_2.$$

A choice for $\arg(z_1 z_2)$ exists to satisfy the second condition if and only if

$$|z_1||z_2| = |d_1 d_2 - \zeta s_1 s_2|.$$

Thus the existence of the desired matrix requires that the polynomial

$$w^2 - (s_1^2 + s_2^2 - |d_1|^2 - |d_2|^2)w + |d_1 d_2 - \zeta s_1 s_2|^2$$

in w have two nonnegative roots $|z_1|^2$, $|z_2|^2$. An equivalent form of this condition is

$$|d_1 d_2 - \zeta s_1 s_2|^2 \leqq \frac{1}{4}(s_1^2 + s_2^2 - |d_1|^2 - |d_2|^2)^2$$

with

$$s_1^2 + s_2^2 - |d_1|^2 - |d_2|^2 \geqq 0.$$

This inequality pair is the necessary and sufficient condition for the $\mathcal{SU}_2 \otimes \mathcal{SU}_2$ orbit of A to contain a matrix with diagonal d_1, d_2.

We require that when d_1, d_2 satisfy these inequalities, so do $\zeta_1 d_1$ and $\zeta_2 d_2$ for any choice of ζ_1, ζ_2 with modulus one. Choosing ζ_1 and ζ_2 to maximize the left side of the first inequality, and using the second, we obtain this necessary and sufficient condition:

$$(|d_1||d_2| + s_1 s_2) \leqq \frac{1}{2}(s_1^2 + s_2^2 - |d_1|^2 - |d_2|^2).$$

This condition rearranges to become $(|d_1| + |d_2|)^2 \leqq (s_1 - s_2)^2$, and therefore to become $|d_1| + |d_2| \leqq s_1 - s_2$. □

4. The proof. The hypotheses and conclusion of the theorem are valid for a matrix A whenever they are valid for ξA, where ξ is a complex number with $|\xi| = 1$. Choosing ξ appropriately, it may be assumed that $\det A \geqq 0$. Then A is in the $\mathcal{SU}_n \otimes \mathcal{SU}_n$ orbit of $\operatorname{diag}(s_1, \ldots, s_n)$.

The following argument uses the proof of Lemma 1 of [3]. We briefly describe the aspect of this proof that we need. Let complex numbers δ_1, δ_2 and nonnegative real numbers $\sigma_1 \geqq \sigma_2$ satisfy $|\delta_1| + |\delta_2| \leqq \sigma_1 + \sigma_2, |\delta_1| - |\delta_2| \leqq \sigma_1 - \sigma_2, |\delta_2| - |\delta_1| \leqq \sigma_1 - \sigma_2$. Then unitary matrices U, V with determinant 1 exist such that $U\operatorname{diag}(\sigma_1, \sigma_2)V$ has principal diagonal $|\delta_1|, |\delta_2|$. Multiplying from the left by $D = \operatorname{diag}(e^{iarg\delta_1}, e^{iarg\delta_2})$, we obtain unitary matrices DU and V with $(DU)\operatorname{diag}(\sigma_1, \sigma_2)V$ having δ_1, δ_2 as principal diagonal elements, with $\det(DU) = e^{i(arg\delta_1 + arg\delta_2)}$, and with $\det V = 1$.

We now prove the theorem by induction on the matrix size, using techniques similar to those in [3]. Let the matrices be $n \times n$, and let $d_1, \ldots, d_n$ be given with $|d_1| \geqq \cdots \geqq |d_n|$. The $n = 1$ case is trivial (the assumptions imply $d_1 = s_1 = 0$), and the $n = 2$ case is settled by the lemma, so let $n > 2$.

First, suppose that $s_i \geqq |d_1| \geqq s_{i+1}$ for some i with $i \leqq n - 2$. Let $t = s_i + s_{i+1} - |d_1|$. Then $s_i \geqq t \geqq s_{i+1}$, $|d_1| + t \leqq s_i + s_{i+1}$, $|d_1| - t \leqq s_i - s_{i+1}$, $t - |d_1| \leqq s_i - s_{i+1}$. By the remarks above, we can find unitary matrices U and V such that

$$\begin{bmatrix} d_1 & * \\ * & t \end{bmatrix} = U\operatorname{diag}(s_i, s_{i+1})V \text{ with } \det U = e^{iarg d_1} \text{ and } \det V = 1.$$

We also have

$$\begin{aligned}
|d_2| &\leqq s_1, \\
&\cdots\cdots \\
|d_2| + \cdots + |d_i| &\leqq s_1 + \cdots + s_{i-1}, \\
|d_2| + \cdots + |d_{i+1}| &\leqq s_1 + \cdots + s_{i-1} + t, \\
&\cdots\cdots \\
|d_2| + \cdots + |d_n| &\leqq s_1 + \cdots + s_{i-1} + t + s_{i+2} + \cdots + s_{n-1} - s_n.
\end{aligned}$$

Therefore, by induction, we can construct a matrix in the $\mathcal{SU}_{n-1} \otimes \mathcal{SU}_{n-1}$ orbit of $\operatorname{diag}(t, s_1, \ldots, s_{i-1}, s_{i+2}, \ldots, s_n)$ with diagonal elements $\zeta^{-1}d_2, \ldots, \zeta^{-1}d_n$, where $\zeta = e^{-i(\arg d_1)/(n-1)}$. Multiplying by ζ, we get a matrix $A_1 = U_1 \operatorname{diag}(t, s_1, \ldots, s_{i-1}, s_{i+2}, \ldots, s_n)V_1$ with diagonal elements $d_2, \ldots, d_n$, where U_1 and V_1 are unitary with $\det U_1 = \zeta^{n-1}$, $\det V_1 = 1$. And now

$$\begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} U & 0 \\ 0 & I_{n-2} \end{bmatrix} \operatorname{diag}(s_i, s_{i+1}, s_1, \ldots, s_{i-1}, s_{i+2}, \ldots, s_n) \begin{bmatrix} V & 0 \\ 0 & I_{n-2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}$$

has $d_1, \ldots, d_n$ as its principal diagonal with the left and right unitary factors

$$\begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} U & 0 \\ 0 & I_{n-2} \end{bmatrix}, \qquad \begin{bmatrix} V & 0 \\ 0 & I_{n-2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}$$

having determinants $\det U_1 \det U = e^{i\arg d_1}\zeta^{n-1} = 1$ and $\det V \det V_1 = 1$. Since $\operatorname{diag}(s_i, s_{i+1}, s_1, \ldots, s_{i-1}, s_{i+2}, \ldots, s_n) = PAQ$ for suitable unitary matrices P, Q with determinant 1, this case is finished.

Now let $|d_1| \leqq s_{n-1}$. We choose a real number t satisfying

$$\left.\begin{matrix} |d_1| - s_{n-1} + s_n \\ 0 \end{matrix}\right\} \leqq t \leqq \begin{cases} s_{n-1} + s_n - |d_1|, \\ s_{n-1} - s_n + |d_1|, \\ s_1 + \cdots + s_{n-2} - |d_2| - \cdots - |d_n|. \end{cases}$$

This number t will exist if and only if the six inequalities comparing the extreme left and right expressions are all satisfied, and this is the case exactly when $|d_2| + \cdots + |d_n| \leqq s_1 + \cdots + s_{n-2}$. Assuming that this holds, we imitate the proof in the previous part with $i = n - 1$.

First, we have

$$\begin{aligned} |d_1| + t &\leqq s_{n-1} + s_n, \\ |d_1| - t &\leqq s_{n-1} - s_n, \\ t - |d_1| &\leqq s_{n-1} - s_n, \end{aligned}$$

implying that $t \leqq s_{n-1}$, and second we have

$$\begin{aligned} |d_2| &\leqq s_1, \\ &\cdots\cdots \\ |d_2| + \cdots + |d_{n-1}| &\leqq s_1 + \cdots + s_{n-2}, \\ |d_2| + \cdots + |d_n| &\leqq s_1 + \ldots s_{n-2} - t. \end{aligned}$$

By the first set of inequalities and the proof of Lemma 1 of [3], unitary matrices U and V exist such that $U\operatorname{diag}(s_{n-1}, s_n)V$ has diagonal elements d_1 and t, with $\det U = e^{i\arg d_1}$ and $\det V = 1$. By the second set of inequalities, and induction, noting that $t \leqq s_{n-1}$, the $\mathcal{SU}_{n-1} \otimes \mathcal{SU}_{n-1}$ orbit of $\operatorname{diag}(t, s_1, \ldots, s_{n-2})$ contains a matrix with diagonal elements $\zeta^{-1}d_2, \ldots, \zeta^{-1}d_n$, where $\zeta = e^{-i(\arg d_1)/(n-1)}$. Multiplying by ζ, we obtain unitary matrices U_1 and V_1 with $\det U_1 = \zeta^{n-1}$ and $\det V_1 = 1$ such that

$$U_1 \operatorname{diag}(t, s_1, \ldots, s_{n-2})V_1$$

has diagonal elements $d_2, \ldots, d_n$. Now complete the argument as in the previous case.

So now assume that

$$|d_2| + \cdots + |d_n| > s_1 + \cdots + s_{n-2}, \qquad |d_1| \leqq s_{n-1}.$$

We choose a number t satisfying

$$|d_1| + |d_n| - s_{n-1} \leqq t \leqq \begin{cases} s_1 + \cdots + s_{n-2} - s_n - |d_2| - \cdots - |d_{n-1}|, \\ s_{n-1} - |d_1| + |d_n|, \\ |d_{n-1}|. \end{cases}$$

The left member does not exceed each of the right members, easily seen using $|d_1| \leqq s_{n-1}$, so a real value for t exists. And $|d_1| + |d_n| - s_{n-1} \geqq 0$ since we have: from $|d_2| + \cdots + |d_n| > s_1 + \cdots + s_{n-2}$, we get $s_1 + \cdots + s_{n-3} + |d_{n-1}| + |d_n| > s_1 + \cdots + s_{n-2}$, and therefore $|d_{n-1}| + |d_n| > s_{n-2}$. This implies that $|d_1| + |d_n| > s_{n-1}$, and therefore t is positive.

Then two sets of conditions are satisfied. The first is

$$\begin{gathered} |d_2| \leqq s_1, \\ \cdots\cdots \\ |d_2| + \cdots + |d_{n-1}| \leqq s_1 + \cdots + s_{n-2}, \\ |d_2| + \ldots |d_{n-1}| + t \leqq s_1 + \cdots + s_{n-2} - s_n, \end{gathered}$$

and the second is

$$\begin{aligned} |d_1| + |d_n| &\leqq t + s_{n-1}, \\ |d_1| - |d_n| &\leqq s_{n-1} - t. \end{aligned}$$

Let ζ satisfy $\zeta^{n-1} e^{i(\arg d_1 + \arg d_n)} = 1$.

Because of the first set of conditions, and by induction, we may find unitary matrices U and V with determinant 1 such that

$$U \operatorname{diag}(s_1, \ldots, s_{n-2}, s_n) V$$

has diagonal elements

$$\zeta^{-1} d_2, \ldots, \zeta^{-1} d_{n-1}, \zeta^{-1} t.$$

Multiplying by ζ, we obtain a matrix

$$(\zeta U) \operatorname{diag}(s_1, \ldots, s_{n-2}, s_n) V$$

with diagonal elements $d_2, \ldots, d_{n-1}, t$ and ζU and V having determinants ζ^{n-1} and 1.

By the second set of conditions, and the proof of Lemma 1 of [3], we may choose unitary matrices U_1 and V_1 such that $U_1 \operatorname{diag}(t, s_{n-1}) V_1$ has diagonal elements d_1, d_n, where U_1 has determinant $e^{i(\arg d_1 + \arg d_n)}$, and $\det V_1 = 1$.

Then

$$\begin{bmatrix} I & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} \zeta U & 0 \\ 0 & 1 \end{bmatrix} \operatorname{diag}(s_1, \ldots, s_{n-2}, s_n, s_{n-1}) \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & V_1 \end{bmatrix}$$

has diagonal elements $d_2, \ldots, d_{n-1}, d_1, d_n$ and the left and right unitary factors have determinant one. □

REFERENCES

[1] L. MIRSKY, *Inequalities and existence theorems in the theory of matrices*, J. Math. Anal. Appl., 9 (1964), pp. 99–118.

[2] F. Y. Sing, *Some results on matrices with prescribed diagonal elements and singular values*, Canad. Math. Bulletin, 19 (1976), pp. 89–92.

[3] Robert C. Thompson, *Singular values, diagonal elements, and convexity*, SIAM J. Appl. Math., 32 (1977), pp. 39–63.

[4] ———, *Singular values and diagonal elements of complex symmetric matrices*, Linear Algebra Appl., 26 (1979), pp. 65–106.

[5] ———, *High, low, and quantitative roads in linear algebra*, Linear Algebra Appl., 162-164 (1992), pp. 23–64.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 974–994, July 1994

018

FAST ESTIMATION OF PRINCIPAL EIGENSPACE USING LANCZOS ALGORITHM*

GUANGHAN XU† AND THOMAS KAILATH‡

Abstract. This paper considers the problem of finding the principal eigenspace and/or eigenpairs of $M \times M$ Hermitian matrices that can be expressed or *approximated* by a low-rank matrix plus a shift, i.e., $\mathbf{A} = \mathbf{B} + \sigma\mathbf{I}$, where $\mathbf{B}$ is a rank d Hermitian matrix and $d \ll M$. Such matrices arise in signal processing, geophysics, dynamic structure analysis, and other fields. The proposed problem can be solved by a full $O(M^3)$ eigendecomposition, or by several more efficient alternatives, e.g., the power, subspace iteration, and Lanczos algorithms. This paper shows that the Lanczos algorithm can exploit the inherent structure and is generally more efficient than other alternatives. More specifically, if $\mathbf{A} = \mathbf{B}+\sigma\mathbf{I}$, the Lanczos algorithm can be used to *exactly* determine the principal eigenspace span$\{\mathbf{B}\}$ and σ with a *finite* amount of computation. If $\mathbf{A}$ is *close* to $\mathbf{B}+\sigma\mathbf{I}$, the Lanczos algorithm can estimate the principal eigenvectors and eigenvalues in $O(M^2d)$ flops. It is shown that the errors in the estimates of the kth principal eigenvalue λ_k and eigenvector $\mathbf{e}_k$ decay at the rate of $\varepsilon^2/(\lambda_k - \sigma)^2$ and $\varepsilon/(\lambda_k - \sigma)$, respectively, where ε is a measure of the mismatch between $\mathbf{A}$ and $\mathbf{B} + \sigma\mathbf{I}$.

Key words. fast eigendecomposition, Krylov subspace, Lanczos algorithm, eigenvalue multiplicity

AMS subject classifications. 65F15, 65J99, 15A18

1. Introduction.

In the fields of signal processing, geophysics, and image compression, we often encounter the problem of estimating the principal eigenvalues and eigenvectors, or often just the principal eigenspace, of an $M \times M$ matrix that can be either *exactly expressed* or *approximated* by a low-rank matrix plus a shift, say

$$\mathbf{A} = \mathbf{B} + \sigma\mathbf{I}, \tag{1}$$

where $\mathbf{B}$ is a rank $d(\ll M)$ positive semidefinite Hermitian matrix and σ is a real number. The eigenvalues of such an $\mathbf{A}$ can be arranged as $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_d > \lambda_{d+1} = \cdots = \lambda_M = \sigma$. We can also easily verify that span$\{\mathbf{B}\}$ = span$\{\mathbf{e}_1, \ldots, \mathbf{e}_d\}$, where $\mathbf{e}_1, \ldots, \mathbf{e}_d$ are the eigenvalues corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_d$, respectively. These will be called the principal eigenvalues and eigenvectors whose span is the principal eigenspace.

1.1. Applications in signal processing.

Such matrix structure arises in several signal processing problems (see, e.g., [14], [15]) where the data vectors $\mathbf{x}(\cdot)$ can be decomposed into a signal part $\mathbf{s}(\cdot)$ and additive noise part $\mathbf{n}(\cdot)$, where $\mathbf{s}(\cdot)$ is confined to a low-dimensional subspace, called the *signal subspace*, while the noise is not so

* Received by the editors July 2, 1990; accepted for publication (in revised form) February 9, 1993. This work was supported in part by Joint Services Program at Stanford University (Army, Navy, Air Force) contract DAAL03-88-C-0011, Army Research Office contract DAAL03-89-K-0109, and grants from General Electric Company and Boeing ARGOSystems.

This manuscript is submitted for publication with the understanding that the U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

† Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, Texas 78712-1084 (xu@dragon.ece.utexas.edu).

‡ Information Systems Laboratory, Stanford University, Stanford, California 94305-4055 (tk@isl.stanford.edu).

restricted. Given such data samples, the goal is to determine the signal subspace. Assuming that the signal and noise are uncorrelated, as is often the case, the ideal data covariance matrix of $\mathbf{x}(\cdot)$ can be expressed as the sum of the signal covariance $\mathbf{B}$ and the noise covariance $\sigma\mathbf{I}$, where σ is the intensity of the white noise,

$$\mathbf{A} = E\{\mathbf{x}(\cdot)\mathbf{x}^H(\cdot)\} = E\{\mathbf{s}(\cdot)\mathbf{s}^H(\cdot)\} + E\{\mathbf{n}(\cdot)\mathbf{n}(\cdot)\} = \mathbf{B} + \sigma\mathbf{I}. \tag{2}$$

Since $\mathbf{s}(\cdot)$ lies in the d-dimensional signal subspace, its covariance also has rank d, and span$\{\mathbf{B}\}$ will be the signal subspace. Then, as mentioned above, span$\{\mathbf{B}\}$ = span$\{\mathbf{e}_1, \ldots, \mathbf{e}_d\}$.

In reality, however, the ideal covariance matrix (obtained from an *infinite* amount of data samples) is often not available. Instead, one has the so-called *sample* covariance matrix $\hat{\mathbf{A}}$ that is estimated based upon a finite number (N) data samples. Nevertheless, for a reasonably large N, the estimation error is of order $O(1/\sqrt{N})$ (see, e.g., [2]), i.e.,

$$\hat{\mathbf{A}} = \mathbf{A} + O(1/\sqrt{N}), \tag{3}$$

and for the eigenvalues and eigenvectors, we have

$$\hat{\mathbf{e}}_i = \mathbf{e}_i + O(1/\sqrt{N}), \quad i = 1, \ldots, d, \quad \hat{\lambda}_i = \lambda_i + O(1/\sqrt{N}), \quad i = 1, \ldots, M, \tag{4}$$

where $(\hat{\lambda}_i, \hat{\mathbf{e}}_i)$ denotes the ith eigenpair of $\hat{\mathbf{A}}$. By (4) $\hat{\lambda}_{d+1} \approx \cdots \approx \hat{\lambda}_M \approx \sigma$ and they are confined in a small region of size $O(1/\sqrt{N})$ about σ. In [4], it was shown that the maximum likelihood (optimal) estimate of the signal subspace is span$\{\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_d\}$, which has the $O(1/\sqrt{N})$ estimation error by (4).

1.2. Other applications. In dynamic structural analysis and image compression, among others, the matrix under consideration $\mathbf{A}$ ($M \times M$) has a few dominant eigenvalues and many much smaller ones, i.e., $\lambda_1 \geq \cdots \geq \lambda_d \gg \lambda_{d+1} \geq \cdots \geq \lambda_M$. The objective is to find a small number of dominant modes, i.e., the principal eigenvalues and eigenvectors $\{\lambda_i, \mathbf{e}_i\}_{i=1}^d$, so that the original large matrix $\mathbf{A}$ can be approximated by a rank-d matrix $\hat{\mathbf{A}} = \sum_{k=1}^d \mathbf{e}_k \lambda_k \mathbf{e}_k^H$. According to the Eckart–Young theorem, such an approximation $\hat{\mathbf{A}}$ is optimal since it yields the minimum error $\mathbf{A} - \hat{\mathbf{A}}$ in the Frobenius norm sense among all possible rank-d matrices. If all the nonprincipal eigenvalues are relatively small and can be bounded by a small quantity ε, i.e., $\lambda_k < \varepsilon$ for $d + 1 \leq k \leq M$, then the approximation error is of the order of ε, i.e.,

$$\|\mathbf{A} - \hat{\mathbf{A}}\|_F = \sqrt{\lambda_{d+1}^2 + \cdots + \lambda_M^2} = O(\varepsilon). \tag{5}$$

1.3. Problem statement and conventional approaches. Summarizing the above potential applications, we can make the following problem statement.

Given a Hermitian matrix $\mathbf{A} = \mathbf{B} + \sigma\mathbf{I} + O(\varepsilon)$, *where* $\mathbf{B}$ *is a* rank-d *matrix,* $d \ll M$ *and* $\varepsilon < \|\mathbf{B}\|_F$ *is a small quantity, find its principal eigenspace, i.e.,* span$\{\mathbf{B}\}$ *or its principal eigenvalues and eigenvectors.*

Obviously, the proposed problem can be solved by a full eigendecomposition, which is computationally intensive ($O(M^3)$ flops) and difficult to implement in very large scale integrated (VLSI) hardware. There are several more efficient alternatives, e.g., the power method, subspace iteration, and the use of the Lanczos algorithm [6]. In this paper, it is shown that the Lanczos algorithm can nicely exploit the inherent structure of the matrices of interest and is more efficient in solving the stated problem

than the other alternatives. More specifically, we will show that the Lanczos algorithm requires only $O(M^2 d)$ flops to estimate the principal eigenpairs. Performance analysis is also presented to show that after d Lanczos steps, the estimation errors of the kth principal eigenvalue and eigenvector decay exponentially at the rates $\varepsilon^2/(\lambda_k - \sigma)^2$ and $\varepsilon/(\lambda_k - \sigma)$, respectively. Since the *optimal* solution (complete eigendecomposition) to the above problems is only an $O(\varepsilon)$ approximation, the analysis results show that the estimates obtained via the Lanczos algorithm achieve the same first-order approximation as those derived from the more costly full eigendecomposition.

We remark that if $\varepsilon = 0$, i.e., $\mathbf{A} = \mathbf{B} + \sigma\mathbf{I}$, then via the Lanczos algorithm, we can obtain the principal eigenspace (and also σ) exactly in a finite number of steps. This may seem surprising at first glance, but note that we are not finding the principal eigenpairs; the fact that a single repeated eigenvalue can always, in principle, be found in a finite number of steps. In other words, the task of finding the span$\{\mathbf{B}\}$ if $\varepsilon = 0$, is clearly not a conventional eigenproblem.

This paper is organized as follows. We briefly outline the *finite-step* procedure for computing the principal eigenspace of an exactly structured matrix ($\varepsilon = 0$) in the following section. This provides the motivation for a more realistic procedure given in §3, along with the appropriate analysis. Some numerical examples are given in §4.

2. Matrices with exact structure. First, we introduce the Krylov subspace

$$\mathcal{K}^m(\mathbf{A}, \mathbf{f}) = \text{span}\{\mathbf{f}, \mathbf{A}\mathbf{f}, \ldots, \mathbf{A}^{m-1}\mathbf{f}\}, \tag{6}$$

where $\mathbf{f}$ is a vector to be specified later. The important property that $\mathcal{K}^m(\mathbf{A}, \mathbf{f}) = \mathcal{K}^m(\mathbf{A} - \rho\mathbf{I}, \mathbf{f})$, for any scalar ρ, shows that (using (1))

$$\mathcal{K}^m(\mathbf{A}, \mathbf{f}) = \mathcal{K}^m(\mathbf{A} - \sigma\mathbf{I}, \mathbf{f}) = \mathcal{K}^m(\mathbf{A}, \mathbf{f}), \tag{7}$$

which we note has dimension at most $d + 1$, where $d = \text{rank } \mathbf{B}$. Suppose that $\mathbf{f}$ is not orthogonal to any of the d principal eigenvectors and to at least one of the nonprincipal eigenvectors,[1] and that the d principal eigenvalues are distinct. Then, it is not difficult to show that $\dim(\mathcal{K}^{d+1}(\mathbf{A}, \mathbf{f})) = d + 1$ [18], [19]. The Lanczos method is an efficient way of finding an orthonormal basis $\mathbf{Q}_m = [\mathbf{q}_1, \ldots, \mathbf{q}_m]$ for $\mathcal{K}^m(\mathbf{A}, \mathbf{f})$ as follows:

Given $\mathbf{A}$ (Hermitian); $\mathbf{r}_0 = \mathbf{q}_1$ (unit-norm); $j = 0$
while $\beta_j \neq 0$
$\mathbf{q}_{j+1} = \mathbf{r}_j/\beta_j$; $j := j + 1$; $\alpha_j = \mathbf{q}_j^H \mathbf{A}\mathbf{q}_j$
$\mathbf{r}_j = \mathbf{A}\mathbf{q}_j - \alpha_j\mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1}$; $\beta_j = \|\mathbf{r}_j\|$
end

Since $\dim(\mathcal{K}^{d+1}(\mathbf{A}, \mathbf{f})) = d + 1$, it turns out that the Lanczos algorithm will terminate at the $(d+1)$th Lanczos step, i.e., β_{d+1} will be zero. This early termination determines d. Furthermore, we can show that the $(d+1) \times (d+1)$ (tridiagonal) matrix $\mathbf{T}_{d+1} = \mathbf{Q}_{d+1}^H \mathbf{A}\mathbf{Q}_{d+1}$ has the eigenvalues $\{\lambda_1, \ldots, \lambda_d, \sigma\}$, where $\{\lambda_1, \ldots, \lambda_d\}$ are the principal eigenvalues of $\mathbf{A}$. Therefore, if an eigendecomposition is performed on this matrix, we can exactly obtain the principal eigenvalues $\{\lambda_i\}$ and $\mathbf{Q}_{d+1}\mathbf{s}_1, \ldots, \mathbf{Q}_{d+1}\mathbf{s}_d$ will be the principal eigenvectors of $\mathbf{A}$, where $\{\mathbf{s}_k\}$ are the eigenvectors of $\mathbf{T}_{d+1}$.

Now, an exact eigendecomposition requires an infinite amount of computation. But in our special case, it is possible to exactly find the principal eigenspace in a finite

[1] If $\mathbf{f}$ is randomly selected, it satisfies this condition with probability one.

number of steps. One way of seeing this is to note that since

$$\text{(8)} \qquad \mathrm{Tr}(\mathbf{A}) = \sum_{k=1}^{M} \lambda_k = \sum_{k=1}^{d} \lambda_k + (M-d)\sigma, \qquad \mathrm{Tr}(\mathbf{T}_{d+1}) = \sum_{k=1}^{d} \lambda_k + \sigma,$$

thus

$$\text{(9)} \qquad \sigma = \frac{\mathrm{Tr}(\mathbf{A}) - \mathrm{Tr}(\mathbf{T}_{d+1})}{M-d-1}.$$

Knowing σ, the principal eigenspace will be the range space of any d independent columns of $\mathbf{A} - \sigma\mathbf{I} = \mathbf{B}$. However, an orthonormal basis can also be easily obtained. Since σ is an eigenvalue of $\mathbf{T}_{d+1}$, we can solve $(\mathbf{T}_{d+1} - \sigma\mathbf{I})\mathbf{s} = \mathbf{0}$ and find the unit-norm vector $\mathbf{s}$. Then $\mathbf{Qs}$ is orthogonal to the principal eigenspace. Let us form a Householder matrix $\mathbf{H}$ that transforms $\mathbf{s}$ to $\mathbf{l}_{d+1}$, where $\mathbf{l}_{d+1}$ has zero elements everywhere except the last one, which is one. Then the first d column of $\mathbf{QH}$ is orthogonal to $\mathbf{Qs}$ and forms an orthonormal basis of $\mathbf{B}$.

However, if some principal eigenvalues are repeated, the Lanczos algorithm terminates before $d+1$ steps and (9) is no longer valid. So we must work a little harder to find σ. Suppose that the Lanczos algorithm stops at the mth step ($m < d+1$), yielding $\mathbf{AQ}_m = \mathbf{Q}_m\mathbf{T}_m$, where m is the number of distinct eigenvalues of $\mathbf{A}$. In this case, $\mathbf{T}_m$ no longer contains all the principal eigenvalues. To handle this, we can select another initial vector $\mathbf{f}_1$ that is orthogonal to $\mathbf{Q}_m$ and apply the Lanczos algorithm to the deflated matrix $\mathbf{A} - \mathbf{Q}_m\mathbf{T}_m\mathbf{Q}_m^H$. Proceeding in this way, we get orthogonal $\mathbf{Q}_{m_1}^{(1)}$, $\mathbf{Q}_{m_2}^{(2)}, \ldots$ to block tridiagonalize $\mathbf{A}$, i.e.,

$$\text{(10)} \qquad \cdots \mathbf{Q}_{m_2}^{(2)}\mathbf{Q}_{m_1}^{(1)}\mathbf{A}\mathbf{Q}_{m_1}^{(1)H}\mathbf{Q}_{m_2}^{(2)H}\cdots = \begin{bmatrix} \mathbf{T}_{m_1} & 0 & \cdots \\ 0 & \mathbf{T}_{m_2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix},$$

where each $\mathbf{T}_{m_i}$ is a tridiagonal matrix of size m_i equal to the number of steps in the ith application of the Lanczos algorithm. We can proceed with the above deflation till a certain $\mathbf{T}_{m_i}$ is of block size one. Then we have exhausted all the principal eigenvalues of $\mathbf{A}$ and the only element of $\mathbf{T}_{m_i}$ must be σ. Knowing σ, we can easily find a basis of the principal eigenspace from $\mathbf{A} - \sigma\mathbf{I}$. In this case, the total number of Lanczos steps is less than $d + d'$, requiring less than $O(M^2(d+d'))$ flops, where $d'(\leq d)$ is the maximum multiplicity among all the principal eigenvalues.

3. Performance analysis of the Lanczos algorithm. In reality, the matrices under consideration do not have *exactly* repeated eigenvalues, so that a more practical problem is to estimate the eigenpairs of a matrix of the form

$$\text{(11)} \qquad \mathbf{A} = \mathbf{B} + \sigma\mathbf{I} + O(\varepsilon).$$

We can still apply the Lanczos approach given in §2 to estimate the principal eigenspace. However, as is made clear in the following, though the corresponding estimation error is of the same order as for the approximation error of an exact eigendecomposition, it may have a larger first-order coefficient. However, if we modify the above algorithm and adopt the well-known Rayleigh–Ritz (RR) approximation [12], we show that the error between the kth RR value and eigenvalue λ_k is $O((\varepsilon/(\lambda_k - \sigma))^{2(m-d)})$, while the error between the kth RR vector and eigenvector

$\mathbf{e}_k$ is $O((\varepsilon/(\lambda_k - \sigma))^{m-d})$, where $k = 1, \ldots, d$ and $m(> d)$ is the number of Lanczos steps. If $m = d + 2$, we can see that the errors are $O(\varepsilon)$, and the Lanczos algorithm achieves the same first-order approximation as an exact eigendecomposition. If $m(\geq d+2)$ Lanczos steps are executed, we can achieve the same $(m-d-1)$th order approximation as the eigendecomposition.

3.1. Existing error bounds.

DEFINITION 3.1. *For the matrix* $\mathbf{A}$ *and its Krylov subspace* $\mathcal{K}^m(\mathbf{A}, \mathbf{f})$ *the* RR *values* θ_i *and vectors* $\mathbf{y}_i$ *are defined such that*

$$\mathbf{A}\mathbf{y}_i - \theta_i \mathbf{y}_i \perp \mathcal{K}^m(\mathbf{A}, \mathbf{f}). \tag{12}$$

In general, it is very difficult to estimate the errors between the eigenpairs and the corresponding RR pairs. The only existing error bounds are stated below [12].

Saad bound: *For* $j = 1, \ldots, m$,

$$|\theta_j - \lambda_j| \leq |\lambda_M - \lambda_j| \left[\frac{\sin\_(\mathbf{f}, \mathcal{E}^j)}{\cos\_(\mathbf{f}, \mathbf{e}_j)} \cdot \frac{\prod_{l=1}^{j-1} \left(\frac{\theta_l - \lambda_M}{\theta_l - \lambda_j} \right)}{T_{m-j}(1 + 2\gamma_{j,j+1,M})} \right]^2 \tag{13}$$

and

$$\tan\_(\mathbf{e}_j, \mathcal{K}^m(\mathbf{A}, \mathbf{f})) \leq \frac{\sin\_(\mathbf{f}, \mathcal{E}^j)}{\cos\_(\mathbf{f}, \mathbf{e}_j)} \cdot \frac{\prod_{l=1}^{j-1} \left(\frac{\lambda_l - \lambda_M}{\lambda_l - \lambda_j} \right)}{T_{m-j}(1 + 2\gamma_{j,j+1,M})}, \tag{14}$$

where $\gamma_{j,j+1,M} = (\lambda_j - \lambda_{j+1})/(\lambda_{j+1} - \lambda_M)$, $\mathcal{E}^j = \operatorname{span}\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_j\}$, *and* $T_{m-j}(\cdot)$ *is the Chebyshev polynomial.*

Kaniel bound: *For* $j = 1, \ldots, m$,

$$|\theta_j - \lambda_j| \leq |\lambda_M - \lambda_j| \left[\frac{\sin\_(\mathbf{f}, \mathcal{E}^j)}{\cos\_(\mathbf{f}, \mathbf{e}_j)} \cdot \frac{\prod_{l=1}^{j-1} \left(\frac{\lambda_l - \lambda_M}{\lambda_l - \lambda_j} \right)}{T_{m-j}(1 + 2\gamma_{j,j+1,M})} \right]^2 + \sum_{l=1}^{j-1} (\lambda_M - \lambda_l) \sin^2\_(\mathbf{y}_l, \mathbf{e}_l), \tag{15}$$

and

$$\sin^2\_(\mathbf{y}_l, \mathbf{e}_l) \leq \left[(\theta_l - \lambda_l) + \sum_{s=1}^{l-1} (\lambda_{l+1} - \lambda_s) \sin^2\_(\mathbf{y}_s, \mathbf{e}_s) \right] / (\lambda_{l+1} - \lambda_l). \tag{16}$$

However, these bounds are not good enough for our purposes. For one thing, they require knowledge of *both* the eigenpairs $\{\lambda_k, \mathbf{e}_k\}$ and the RR pairs $\{\theta_k, \mathbf{s}_k\}$. Second, the bounds are not tight enough. For example, when $\varepsilon = 0$, we know from §2 that the error is zero, while the bounds are obviously nonzero.

It is very difficult to derive a tighter error bound since, as is revealed in the following, the relationship between RR values and the eigenvalues are highly nonlinear. Now, it is known (see [12]) that by exploiting more knowledge of the eigenvalue distribution, one can get a tighter error bound.

The improved bound: If there are significant gaps between $\{\lambda_1, \ldots, \lambda_{k-1}\}$ *and* $\{\lambda_k, \ldots, \lambda_l\}$ *and between* $\{\lambda_k, \ldots, \lambda_l\}$ *and* $\{\lambda_{l+1} \ldots \lambda_M\}$, *then* $T_{m-j}(1 + 2\gamma_{j,j+1,M})$ *in the Kaniel and Saad bounds can be replaced by*

$$\prod_{\mu=j+1}^{k-1} \left(\frac{\lambda_\mu - \lambda_j}{\lambda_l - \lambda_\mu}\right) \prod_{\nu=l+1}^{M} \left(\frac{\lambda_\nu - \lambda_j}{\lambda_\nu - \lambda_k}\right) T_{M-j-(l-k)}(1 + 2\gamma_{jkl}), \tag{17}$$

where $j < k < l$ *and* $\gamma_{jkl} \triangleq (\lambda_j - \lambda_k)/(\lambda_k - \lambda_l)$.

For the matrix of interest in this paper, there is a gap between $\{\lambda_1, \ldots, \lambda_d\}$ and $\{\lambda_{d+1}, \ldots, \lambda_M\}$. In this case, $k = d + 1$, $l = M$, and for $j \leq d$, $\gamma_{jkl} = (\lambda_j - \lambda_{d+1})/(\lambda_{d+1} - \lambda_M)$. We know that $|\lambda_{d+1} - \lambda_M|$ is significantly smaller than $(\lambda_j - \lambda_{d+1})$. Therefore, the gap ratio γ_{jkl} is much larger than its counterparts in the Kaniel and Saad bounds; hence the improved bound for this special matrix is tighter than the Kaniel and Saad bounds. However, it is still difficult to know how tight the error bound is. As with the Kaniel and Saad bounds, the improved error bound is also a function of the *unknown* RR values $\{\theta_k\}$ and it is difficult to gain more insight from the fairly complicated expression (e.g., Chebychev polynomial).

In the following, we introduce a *new* approach to obtain a tighter and more useful error estimate of the RR approximation that also determines the speed of convergence. Although such an error estimate is not a strict bound, it shows the order of the error (in terms of ε), which can be important in, for example, determining the asymptotic performance of certain signal processing algorithms [17].

3.2. Convergence analysis and error estimate.

THEOREM 3.2. *Let* $\lambda_1 > \lambda_2 > \cdots > \lambda_M$ *be the eigenvalues of a Hermitian matrix* $\mathbf{A}$. *Suppose that the* $M - d$ *smaller eigenvalues cluster in a small region with center* $\sigma = (\lambda_{d+1} + \lambda_M)/2$, *while the principal eigenvalues are reasonably far away from* σ, *i.e.,*

$$\lambda_d - \sigma > \varepsilon, \qquad \lambda_{d+1} - \lambda_M < \varepsilon. \tag{18}$$

Let $(\theta_i, \mathbf{y}_i)$ *be the* RR *pairs from* $\mathcal{K}^m(\mathbf{A}, \mathbf{f})$, *arranged in descending order according to* θ_i, $i = 1, 2, \ldots, m$, *and* $m \geq d + 1$. *Then the error of the* k*th* RR *approximation is of the order* $O((\varepsilon/(\lambda_k - \sigma))^{2(m-d)})$, *for* $1 \leq k \leq d$, *More specifically, for* $k = 1, 2, \ldots, d$,

$$\frac{\lambda_k - \theta_k}{\lambda_k - \sigma} \sim \left\{ \frac{\sin^2 \angle(\mathcal{E}^d, \mathbf{f})}{\cos^2 \angle(\mathbf{e}_k, \mathbf{f})} \prod_{\substack{i=1 \\ i \neq k}}^{d} \frac{(\lambda_i - \sigma)^2}{(\lambda_i - \lambda_k)^2} \right\} \cdot \left(\frac{\varepsilon}{\lambda_k - \sigma}\right)^{2(m-d)}, \tag{19}$$

$$\sin^2 \angle(\mathbf{y}_k, \mathbf{e}_k) \sim \left\{ \frac{\sin^2 \angle(\mathcal{E}^d, \mathbf{f})}{\cos^2 \angle(\mathbf{e}_k, \mathbf{f})} \prod_{\substack{i=1 \\ i \neq k}}^{d} \frac{(\lambda_i - \sigma)^2}{(\lambda_i - \lambda_k)^2} \right\} \cdot \left(\frac{\varepsilon}{\lambda_k - \sigma}\right)^{2(m-d)}, \tag{20}$$

where $\mathcal{E}^d = \operatorname{span}\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d\}$.

First, we need to clarify the meaning of the symbol $\sim$ introduced in (19) and (20). By $a \sim b$, we mean (i) $a, b = O(\varepsilon^k)$; (ii) $\lim_{\varepsilon \to 0} a\varepsilon^{-k} \leq \lim_{\varepsilon \to 0} b\varepsilon^{-k}$. In other words, $a \sim b$ means that the term in a with the highest power of ε is no larger than its counterpart in b.

The proof of Theorem 3.2 is quite complicated and is described in Appendix A. Its main ideas can be summarized as follows. First, by definition, the RR pairs obey (see [12]) $\mathbf{A}\mathbf{y}_i - \theta_i \mathbf{y}_i \perp \mathcal{K}^m(\mathbf{A}, \mathbf{f}) = \mathcal{K}^m(\mathbf{A} - \sigma\mathbf{I})$ or, equivalently,

$$(\mathbf{A}_\mathbf{s}^j \mathbf{f})^H (\mathbf{A}_\mathbf{s}\mathbf{y}_i - \rho_i \mathbf{y}_i) = \mathbf{0}, \qquad j = 1, 2, \ldots, m, \tag{21}$$

where $\mathbf{A}_\mathbf{s} = \mathbf{A} - \sigma\mathbf{I}$ and $\rho_i = \theta_i - \sigma$. Using the fact that $\mathbf{y}_i = \pi_k(\mathbf{A}_\mathbf{s})\mathbf{f} / \|\pi_k(\mathbf{A}_\mathbf{s})\mathbf{f}\|$ (by a corollary in [12, p. 240]), we can obtain m independent equations for $\theta_1, \ldots, \theta_m$, where the polynomial

$$\pi_k(x) = \prod_{\substack{i=1 \\ i=k}}^{m} (x - \theta_i).$$

By the Cauchy interlace theorem [12], for $k \geq d+1$, θ_k lies between λ_{d+1} and λ_M, or by (18) $\theta_k - \sigma \sim \varepsilon$ for $k \geq d+1$. Then, let us keep the terms with the lowest power of ε and drop all the higher power terms. After some algebra, we end up with a set of Vandermonde equations from which we can recursively find the explicit expressions of $\lambda_1 - \theta_1$, $\lambda_2 - \theta_2, \ldots$. Based on this and the relationship between eigenvalues and eigenvectors, we obtain the error estimate of the principal eigenvectors (20).

A referee suggested an alternative method of proof that is listed in Appendix B. Extending the results in this proof, we can also obtain the following result, where we have strict upper bounds. For our further arguments, Theorem 3.2 is more useful, but Theorem 3.3 has independent value.

THEOREM 3.3. *For the same conditions and notation as defined in Theorem* 3.2, *we claim*

$$\lambda_k - \theta_k < \left\{ \frac{\sum_{j=d+1}^{M} \cos^2\angle(\mathbf{e}_j, \mathbf{f})(\lambda_k - \lambda_j) \prod_{\substack{i=1 \\ i=k}}^{d} (\lambda_j - \lambda_i)^2}{\cos^2\angle(\mathbf{e}_k, \mathbf{f}) \prod_{i=1}^{k-1} (\lambda_k - \theta_i)^2 \cdot \prod_{i=k}^{d} (\lambda_i - \lambda_k)^2} \right\} \cdot \left(\frac{\varepsilon}{\lambda_k - \sigma} \right)^{2(m-d)}, \tag{22}$$

$$\sin^2\angle(\mathbf{y}_k, \mathbf{e}_k) < \frac{\sum_{\substack{j=1 \\ j=k}}^{M} \cos^2\angle(\mathbf{e}_j, \mathbf{f}) \prod_{\substack{i=1 \\ i=k}}^{m} (\lambda_j - \theta_i)^2}{\cos^2\angle(\mathbf{e}_k, \mathbf{f}) \prod_{i=1}^{k-1} (\theta_i - \lambda_k)^2 \prod_{i=k+1}^{d} (\lambda_k - \lambda_i)^2}. \tag{23}$$

3.3. Other relevant results. We can use the same approach to derive the following results, whose proof is omitted.

COROLLARY 3.4. *It holds that*

$$\sin^2\angle(\mathcal{E}^d, \mathbf{y}_k) \sim \left\{ \frac{\sin^2\angle(\mathcal{E}^d, \mathbf{f})}{\cos^2\angle(\mathbf{e}_k, \mathbf{f})} \prod_{\substack{j=1 \\ j=k}}^{d} \frac{(\lambda_j - \sigma)^2}{(\lambda_k - \lambda_j)^2} \right\} \cdot \left(\frac{\varepsilon}{\lambda_k - \sigma} \right)^{2(m-d)}, \tag{24}$$

$$\gamma_k = \|\mathbf{A}\mathbf{y}_k - \theta_k \mathbf{y}_k\|^2$$

$$(25) \qquad \sim (\lambda_k - \sigma)^2 \left\{ \frac{\sin^2 \angle(\mathcal{E}^d, \mathbf{f})}{\cos^2 \angle(\mathbf{e}_k, \mathbf{f})} \prod_{\substack{j=1 \\ j \neq k}}^{d} \frac{(\lambda_j - \sigma)^2}{(\lambda_k - \lambda_j)^2} \right\} \cdot \left(\frac{\varepsilon}{\lambda_k - \sigma} \right)^{2(m-d)},$$

$$\prod_{k=d+1}^{m} \beta_k^2 \sim \sin^2 \angle(\mathcal{E}^d, \mathbf{f}) \cdot \prod_{k=1}^{d} (\lambda_k - \sigma)^2 \cdot \sum_{k=1}^{d} \left(\frac{(\lambda_k - \sigma)^{d-1}}{\prod_{\substack{j=1 \\ j \neq k}}^{d} (\lambda_k - \lambda_j)} \right)^2 \cdot \prod_{k=1}^{d} \beta_k^{-2} \cdot \varepsilon^{2(m-d)},$$

where $d+1 \leq m \leq M$, $k = 1, 2, \ldots, d$ $\mathbf{P}_{\mathcal{K}^m}^{\perp}$ *is a projection matrix that projects a vector onto the orthogonal subspace of* $\mathcal{K}^m(\mathbf{A}, \mathbf{f})$. In the exact repeated eigenvalue case, i.e., $\varepsilon = 0$, Corollary 3.4 yields $\beta_{d+1} = 0$, which is consistent with the results in §2. It is also interesting to note that the error of the RR approximation decays exponentially at the rate ε, while β_m seems to stay at the $O(\varepsilon)$ level when $m \geq d+1$. The possible reason is that $\mathbf{q}_k$ is normalized by β_k at each step.

3.4. Significance of Theorem 3.2. By Theorem 3.2, we can see that all the quantities associated with the estimation error of the eigenvalues and eigenvectors in (19), (20), (24), and (25) decay *uniformly* and *exponentially* after d Lanczos steps at the rate of $\varepsilon/(\lambda_k - \sigma)$. Note that here ε is not the smallest eigenvalue of $\mathbf{A}$ but the maximum difference between the smaller eigenvalues of $\mathbf{A}$, which is usually small for the structured matrices defined in (11). For the structured sample covariance matrix, $\varepsilon = O(1/\sqrt{N})$, where N is the number of data samples. After the $(d+2)$th Lanczos step, the errors between the RR vectors and the corresponding eigenvectors of the sample covariance matrix are $O(1/N)$ or $O(1/\sqrt{N})$. Since in this case, the principal eigenvectors are $O(1/\sqrt{N})$ approximation of the *true* basis vectors of the signal subspace, the RR vectors achieve the same first order approximation as the principal eigenvectors. In other words, $\operatorname{span}\{\mathbf{e}_1, \ldots, \mathbf{e}_d\}$ and $\operatorname{span}\{\mathbf{y}_1, \ldots, \mathbf{y}_d\}$ are *asymptotically equivalent* estimates of the signal subspace (see [18], [17] for more details).

3.5. Selection of the initial vector f. From (19)–(20) in Theorem 3.2, we see that the initial vector $\mathbf{f}$ plays an important role to achieve fast convergence. First, if $\mathbf{f}$ is orthogonal to some eigenvector, then this eigenvector will not appear in the Krylov subspace $\mathcal{K}^m(\mathbf{A}, \mathbf{f})$. However, this is not the generic case. If $\mathbf{f} \in \mathcal{E}^d$, then $\sin \angle(\mathbf{f}, \mathcal{E}^d) = 0$. In this case, after d Lanczos steps, all the errors will be zero, according to (19), (20), (24), and (25). If $\mathbf{f}$ is chosen to be very close to the principal subspace, i.e., $\sin \angle(\mathbf{f}, \mathcal{E}^d)$ is very small, the initial error will be small as well. In this case, the total error can be made small enough in a very small number of Lanczos steps.

According to (19), (20), (24), and (25), the accuracy of the kth eigenvalue and eigenvector is also related to $\cos^2 \angle(\mathbf{e}_k, \mathbf{f})$. If $\mathbf{f}$ is chosen to be close to $\mathbf{e}_k$, then $\cos^2 \angle(\mathbf{e}_k, \mathbf{f})$ is large and the error associated with the kth eigenpair estimation tends to be small. Note that $\sum_{i=1}^{d} \cos^2 \angle(\mathbf{e}_k, \mathbf{f}) = 1 - \sin^2 \angle(\mathcal{E}^d, \mathbf{f}) \leq 1$ and $\cos \angle(\mathbf{e}_i, \mathbf{f})$, $i \neq k$ can be very small. Then, the error associated with other eigenpairs can be large and require more Lanczos steps.

If we have any rough a priori information on the principal eigenpairs, we can take advantage of this information to choose an appropriate $\mathbf{f}$ to achieve a well-balanced convergence in terms of the estimation of the whole principal eigenspace in the smallest number of steps [19]. In many applications, we do have such a priori information. In

target tracking problems, the sample covariance matrix at the current time sector is close to the one at the previous time sector, whose principal eigenpairs have been estimated. These eigenpairs can be used as the a priori information to find a good initial vector $\mathbf{f}$.

3.6. Estimation of the principal eigenspace dimension. Detection schemes for estimating the dimension of the principal eigenspace are problem dependent and are very difficult to unify into a common framework. In fact, several papers [1], [3], [13], [16] appeared in various special problems, e.g., direction finding, harmonic retrieval, autoregressive moving averaging (ARMA) estimation, etc. Most of the techniques used in these papers (such as the so-called minimum description length (MDL) and an information theoretic criterion (AIC) schemes [16]) require all the M eigenvalues, which are not available at any intermediate Lanczos step. Therefore, *new* detection schemes based on the accessible information at certain Lanczos steps need to be developed. In the following, we just give a general discussion on the kind of information that can be exploited to design a proper detection scheme.

In the exact repeated eigenvalue case, $\beta_m \doteq 0$ is the criterion for estimating d. By Corollary 3.4, β_{d+1} falls below the $O(\varepsilon)$ level at the $(d+1)$th Lanczos step. Therefore, the fundamental difference between the value of β_m for $m < d+1$ and $m = d+1$ can be used as one criterion to estimate the dimension of the principal eigenspace. One approach is to compare β_m with a predetermined threshold δ_β; once $\beta_m < \delta_\beta$, $d = m-1$. In this case, the RR approximation is carried out only at the last Lanczos step, but not at the intermediate steps.

A more reliable scheme relies on the RR values at each Lanczos step. Since the smallest $M-d$ eigenvalues of $\mathbf{A}$, viz., $\lambda_{d+1}, \ldots, \lambda_M$, are close to each other, their quadratic mean and arithmetic mean should be also approximately the same. Hence the ratio of these two means should be close to one or

$$\varphi_d = \log \left(\frac{\sqrt{\frac{1}{M-d} \sum_{k=d+1}^{M} \lambda_k^2}}{\frac{1}{M-d} \sum_{k=d+1}^{M} \lambda_k} \right) \approx 0. \tag{26}$$

Clearly, if $\hat{d} < d$ or the hypothesis is wrong, $\varphi_{\hat{d}}$ is significantly larger than φ_d. Based on this fact, we can design a detection scheme by checking $\varphi_{\hat{d}} \lessgtr \gamma_{\hat{d}}$ where $\gamma_{\hat{d}}$ is a properly predetermined threshold. Now the problem is how to compute $\varphi_{\hat{d}}$ without knowing these $M - \hat{d}$ smallest eigenvalues. The trick to circumvent this difficulty is to properly use the available information, i.e., RR values, $\mathrm{Tr}(\mathbf{A})$ (Trace of $\mathbf{A}$), $\|\mathbf{A}\|_F$ (F-norm of $\mathbf{A}$). It is well known that

$$\mathrm{Tr}(\mathbf{A}) = \sum_{k=1}^{M} \lambda_k, \qquad \|\mathbf{A}\|_F^2 = \sum_{k=1}^{M} \lambda_k^2. \tag{27}$$

Since by Theorem 3.2, the d principal eigenvalues can be well approximated (to order $O(\varepsilon^{m-d})$) by their corresponding RR values from the mth Lanczos step, if $\hat{d} \leq d$,

$m \geq d+1$,

$$\varphi_{\hat{d}} \approx \log \left(\frac{\sqrt{\frac{1}{M-\hat{d}} \left(\|\mathbf{A}\|^2 - \sum_{k=1}^{\hat{d}} \theta_k^2 \right)}}{\frac{1}{M-\hat{d}} \left(\mathrm{Tr}(\mathbf{A}) - \sum_{k=1}^{\hat{d}} \theta_k \right)} \right). \tag{28}$$

It can be shown that $\varphi_{\hat{d}} = O(1)$, for $\hat{d} < d$ and $\varphi_{\hat{d}} = O(\varepsilon^2)$, when $\hat{d} = d$. To avoid any possible numerical problems, it is recommended to use only the well-converged RR values in (28) instead of all the m RR values. We can summarize the new detection scheme for the mth Lanczos step below.

NEW DETECTION SCHEME

1. Set $\hat{d} = 1$.
2. Set null hypothesis $\mathrm{H}_0 : d = \hat{d}$.
3. Evaluate $\varphi_{\hat{d}}$ using the converged RR values.
4. If $\varphi_{\hat{d}} \leq \gamma_{\hat{d}}$ accept H_0 and stop.
5. Otherwise, reject H_0; if $\hat{d} < l$, $\hat{d} := \hat{d}+1$, return to 2, where l is the number of the converged RR values. Otherwise, $m := m+1$, continue the mth Lanczos step.

As we know, the computation of $\mathrm{Tr}(\mathbf{A})$ and $\|\mathbf{A}\|_F^2$ requires about M and $M^2/2$ flops. According to [12], it takes approximately $9m^2$ flops to find the RR values (not the RR vectors) of an $m \times m$ tridiagonal matrix. Therefore, the computational complexity involved in the above detection scheme is marginal.

The last issue is how to choose the threshold $\gamma_{\hat{d}}$. It all depends on the a priori information on $\varphi_{\hat{d}}$ and the answer to this question varies from problem to problem. For example, if $\mathbf{A}$ is a sample covariance matrix as in many signal processing problems, we can show that $N(M-d)\varphi_d$ is asymptotically Chi-square ($\mathcal{X}^2$) distributed with $\frac{1}{2}(M-d)(M-d+1)-1$ degrees of freedom, where N is the length of data samples. Therefore, $\gamma_{\hat{d}}$ can be chosen based on the tail area of the $\mathcal{X}^2$ distribution function. In this case, we can also show that the above detection scheme is *strongly consistent* [17], [19].

3.7. A basic algorithm. According to (19), (20), and (25), the errors regarding the RR values and vectors, i.e., $\theta_k - \lambda_k$ and $\sin^2 \angle(\mathbf{y}_k, \mathbf{e}_k)$, converge at the same rate. Since it only takes about $9m^2$ flops to find the RR values at the mth Lanczos step versus $O(m^3)$ flops for computing the RR vectors, one can save a reasonable amount of computation by using only the RR values to check convergence at intermediate Lanczos steps. The RR vectors will be computed at the last Lanczos step after convergence is achieved.

Now, we can present a basic algorithm.

THE BASIC ALGORITHM

1. Start with a nondegenerate initial vector $\mathbf{f}$.
2. Carry out the mth Lanczos step, i.e., Steps 1–6 of the Lanczos algorithm.
3. Compute the RR values and vectors $(\theta_i^{(m)}, \mathbf{y}_i^{(m)}, \ i = 1, 2, \ldots, m)$.
4. Check the convergence of the RR values: $|\theta_k^{(m)} - \theta_k^{(m-1)}| \lesssim \delta_\theta$. If yes, store them in the set of the converged RR values.

5. Perform an appropriate detection scheme based on the converged RR values. If d is determined, then go to Step 7. Otherwise, return to Step 2 and execute the $(m+1)$th Lanczos step.
6. Compute the RR vectors and stop.

In the above algorithm, δ_θ is the threshold for θ. We present this basic algorithm only to show the basic ideas of the proposed approach. Interested readers can modify it to their problems. Obviously, if d is known, Step 6 can be skipped.

3.8. Numerical issues. According to Paige [10]–[12], when β_m and $\gamma_k^{(m)} = \|\mathbf{A}\mathbf{y}_k^{(m)} - \theta_k^{(m)}\mathbf{y}_k^{(m)}\|$ are too small at the mth Lanczos step, numerical problems may arise because the $\mathbf{q}_m$ may not be orthogonal to $\mathbf{q}_j$, $j \leq m-1$. By Corollary 3.4, for $1 \leq m \leq d$, β_m and $\gamma_k^{(m)}$ are not necessarily small until $m \geq d+1$, in which case, $\beta_{d+1}\cdots\beta_m \sim O(\varepsilon^{m-d})$ and $\gamma_k^{(m)} \sim O(\varepsilon^{m-d})$. If the initial vector does not lie too close to one specific eigenvector, the errors in all the *principal* eigenvalue and eigenvector estimates are $O(\varepsilon^{2(m-d)})$ and $O(\varepsilon^{m-d})$, respectively, at the mth Lanczos step. Therefore, once β_m and $\gamma_k^{(m)}$ become really small, the estimation of these principal eigenpairs becomes accurate enough and the algorithm should be terminated. Nevertheless, if $\mathbf{f}$ is selected improperly to be very close to one eigenvector, e.g., $\cos\angle(\mathbf{f}, \mathbf{e}_i)$ is close to one, but $\cos\angle(\mathbf{f}, \mathbf{e}_k)$ close to zero, for $k \neq i$, then by Theorem 3.2, $\gamma_i^{(m)}$ is very small but $\gamma_k^{(m)}$, $k \neq i$ is large. As discussed above, if we have some a priori information about the principal eigenvalues and eigenvectors, we can find a good initial $\mathbf{f}$ so that the convergence is well balanced and the algorithm is numerically better. At any rate, even if β_m or $\gamma_k^{(m)}$ is small but the estimation error is still above the tolerance level for various reasons, we can always alleviate the numerical problems by adopting selective reorthogonalization [12] or even complete reorthogonalization. Since $d \ll M$, there will not be a significant increase of computation load. In short, the numerical problem is not that serious in the proposed algorithm, because the algorithm usually terminates before β_k becomes very small and $d \ll M$. A large number of computer simulations also verified this point.

3.9. Computational complexity and parallel implementation. To insure the required precision of the results, we may need to carry out a few more Lanczos steps than in the repeated eigenvalue case. Nevertheless, since the error for the kth principal eigenvalues decays exponentially at the rate of $(\varepsilon/(\lambda_k - \sigma))$, there is not any significant increase of computational load compared to the finite-step algorithm in §2. Another small increase of computation arises from the calculation of the RR values for some Lanczos steps and the calculation of the RR vectors at the final step. The total cost for the RR approximations is no more than $O(\log M d + d^3)$ multiplications. Since we only calculate the RR vectors in the final step and $d \ll M$, the total computational load is essentially the same as that in the exact repeated eigenvalue case, i.e., $O(M^2 d)$ flops.

Of course, if the matrix has more structure, e.g., Toeplitz, Hankel, quasi-Toeplitz, or sparse, another order of computation reduction may be achieved. Since the covariance matrix of a stationary process is Toeplitz, Toeplitz matrices may often arise in detection and estimation problems. Due to the fact that the Toeplitz/Hankel and related matrices have a so-called displacement structure [5], the matrix-vector product can be accomplished via a fast convolution that only takes $O(M \log M)$ flops instead of $O(M^2)$. Therefore, only $O(Md \log M)$ flops in total are required for the displacement structured matrices if the proposed algorithm is employed.

TABLE 1
Errors estimated based on (19), (20), *i.e.*, $\theta_k - \lambda_k$ *and* $\sin^2 \angle(\mathbf{y}_k, \mathbf{e}_k)$.

Step	1st eigenpair	2nd eigenpair	3rd eigenpair	4th eigenpair
5	0.03603206781589	108.4226447382217	65.79516628446760	20.68879578437555
6	0.00041257928669	2.69068133137723	1.78579056097178	1.22903248875567
7	0.00000475146281	0.06715930815559	0.04874933959596	0.07343332502233
8	0.00000005432513	0.00166419298210	0.00132117538868	0.00435588642811
9	0.00000000049059	0.00003257192336	0.00002828098024	0.00020408079980
10	0.00000000000490	0.00000070498603	0.00000066946184	0.00001057365497
11	0.00000000000006	0.00000001759064	0.00000001826927	0.00000063155630
12	0.00000000000000	0.00000000030903	0.00000000035102	0.00000002655900

Obviously, the most computational intensive operation in our approach is the matrix-vector product ($O(M^2)$ flops), which is very easy to implement in parallel. If we have M or M^2 multipliers or array processors, we can reduce the computational time to $O(Md)$ or $O(d^3)$ accordingly. Moreover, the computation of the RR values and RR vectors can also be done in parallel with the calculation of $\mathbf{q}_m$.

4. Numerical examples. To give some intuitive ideas of how the Lanczos algorithm achieves fast convergence via exploitation of the matrix structure, we applied it to the following two numerical examples. For comparison, we also tried the power method and the subspace iteration method for the same matrices. In the first example, we have a 20×20 matrix with exact repeated eigenvalues. The matrix in the second example is a perturbed version of the first one. The eigenvalues of both matrices are listed below.

Matrix with repeated eigenvalues.

$\{\lambda_i\}$ = {11.2115, 9.2050, 9.0024, 7.8380, $\vdots$ 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000, 5.0000}.

Matrix with near-repeated eigenvalues.

$\{\lambda_i\}$ = {11.1178, 9.1882, 9.0093, 7.7430, $\vdots$ 5.4246, 5.4103, 5.3240, 5.2159, 5.1959, 5.1501, 5.1130, 5.0707, 5.0470, 5.0034, 4.9937, 4.9190, 4.8851, 4.8705, 4.8424, 4.7789}.

Obviously, according to the eigenvalue distribution, $d = 4$. To simplify the comparison, let us assume that we know the principal eigenspace dimension and set the convergence threshold to be 10^{-2}. We begin with the matrix with repeated eigenvalues. It took 19, 37, 19, 13 iterations for the power method to converge to the corresponding four principal eigenvectors. Similarly, the subspace iteration method required 14 iterations. Since each iteration of the subspace iteration requires four matrix-vector multiplications, reorthogonalization, and RR approximation, the total cost is more than $14 \times 4 = 56$ equivalent power iterations. Therefore, the total computational load for the power method and subspace iteration methods exceed M^3 flops and these methods may not gain much advantage over a conventional eigendecomposition. Nevertheless, as was shown in §2, the error of the Lanczos algorithm became exactly zero at the fifth or $(d+1)$th Lanczos step.

Now let us consider the perturbed matrix. For the power method, the iteration numbers were similar to those in the previous case, i.e., 21, 23, 21, 14. The subspace iteration method also required 14 iterations. In this case, we notice that the distances among these 16 eigenvalues are clustered together, i.e., $|\lambda_i - \lambda_j| \leq 5.4246 - 4.7789 =$

TABLE 2
Errors between the four principal eigenvalues and RR *values*: $\lambda_k - \theta_k$.

Step	1st eigenvalue	2nd eigenvalue	3rd eigenvalue	4th eigenvalue
5	0.00127069170465	0.11723150268288	1.24173355526221	2.43979635361673
6	0.00029638584599	0.10884784998719	1.04544880742115	0.97290479633678
7	0.00000028814013	0.00129038695552	0.00083054572480	0.00039517091821
8	0.00000000017460	0.00000167845553	0.00000116550693	0.00000119693565
9	0.00000000000018	0.00000000381801	0.00000000290035	0.00000000654457
10	-0.00000000000000	0.00000000000379	0.00000000000315	0.00000000001583
11	-0.00000000000000	0.00000000000001	0.00000000000001	0.00000000000007
12	-0.00000000000000	-0.00000000000000	0.00000000000000	-0.00000000000000

TABLE 3
Errors between the four principal eigenvectors and RR *vectors*: $\sin^2 \angle(\mathbf{y}_k, \mathbf{e}_k)$.

Step	1st eigenvector	2nd eigenvector	3rd eigenvector	4th eigenvector
5	0.00051569936262	0.64204461430267	0.98085706907710	0.99997877981132
6	0.00012468583362	0.59511588056001	0.89825111642592	0.80663424313299
7	0.00000004752066	0.00034470804766	0.00024292507197	0.00014696836596
8	0.00000000002858	0.00000040255868	0.00000029218890	0.00000044207957
9	0.00000000000003	0.00000000096714	0.00000000076980	0.00000000262875
10	0.00000000000000	0.00000000000092	0.00000000000081	0.00000000000601
11	0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000003
12	0.00000000000000	0.00000000000000	0.00000000000000	0.0000000000000

0.6457. The Lanczos algorithm was used to exploit this property. The errors $\lambda_k - \theta_k$ and $\sin^2 \angle(\mathbf{y}_k, \mathbf{e}_k)$ estimated according to (19), (20) are listed in Table 1, while the actual estimation errors are in Tables 2 and 3. By Table 1, the error estimates of Theorem 3.2 became below 10^{-4} (threshold) at the ninth step. However, the actual errors were below 10^{-6} at the eighth step. Therefore, only a few more Lanczos steps are required to remedy the perturbation according to the analytic error bound and the actual error illustrated in Tables 1, 2, and 3. In fact, this example should be a difficult case for the Lanczos algorithm since there are two close principal eigenvalues, 9.1882 and 9.0093, whose difference is even much smaller than $\epsilon = 0.6457$. Nevertheless, as shown in these tables, this difficulty could be easily overcome by three more Lanczos steps. Therefore, in most cases, the closely spaced principal eigenvalues do not pose a very serious problem since there are many more near (almost) repeated nonprincipal eigenvalues. Of course, if the principal eigenvalues are reasonably far apart, then more rapid convergence is expected.

In the following, we give more practical examples in array signal processing. Here, a 15-element uniform linear array was used to estimate the direction-of-arrival (DOA) of two uncorrelated sources from 35° and 45°, respectively. In this case, the sample covariance matrix estimated based on 500 data samples should be close to a rank-2 matrix plus shift. Standard eigendecomposition and the Lanczos-based algorithms were used to estimate the principal eigenspace. Since d was unknown, the aforementioned AIC and MDL techniques were used to detect d based on all 15 eigenvalues. Alternatively, the suggested Lanczos-based algorithm plus a new detection scheme was also carried out for comparison. The ESPRIT algorithm [14] was utilized to estimate the DOAs based on the principal eigenspace estimates from both eigendecomposition

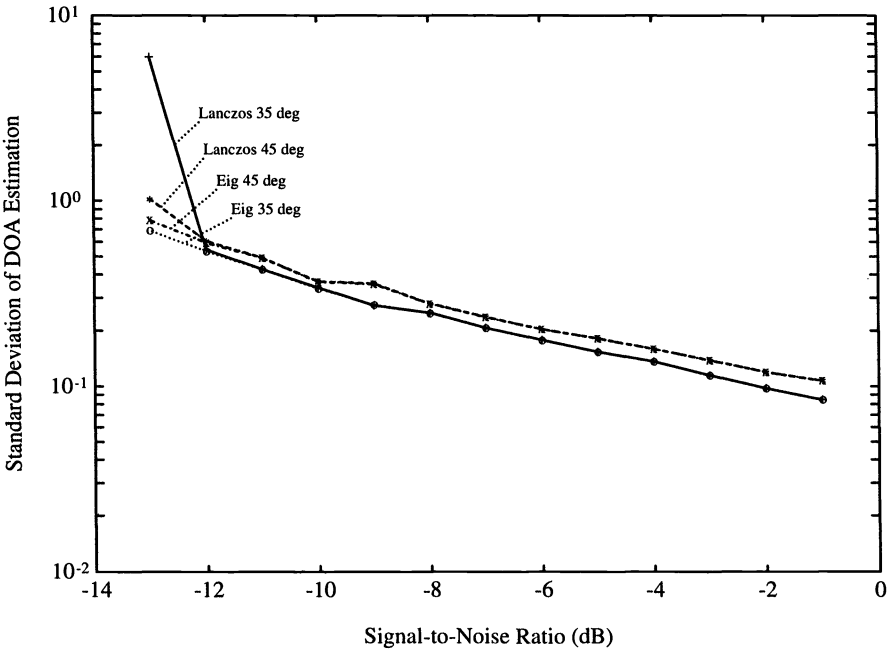


FIG. 1. *Comparison of the* MDL, AIC, *and Lanczos-based detection schemes.*

and Lanczos (with random **f**). The quality of the detection is characterized by the probability of detection based on 500 independent trials for each case, while the accuracy of the signal subspace estimation is measured by the standard deviation of the DOA estimates based on the same number of trials. Here, the signal-to-noise-ratio (SNR) varied from -20 dB to 0 dB, and the results are shown in Figs. 1 and 2. From Fig. 1, it is clear that the AIC scheme is not strongly consistent since it stays at 90% even if the SNR increases. As shown in [16] and [17], MDL and the new scheme are strongly consistent. Hence, their probability of correct detection approaches to one after SNR ≥ -14 dB, as illustrated in Fig. 1. It is interesting to note that the new schemes work a little better in the low SNR case. When SNR ≥ -14 dB, typical eigenvalue distribution of the sample covariance matrix is illustrated below:

{76.00, 65.67 ⋮ 49.91, 47.10, 45.89, 44.50, 42.50, 40.78, 39.85, 38.46, 38.40, 36.68, 34.56, 32.30, 30.10}

Therefore, the gap between the principal and nonprincipal eigenvalues is not very large and the proposed fast method still works reasonably well. For the cases where the correct detection is made, i.e., SNR ≥ -14 dB, the DOA estimates are given in Fig. 2, from which we can tell that there is no significant difference between these two methods except when SNR $= 14$ dB. However, the flop counts were also recorded and the fast approach achieves a factor of 10 computational savings on average, which is quite significant.

5. Concluding remarks. In this paper, we studied the problem of estimating the principal eigenspace or eigenpairs of certain structured matrices that can be expressed as a low-rank matrix plus a shift. Though this problem can be solved by a full eigendecomposition and several other faster alternatives, we observed that the Lanczos algorithm can exploit the inherent matrix structure and is more efficient than

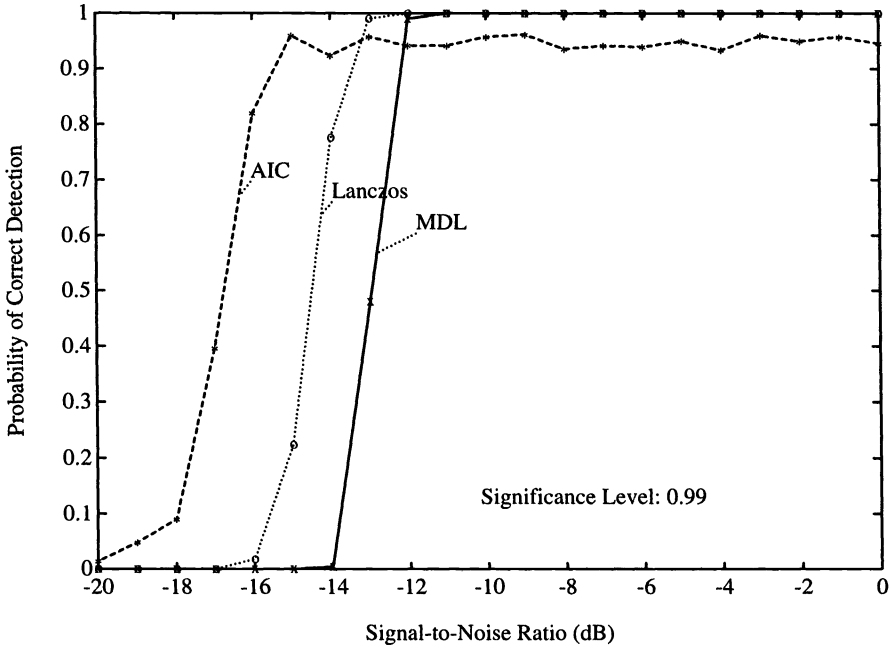


FIG. 2. *Comparison of the* DOA *estimation of eigendecomposition and Lanczos.*

most of the other existing algorithms. We first briefly described a finite-step Lanczos based algorithm to find the principal eigenspace of an exact low-rank-plus-shift matrix. A new analytic error estimate of the RR approximation for the structured matrix was then presented, which is more useful and accurate than the existing error bounds. Such an analysis also showed that the estimation error of the RR approximation decreases exponentially at a small rate as the Lanczos algorithm proceeds. We discussed several important implementation issues such as parallel computation, estimation of the dimension of the principal eigenspace, and exploitation of other matrix structure. Numerical examples were given to demonstrate the superior performance of the Lanczos-based algorithm over its alternatives.

Appendix A. Proof of Theorem 3.2. By definition, $\mathbf{A}\mathbf{y}_k - \theta_k \mathbf{y}_k \perp \mathcal{K}^m(\mathbf{A}, \mathbf{f})$. It is well known that Krylov subspace is invariant of shift, i.e., $\mathcal{K}^m(\mathbf{A}, \mathbf{f}) = \mathcal{K}^m(\mathbf{A} - \sigma\mathbf{I}, \mathbf{f})$ (see, e.g., [12]). Let $\mu_i = \lambda_i - \sigma$ and $\rho_i = \theta_i - \sigma$, where $\sigma = (\lambda_{d+1} + \lambda_M)/2$. Clearly $|\mu_i - \mu_j| = |\lambda_i - \lambda_j| < \epsilon$ and $\mu_i < \epsilon$ for $d+1 \le i, j \le M$. Also, $\mathbf{A}\mathbf{y}_k - \theta_k \mathbf{y}_k = (\mathbf{A} - \sigma\mathbf{I})\mathbf{y}_k - \rho_k \mathbf{y}_k$. Hence, it is equivalent to study the problem with $\mathbf{A}_\mathbf{s} = \mathbf{A} - \sigma\mathbf{I}$ instead of $\mathbf{A}$ (except for a shift in the eigenvalues). Starting again from the definition of the RR pair, viz., $\mathbf{A}_\mathbf{s}\mathbf{y}_k - \theta_k \mathbf{y}_k \perp \mathcal{K}^m(\mathbf{A}_\mathbf{s}, \mathbf{f})$, we obtain

$$(\mathbf{A}_\mathbf{s}^j \mathbf{f})^H (\mathbf{A}_\mathbf{s}\mathbf{y}_k - \rho_k \mathbf{y}_k) = 0, \qquad j = 0, 1, \ldots, m-1. \tag{A.1}$$

By a corollary in [12, p. 142], $\mathbf{y}_k = \pi_k(\mathbf{A}_\mathbf{s})\mathbf{f} / \|\pi_k(\mathbf{A}_\mathbf{s})\mathbf{f}\|$. Substituting into (A.1) and recalling that $\pi(\mathbf{A}_\mathbf{s}) = \pi_k(\mathbf{A}_\mathbf{s})(\mathbf{A}_\mathbf{s} - \rho_k \mathbf{I})$, gives

$$\mathbf{f}^H \mathbf{A}_\mathbf{s}^j \pi(\mathbf{A}_\mathbf{s})\mathbf{f} = \mathbf{f}^H \mathbf{A}_\mathbf{s}^j (\mathbf{A}_\mathbf{s} - \rho_k \mathbf{I}) \pi_k(\mathbf{A}_\mathbf{s})\mathbf{f} = 0. \tag{A.2}$$

Obviously, the eigendecomposition of $\mathbf{A}_\mathbf{s}$ is $\mathbf{E}\mathbf{\Lambda}_\mathbf{s}\mathbf{E}^H$, where $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M]$ and $\mathbf{\Lambda}_\mathbf{s} = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_M)$. Let us define f_k to be $|\mathbf{e}_k^H \mathbf{f}| = \|\mathbf{f}\| \cos \angle(\mathbf{e}_i, \mathbf{f})$. Then (A.2)

becomes

$$(\mathbf{E}^H\mathbf{f})^H\mathbf{\Lambda}_\mathbf{s}^j\pi(\mathbf{\Lambda}_\mathbf{s})\mathbf{E}^H\mathbf{f}=0$$

or

$$\sum_{k=1}^{M}\mu_k^j f_k^2\pi(\mu_k)=0. \tag{A.3}$$

Write out all d equations for $m-d\le j\le m-1$ and move the terms containing $\pi(\mu_i)$, $i\ge d+1$ to the right-hand side. Then

(A.4)
$$\begin{aligned}\mu_1^{m-d}f_1^2\pi(\mu_1)+\cdots+\mu_d^{m-d}f_d^2\pi(\mu_d)&=-\mu_{d+1}^{m-d}f_{d+1}^2\pi(\mu_{d+1})-\cdots-\mu_M f_M^2\pi(\mu_M),\\ \mu_1^{m-1}f_1^2\pi(\mu_1)+\cdots+\mu_d^{m-1}f_d^2\pi(\mu_d)&=-\mu_{d+1}^{m-1}f_{d+1}^2\pi(\mu_{d+1})-\cdots-\mu_M^{m-1}f_M^2\pi(\mu_M).\end{aligned}$$

Let $x_k=f_k\mu_k^{m-d}\pi(\mu_k)$, $k=1,2,\ldots,d$. Then (A.4) becomes a set of d linear equations. Now let us examine the quantities on the right-hand side of (A.4). For $k\ge d+1$, $\pi(\mu_k)=\prod_{i=1}^{d}(\mu_k-\rho_i)\prod_{i=d+1}^{m}(\mu_k-\rho_i)$. We know that the RR values ρ_i, $1\le i\le m$ are the eigenvalues of $\mathbf{Q}^H\mathbf{A}_\mathbf{s}\mathbf{Q}$. Then, according to the Cauchy interlace theorem, we have $\mu_{M-m+i}\le\rho_i\le\mu_i$. For $i\ge d+1$, $M-m+1>d+1$. Therefore $|\rho_i-\mu_k|<\max(|\mu_i-\mu_k|,|\mu_{i+M-m}-\mu_k|)<\epsilon$, $k\ge m+1\ge d+1$ by assumption. So $\pi(\mu_k)\sim o(\epsilon^{m-d})$. Also note that $|\mu_k|<\epsilon$. For $k>d$, $\mu_k^j\pi(\mu_k)\sim o(\epsilon^{m-d+j})$. Clearly, the dominant term on the right-hand side is the one with the lowest power of ϵ, i.e., $\epsilon^{2(m-d)}$, when $j=m-d$. Let us neglect the higher power terms of ϵ and write (A.4) in matrix form

$$\begin{bmatrix}1&1&\cdots&1\\ \mu_1&\mu_2&\cdots&\mu_d\\ \vdots&\vdots&\ddots&\vdots\\ \mu_1^{d-1}&\mu_2^{d-1}&\cdots&\mu_d^{d-1}\end{bmatrix}\begin{bmatrix}f_1^2\mu_1^{m-d}\pi(\mu_1)\\ f_2^2\mu_2^{m-d}\pi(\mu_2)\\ \vdots\\ f_d^2\mu_d^{m-d}\pi(\mu_d)\end{bmatrix}=\begin{bmatrix}-\sum_{i=d+1}^{M}\mu_i^{m-d}f_i^2\pi(\mu_i)\\ 0\\ \vdots\\ 0\end{bmatrix}.$$

The matrix on the left-hand side of the above equation is a Vandermonde matrix. Hence, there is a closed form solution for this set of linear equations, namely,

$$f_k^2\mu_k^{m-d}\pi(\mu_k)=-\prod_{\substack{i=1\\ i=k}}^{d}\left(\frac{\mu_i}{\mu_i-\mu_k}\right)\sum_{i=d+1}^{M}\mu_i^{m-d}f_i^2\pi(\mu_i), \tag{A.5}$$

where $k=1,2,\ldots,d$. Let us start with $k=1$. In this case, by the Cauchy interlace theorem, $\rho_i\le\mu_i$ for $i=1,2,\ldots,m$. Since $\mu_1>\mu_i$, $\mu_1-\rho_i>\mu_1-\mu_i$, for $i\ge 2$. Hence

$$\mu_1-\rho_1\le(-1)^d\frac{1}{\mu_1^{m-d}}\prod_{i=2}^{d}\frac{\mu_i}{(\mu_i-\mu_1)^2}\prod_{i=d+1}^{m}\frac{1}{\mu_1-\rho_i}\sum_{i=d+1}^{M}\mu_i^{m-d}f_i^2\pi(\mu_i). \tag{A.6}$$

Since $|\mu_i-\mu_j|<\epsilon$ for $d+1\le i,j\le M$,

$$\left|-\sum_{i=d+1}^{M}\mu_i^{m-d}f_i^2\pi(\mu_i)\right|\le\sum_{i=d+1}^{M}f_i^2\prod_{i=1}^{d}\rho_i\epsilon^{2(m-d)}\le\sum_{i=d+1}^{M}f_i^2\prod_{i=1}^{d}\mu_i\epsilon^{2(m-d)}. \tag{A.7}$$

Also, we know $\mu_1 - \rho_i = \mu_1 + O(\epsilon)$, for $d+1 \le i \le m$. Thus,

$$\text{(A.8)} \qquad \prod_{i=d+1}^{m} \frac{1}{\mu_1 - \rho_i} \left| \sum_{i=d+1}^{M} \mu_i^{m-d} f_i^2 \pi(\mu_i) \right| \sim \mu_1^{-(m-d)} \sum_{i=d+1}^{M} f_i^2 \prod_{i=1}^{d} \mu_i \epsilon^{2(m-d)}.$$

Hence,

$$\text{(A.9)} \qquad \frac{\mu_1 - \rho_1}{\mu_1} \sim \frac{\sum_{i=d+1}^{M} f_i^2}{f_1^2} \prod_{i=2}^{d} \frac{\mu_i^2}{(\mu_1 - \mu_i)^2} \left(\frac{\epsilon}{\mu_1} \right)^{2(m-d)}.$$

Let us try $k = 2$. Using the Cauchy interlace theorem again, we can still replace ρ_i by μ_i for $i > 2$. Hence,

$$\text{(A.10)} \qquad |(\mu_2 - \rho_1)(\mu_2 - \rho_2)| \sim \mu_2(\mu_2 - \mu_1) \frac{\sum_{i=d+1}^{M} f_i^2}{f_2^2} \prod_{\substack{i=1 \\ i=2}}^{m} \frac{\mu_i^2}{(\mu_2 - \mu_i)^2} \left(\frac{\epsilon}{\mu_2} \right)^{2(m-d)}.$$

From (A.9), $\mu_1 - \rho_1 = O(\epsilon/\mu_1)$ for $m \ge d+1$. Therefore, $\mu_2 - \rho_1 = (\mu_2 - \mu_1) + (\mu_1 - \rho_1) = (\mu_2 - \mu_1) + O(\epsilon/\mu_1)$ and we can replace $\mu_2 - \rho_1$ by $\mu_2 - \mu_1$ and maintain the same approximation order $((\epsilon/\mu_2)^{2(m-d)})$.

$$\text{(A.11)} \qquad \frac{\mu_2 - \rho_2}{\mu_2} \sim \frac{\sum_{i=d+1}^{M} f_i^2}{f_2^2} \prod_{\substack{i=1 \\ i=2}}^{d} \frac{\mu_i^2}{(\mu_2 - \mu_i)^2} \left(\frac{\epsilon}{\mu_2} \right)^{2(m-d)}.$$

Similarly, trying $k = 1, 2, \ldots, d$, we will be able to obtain

$$\text{(A.12)} \qquad \frac{\mu_k - \rho_k}{\mu_k} \sim \frac{\sum_{i=d+1}^{M} f_i^2}{f_k^2} \prod_{\substack{i=1 \\ i=k}}^{d} \frac{\mu_i^2}{(\mu_k - \mu_i)^2} \left(\frac{\epsilon}{\mu_k} \right)^{2(m-d)}.$$

Since $\mu_k = \lambda_k - \sigma$, $\rho_k = \theta_k - \sigma$, $f_k^2 = \|\mathbf{f}\|^2 \cos^2 \angle(\mathbf{e}_k, \mathbf{f})$, $\sum_{i=d+1}^{M} f_i^2 = \|\mathbf{f}\|^2 - \sum_{i=1}^{d} f_i^2 = \|\mathbf{f}\|^2(1 - \cos^2 \angle(\mathcal{E}^d, \mathbf{f})) = \|\mathbf{f}\|^2 \sin^2 \angle(\mathcal{E}^d, \mathbf{f})$

$$\text{(A.13)} \qquad \frac{\lambda_k - \theta_k}{\lambda_k - \sigma} \sim \left\{ \frac{\sin^2 \angle(\mathcal{E}^d, \mathbf{f})}{\cos^2 \angle(\mathbf{e}_k, \mathbf{f})} \prod_{\substack{i=1 \\ i=k}}^{d} \frac{(\lambda_i - \sigma)^2}{(\lambda_k - \lambda_i)^2} \right\} \cdot \left(\frac{\epsilon}{\lambda_k - \sigma} \right)^{2(m-d)}.$$

Now let us evaluate the error between the eigenvectors and RR vectors, i.e., the angle between the $\mathbf{e}_k$ and $\mathbf{y}_k$, $k = 1, 2, \ldots, d$

$$\text{(A.14)} \qquad \cos^2 \angle(\mathbf{e}_k, \mathbf{y}_k) = |\mathbf{e}_k^H \mathbf{y}_k|^2 = \frac{|\mathbf{e}_k^H \pi_k(\mathbf{A_s})\mathbf{f}|^2}{\|\pi_k(\mathbf{A_s})\mathbf{f}\|^2}.$$

Since $\mathbf{e}_k^H \mathbf{A_s} = \mu_k \mathbf{e}_k^H$, it is easy to see that

$$\|\mathbf{e}_k^H \pi_k(\mathbf{A_s})\mathbf{f}\|^2 = \|\mathbf{e}_k^H \pi_k(\mu_k)\mathbf{f}\|^2 = \pi_k(\mu_k)^2 f_k^2.$$

Write $\mathbf{A_s}$ in terms of its eigenvalues and eigenvectors, i.e., $\mathbf{A_s} = \mathbf{E}\Lambda_\mathbf{s}\mathbf{E}^H$. Then, $\pi_k(\mathbf{A_s})\mathbf{f} = \mathbf{E}\pi_k(\Lambda_\mathbf{s})\mathbf{E}^H\mathbf{f}$. Thus,

$$\|\pi_k(\mathbf{A_s})\mathbf{f}\|^2 = \sum_{i=1}^{M} \pi_k(\mu_i)^2 f_i^2,$$

$$\text{(A.15)}\quad \sin^2{}_-(\mathbf{e}_k, \mathbf{y}_k) = 1 - \cos^2{}_-(\mathbf{e}_k, \mathbf{y}_k) = 1 - \frac{\pi_k^2(\mu_k) f_k^2}{\sum_{i=1}^{M} \pi_k^2(\mu_i) f_i^2} = \frac{\sum_{\substack{i=1 \\ i=k}}^{M} \pi_k^2(\mu_i) f_i^2}{\sum_{i=1}^{M} \pi_k^2(\mu_i) f_i^2}.$$

Let us examine $\pi_k^2(\lambda_i)$ in three cases: (i) $i \le d$, $i \ne k$; (ii) $i = k$; (iii) $i \ge d+1$. Starting with case (i), by (A.5), we have

(A.16)

$$\pi_k^2(\mu_i) f_i^2 = \frac{\pi^2(\mu_i)}{(\mu_i - \rho_k)^2} f_i^2 = \prod_{\substack{j=1 \\ j=i}}^{d} \frac{\mu_j^2}{(\mu_j - \mu_i)^2} \cdot \left(\sum_{j=d+1}^{M} f_j^2 \pi(\mu_j) \frac{\mu_j^{m-d}}{\mu_i^{m-d}} \right)^2 f_i^{-2} (\mu_i - \rho_k)^{-2}$$

$$\sim \frac{\left(\sum_{j=d+1}^{M} f_j^2 \right)^2 \mu_i^{2(m-d)}}{f_i^2 (\mu_k - \mu_i)^2} \prod_{\substack{j=1 \\ j=i}}^{d} \frac{\mu_j^4}{(\mu_j - \mu_i)^2} \cdot \left(\frac{\epsilon}{\mu_i} \right)^{4(m-d)}.$$

In case (ii), i.e., $i = k$.

$$\text{(A.17)}\qquad \pi_k^2(\mu_k) f_k^2 = \prod_{\substack{j=1 \\ j=k}}^{m} (\mu_k - \rho_j)^2 f_k^2 \sim \mu_k^{2(m-d)} f_k^2 \prod_{\substack{j=1 \\ j=k}}^{d} (\mu_k - \mu_j)^2.$$

In case (iii), viz., $i \ge d+1$, since $\mu_i = O(\epsilon)$ and $\rho_j = \mu_j + O(\epsilon)$, for $1 \le j \le d$, thus

$$\pi_k^2(\mu_i) f_i^2 = \prod_{\substack{j=1 \\ j=k}}^{m} (\mu_i - \rho_j)^2 f_i^2 = f_i^2 \prod_{\substack{j=1 \\ j=k}}^{d} (\mu_i - \rho_j)^2 \prod_{j=d+1}^{m} (\mu_i - \rho_j)^2$$

(A.18)

$$\sim f_i^2 \mu_k^{2(m-d)} \left(\prod_{\substack{j=1 \\ j=k}}^{d} \mu_j^2 \right) \cdot \left(\frac{\epsilon}{\mu_k} \right)^{2(m-d)}.$$

Hence, the term in case (ii) is the dominant term of the sum $\sum_{i=1}^{M} \pi_k^2(\mu_i) f_i^2$, i.e.,

$$\text{(A.19)}\qquad \sum_{i=1}^{M} \pi_k^2(\mu_i) f_i^2 \sim \mu_k^{2(m-d)} f_k^2 \prod_{\substack{j=1 \\ j=k}}^{d} (\mu_k - \mu_j)^2.$$

Nevertheless, since the numerator of (A.15) does not have this term (A.17), its smallest power of ϵ is then $2(m-d)$ corresponding to case (iii). Hence,

$$\sum_{\substack{i=1\\ i=k}}^{M} \pi_k^2(\mu_i) f_i^2 \sim \sum_{i=d+1}^{M} f_i^2 \left(\prod_{\substack{j=1\\ j=k}}^{d} \mu_j^2 \right) \mu_k^{2(m-d)} \cdot \left(\frac{\epsilon}{\mu_k} \right)^{2(m-d)}. \tag{A.20}$$

Therefore, by (A.15), we obtain

$$\sin^2\_(\mathbf{e}_k, \mathbf{y}_k) \sim \frac{\sum_{i=d+1}^{M} f_i^2}{f_k^2} \prod_{\substack{j=1\\ j=k}}^{d} \frac{\mu_j^2}{(\mu_k - \mu_j)^2} \cdot \left(\frac{\epsilon}{\mu_k} \right)^{2(m-d)}. \tag{A.21}$$

Since $\sum_{i=d+1}^{M} f_i^2 = \|\mathbf{f}\|^2 \sin^2\_(\mathcal{E}^d, \mathbf{f})$ and $f_k^2 = \|\mathbf{f}\|^2 \cos^2\_(\mathbf{e}_k, \mathbf{f})$,

$$\sin^2\_(\mathbf{e}_k, \mathbf{y}_k) \sim \left\{ \frac{\sin^2\_(\mathcal{E}^d, \mathbf{f})}{\cos^2\_(\mathbf{e}_k, \mathbf{f})} \prod_{\substack{j=1\\ j=k}}^{d} \frac{(\lambda_j - \sigma)^2}{(\lambda_k - \lambda_j)^2} \right\} \cdot \left(\frac{\epsilon}{\lambda_k - \sigma} \right)^{2(m-d)}. \tag{A.22}$$

□

Appendix B. Another proof of Theorem 3.2. For $m \geq d+1$, let us create the $(m-1)$th order polynomials $p_k(x)$, $1 \leq k \leq d$:

$$p_k(x) = \prod_{j=1}^{k-1} (x - \theta_j) \prod_{j=k+1}^{d} (x - \lambda_j) \cdot (x - \sigma)^{m-d}. \tag{B.23}$$

It is clear that $p_k(\mathbf{A})\mathbf{f} \in \mathcal{K}^m(\mathbf{A}, \mathbf{f})$. Since $\{\theta_j\}_{j=1}^{k-1}$ are roots of $p_k(x)$, then as shown in [12], $\mathbf{t}_k = p_k(\mathbf{A})\mathbf{f} \perp \mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$. Clearly, $(\theta_i, \mathbf{y}_i)$, $1 \leq i \leq m$ are the eigenpairs corresponding the projection of $\mathbf{A}$ onto $\mathcal{K}^m(\mathbf{A}, \mathbf{f})$. Since $\mathbf{t}_k \in \mathcal{K}^m(\mathbf{A}, \mathbf{f})$ and $\mathbf{t}_k \perp \mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$, it is not difficult to show that the Rayleigh quotient $\rho(\mathbf{A}, \mathbf{t}_k) = \mathbf{t}_k^H \mathbf{A} \mathbf{t}_k / \|\mathbf{t}_k\|^2 < \theta_k$ (see e.g., [12]). Then,

$$\lambda_k - \theta_k < \lambda_k - \rho(\mathbf{A}, \mathbf{t}_k) = \rho(\lambda_k \mathbf{I} - \mathbf{A}, \mathbf{t}_k). \tag{B.24}$$

Let us write $\mathbf{A} = \sum_{i=1}^{M} \lambda_i \mathbf{e}_i \mathbf{e}_i^H$ and define $f_i = |\mathbf{e}_i^H \mathbf{f}| = \cos\_(\mathbf{e}_i, \mathbf{f})$. It is not difficult to show that

$$\|\mathbf{t}_k\|^2 = \sum_{i=1}^{M} p_k^2(\lambda_i) f_i^2, \qquad \mathbf{t}_k^H (\lambda_k \mathbf{I} - \mathbf{A}) \mathbf{t}_k = \sum_{i=1}^{M} p_k^2(\lambda_i)(\lambda_k - \lambda_i) f_i^2, \tag{B.25}$$

and

$$\rho(\lambda_k \mathbf{I} - \mathbf{A}, \mathbf{t}_k) = \frac{\sum_{i=1}^{M} p_k^2(\lambda_i)(\lambda_k - \lambda_i) f_i^2}{\sum_{i=1}^{M} p_k^2(\lambda_i) f_i^2} \leq \frac{\sum_{i=k+1}^{M} p_k^2(\lambda_i)(\lambda_k - \lambda_i) f_i^2}{p_k^2(\lambda_k) f_k^2} \tag{B.26}$$

$$= \frac{\sum_{i=d+1}^{M} p_k^2(\lambda_i)(\lambda_k - \lambda_i) f_i^2}{p_k^2(\lambda_k) f_k^2}. \tag{B.27}$$

The inequality in (B.26) is due to the fact that $\lambda_k - \lambda_i \leq 0$ for $i < k$ and $p_k^2(\lambda_i) f_i^2 \geq 0$. The equality from (B.26) to (B.27) results from the fact that $p_k(\lambda_i) = 0$ for $k+1 \leq i \leq d$, as these λ_i's are the roots of $p_k(x)$ according to (B.23).

Let us start with $k = 1$. Combining (B.24)–(B.27), we obtain

$$\text{(B.28)}\quad \lambda_1 - \theta_1 < \frac{\sum_{i=d+1}^{M} f_i^2 (\lambda_1 - \lambda_i) \cdot (\lambda_i - \lambda_2)^2 \cdots (\lambda_i - \lambda_d)^2 (\lambda_i - \sigma)^{2(m-d)}}{f_1^2 (\lambda_1 - \lambda_2)^2 \cdots (\lambda_1 - \lambda_d)^2 (\lambda_1 - \sigma)^{2(m-d)}}$$

$$\text{(B.29)}\quad < \frac{\sum_{i=d+1}^{M} f_i^2 (\lambda_i - \lambda_1) \cdot (\lambda_i - \lambda_2)^2 \cdots (\lambda_i - \lambda_d)^2}{f_1^2 (\lambda_1 - \lambda_2)^2 \cdots (\lambda_1 - \lambda_d)^2} \cdot \left(\frac{\varepsilon}{\lambda_1 - \sigma} \right)^{2(m-d)}.$$

The inequality in (B.29) holds because $|\lambda_i - \sigma| < \varepsilon$, $d+1 \leq i \leq M$. For the same reason, we can replace the λ_i in (B.29) by σ without affecting the order $\varepsilon^{2(m-d)}$ term. Realizing that $f_i^2 = \cos^2 \angle(\mathbf{e}_i, \mathbf{f})$ and $\sum_{i=d+1}^{M} f_i^2 = \sin^2 \angle(\mathbf{f}, \mathcal{E}^d)$, we can easily obtain (19) for $k = 1$. Let us consider the case: $k = 2$. Then (B.27) yields

(B.30)

$$\lambda_2 - \theta_2 < \frac{\sum_{i=d+1}^{M} f_i^2 (\lambda_i - \theta_1)^2 (\lambda_2 - \lambda_i)(\lambda_i - \lambda_3)^2 \cdots (\lambda_i - \lambda_d)^2}{f_2^2 (\lambda_2 - \theta_1)^2 (\lambda_2 - \lambda_3)^2 \cdots (\lambda_2 - \lambda_d)^2} \cdot \left(\frac{\varepsilon}{\lambda_2 - \sigma} \right)^{2(m-d)}$$

$$< \frac{\sum_{i=d+1}^{M} f_i^2 (\lambda_i - \lambda_1)^2 (\lambda_2 - \lambda_i)(\lambda_i - \lambda_3)^2 \cdots (\lambda_i - \lambda_d)^2}{f_2^2 (\lambda_2 - \theta_1)^2 (\lambda_2 - \lambda_3)^2 \cdots (\lambda_2 - \lambda_d)^2} \cdot \left(\frac{\varepsilon}{\lambda_2 - \sigma} \right)^{2(m-d)}.$$

Since, by (B.29) $\theta_1 = \lambda_1 + O(\varepsilon^{2(m-d)})$, θ_1 in (B.30) can be replaced by λ_1 without affecting the most significant term (i.e., the $\varepsilon^{2(m-d)}$ term) on the right-hand side. (19) can be obtained for $k = 2$. Following the similar procedure, we can show (19) for $k = 2, \ldots, d$. With (19) and the relationship between eigenvalues and eigenvectors, (20) can be shown as in Appendix A. □

Although this simpler proof and the one in Appendix A seem to give the same results, they are significantly different in several respects. Since the simpler one starts with a particular trial polynomials $\{p_k(x)\}$, it is an ad hoc approach and does not indicate how tight the bound is. In other words, we do not know whether $\varepsilon^{2(m-d)}$ is the largest order of the eigenvalue error after the mth Lanczos step, while the other one (in Appendix A) starts with the exact equations on θ_k's and obtain the final error estimates by sequentially removing the higher order error terms and keeping the dominating terms. Knowing that $\theta_i - \sigma = O(\varepsilon)$, it indicates that the best possible order of error in terms of ε is $2(m-d)$ for the principal eigenvalues and $m-d$ for the principal eigenvectors.

There are also some differences between this proof and the proof of the improved bound in [12]. Although the improved bound is a strict bound and Theorem 3.2 is an error estimate, the improved bound is not as tight as the intermediate results of this proof, e.g., (B.29) for $k = 1$ and (B.30) for $k = 2$, which are also strict bounds. The reason is that the improved bound [12] is derived with certain unnecessary enlargement such as

$$\text{(B.31)}\quad \|p_k(\mathbf{A})\mathbf{h}\| < \|(\mathbf{A} - \theta_1 \mathbf{I}) \cdots (\mathbf{A} - \theta_{k-1})\mathbf{I}\| \cdot \|\tilde{p}_k(\mathbf{A})\mathbf{h}\|,$$

where $\tilde{p}_k(x) = p_k(x) / ((x - \theta_1) \cdots (x - \theta_{k-1}))$ and $\mathbf{h}$ is the projection of $\mathbf{f}$ onto the orthogonal subspace of $\mathcal{E}^k = \text{span}\{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$.

Acknowledgments. The authors would like to express their gratitude to Prof. B. N. Parlett of the University of California at Berkeley for his reading of this paper and constructive comments. We also thank the anonymous referees for their useful comments, especially the detailed suggestions on a simpler proof of Theorem 3.2 and a clearer presentation of certain results.

REFERENCES

[1] H. AKAIKE, *A new look at the statistical model identification*, IEEE Trans. Automat. Control, AC-19 (1974), pp. 716–723.

[2] T.W. ANDERSON *Asymptotic theory for principal component analysis*, Ann. Math. Statist., 34 (1963), pp. 122–148.

[3] M. BARTLETT, *The effect of standardization on a $\mathcal{X}^2$ approximation in factor analysis*, Biometrika, 38 (1951), pp. 337–344.

[4] G. BIENVENU AND L. KOPP, *Optimality of high resolution array processing*, IEEE Trans. ASSP, ASSP-31 (1983), pp. 953–964,

[5] J. CHUN, *Fast Array Algorithms for Structured Matrices*, Ph.D. thesis, Stanford University, Stanford, CA, June 1989.

[6] P. COMON AND G. GOLUB, *Tracking a few extreme singular values and vectors in signal processing*, Proc. IEEE, 78 (1990), pp. 1327–1343.

[7] J. CULLUM AND R. WILLOUGHBY, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol.* I *Theory*, Birkhäuser Boston Inc., Boston, MA, 1985.

[8] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1990.

[9] C. LANCZOS, *An iterative method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, B (1950), pp. 225–280.

[10] C. PAIGE, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, University of London, London, UK, 1971.

[11] ———, *Computational variants of the Lanczos method for eigenproblem*, J. Inst. Math. Appl., 10 (1987), pp. 373–381.

[12] B. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood, NJ, 1980.

[13] J. RISSANEN, *Modeling by shortest data description*, Automatica, 14 (1978), pp. 465–471.

[14] R.H. ROY, *ESPRIT, Estimation of Signal Parameters via Rotational Invariance Techniques*, Ph.D. thesis, Stanford University, Stanford, CA, August 1987.

[15] R.O. SCHMIDT, *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*, Ph.D. thesis, Stanford University, Stanford, CA, November 1981.

[16] M. WAX AND T. KAILATH, *Detection of signals by information theoretic criteria*, IEEE Trans. ASSP, ASSP-33 (1985), pp. 387–392.

[17] G. XU AND T. KAILATH, *Fast Subspace Decomposition*, IEEE Trans. Signal Processing, March 1994.

[18] ———, *A fast algorithm for signal subspace decomposition and its performance analysis*, Proc. Internat. Conf. of Acoustics, Speech and Signal Processing, Toronto, Canada, (1991), pp. 3069–3072.

[19] G. XU, *Fast Signal Subspace Decomposition and Its Applications*, Ph.D. thesis, Stanford University, Stanford, CA, September 1991.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 995–1006, July 1994

LCP DEGREE THEORY AND ORIENTED MATROIDS*

WALTER D. MORRIS, JR.†

Abstract. It is shown that the degree of a square-oriented matroid $\mathcal{M}$ can be defined in terms of the number of solutions to the oriented matroid complementarity problem defined by a point extension of $\mathcal{M}$, and that this definition is independent of the point extension. If $\mathcal{M}$ is represented by a matrix $[I, -M]$, then this degree is the same as the degree of the LCP mapping defined by M. The average value of the degree is determined. A new characterization of P-matrices in terms of degree theory is given. A negative result concerning Q-matrices defining maps of degree zero is presented.

Key words. linear complementarity problem, degree theory, oriented matroids

AMS subject classification. 90C33

1. Introduction. In [23] and [24], Todd developed a framework for using oriented matroid theory for studying the linear complementarity problem (LCP). One of the main results of that work was the development of an algorithm for oriented matroid programming that both preserved feasibility and avoided cycling. This was a major discovery for oriented matroid theory. A great deal of LCP theory that had been developed to consider more general instances of LCPs than those considered in the papers by Todd was shown in [11] to be derived in a unified way by considering the degree of a mapping, given in [4], defined by the matrix of an LCP. In particular, degree theory can be used in many instances to show that the number of solutions of a certain type to an LCP is the same for all nondegenerate choices of the right-hand side. LCP degree theory has been applied in several papers [8], [9] recently to study the stability of solutions to LCPs. LCP degree theory is also central to the study of the class of Q-matrices (matrices that define LCPs that have solutions for any right-hand side).

We show that one can also incorporate the theory of the degree of an LCP mapping into Todd's oriented matroid LCP framework. The LCP mapping itself has no direct analog in oriented matroid theory, but the calculation of its degree is made by examining the circuits of the oriented matroid represented by the matrix of the LCP. Thus LCP degree theory gives a new integer valued function on the set of oriented matroids, and we investigate some properties of this function. Readers familiar with LCP degree theory know that many results are proved by theorems involving homotopies. The analogous process of changing one oriented matroid into another by changing the orientation of one basis at a time has received much attention, and it is known that in some cases it is not possible. Therefore, we cannot use arguments of this type. However, we show that such analytical tools are unnecessary and derive all our results combinatorially.

The mapping of an LCP is determined by its matrix, and the degree is calculated using the right-hand side. It is proved in [11] that under certain nondegeneracy assumptions, the number calculated is independent of the right-hand side at which

* Received by the editors February 18, 1992; accepted for publication (in revised form) February 26, 1993.

† Department of Mathematical Sciences, George Mason University, Fairfax, Virginia 22030 (wmorris@gmuvax.gmu.edu).

it is calculated. We develop an analogous result for oriented matroids, showing that the degree of a square oriented matroid, calculated with an extension of the oriented matroid, does not depend on the extension chosen. Once this is accomplished, we determine that the average value of the degree of a square oriented matroid on $2n$ elements is 2^{-n}. This is proved using results on the degrees of oriented matroids obtained from each other by reorientations. We will also derive a characterization of P-oriented matroids (generalizations of P-matrices) in terms of their "reorientation vectors."

For matrices, this implies that the average degree of the LCP mapping defined by a matrix chosen "at random" is 2^{-n}, and that a matrix is a P-matrix if and only if the degrees of the mappings defined by it and the matrices obtained by changing the signs of its rows and columns follow a certain pattern.

The Q-matrix problem, finding necessary and sufficient conditions on a matrix M so that an LCP defined by it always has a solution, is one of the main problems of LCP theory. The discovery of Q-matrices representing maps of degree zero, in [12], put limits on the extent to which oriented matroid theory could aid in the investigation of the Q-matrix problem. On the other hand, it is as yet unknown how large the set of such matrices is. We show that the degree zero Q-matrix given by Howe [10] represents a Q-oriented matroid; that is, every oriented matroid complementarity problem defined by an extension of the oriented matroid represented by this matrix has a solution.

2. Definitions. Oriented matroid theory was developed independently by various researchers. The papers of Bland and Las Vergnas [2] and Folkman and Lawrence [5] are two of the earliest papers on the subject. Todd [23] introduced the study of complementarity in oriented matroids. A simplified development of Todd's theory in a more general setting is found in a recent paper by Fukuda and Terlaky [6]. As in [23], we mostly use the notation found in [2]. The oriented matroid axioms that we have found most convenient to use are from [16]. Our definition of lexicographic extensions becomes simpler using these axioms. Apart from these considerations, the definitions and facts from this section follow those in [23] directly.

Let M be in $\mathbf{R}^{n\times n}$ and q in $\mathbf{R}^n$. The *linear complementarity problem* defined by M and q (the LCP(M,q)) is to find vectors y and x in $\mathbf{R}^n$ satisfying $y - Mx = q$, $y^T x = 0$, $y \geq 0, x \geq 0$. Background on the linear complementarity problem, as well as definitions of matrix classes used in this paper, can be found in [3]. The LCP can be seen as the search for a vector in the nullspace of the matrix $[I, -M, -q]$ having a prescribed sign pattern. We can think of such sign patterns as combinatorial objects that can be manipulated according to certain rules. This leads us to oriented matroids.

Let E be a finite set. A *signed set* on E is a pair $X = (X^+, X^-)$ of disjoint subsets of E. The set $\underline{X} = X^+ \cup X^-$ is called the set *underlying* X and $-X = (X^-, X^+)$ is called the *opposite* of X. An *oriented matroid* $\mathcal{M}$ on E is a pair $(E, \mathcal{K})$, where E is a finite set and $\mathcal{K}$ is a collection of signed sets on E satisfying the following:

(K1) $(\emptyset, \emptyset) \in \mathcal{K}$, and $K \in \mathcal{K}$ implies that $-K \in \mathcal{K}$.

(K2) If $K_1, K_2 \in \mathcal{K}$, then $K_3 = (K_1^+ \cup (K_2^+ \setminus K_1^-), K_1^- \cup (K_2^- \setminus K_1^+)) \in \mathcal{K}$.

(K3) If $K_1, K_2, \in \mathcal{K}, e \in (K_1^+ \cap K_2^-) \cup (K_1^- \cap K_2^+)$, there exists $K_3 \in \mathcal{K}$ with $e \notin \underline{K}_3, ((K_1^+ \setminus K_2^-) \cup (K_2^+ \setminus K_1^-)) \subseteq K_3^+, ((K_1^- \setminus K_2^+) \cup (K_2^- \setminus K_1^+)) \subseteq K_3^-, \underline{K}_3 \subseteq \underline{K}_1 \cup \underline{K}_2$.

The signed set K_3 obtained in $(K2)$ is said to be the *composition* of K_1 and K_2, written $K_3 = K_1 \circ K_2$. A signed set K_3 obtained as in $(K3)$ is said to be obtained by *eliminating* e between K_1 and K_2. The nonzero members of $\mathcal{K}$ with minimal underlying sets are called *circuits* of $\mathcal{M}$. We write $\mathcal{K} = \mathcal{K}(\mathcal{M})$. A fundamental

result of [2] is that members of $\mathcal{K}$ can be *conformally decomposed* into circuits of $\mathcal{M}$; that is, for any $K \in \mathcal{K}(\mathcal{M})$ there exist circuits $C_1, C_2, \ldots, C_k$ of $\mathcal{K}(\mathcal{M})$ so that $K^+ = C_1^+ \cup C_2^+ \cup \cdots \cup C_k^+$, $K^- = C_1^- \cup C_2^- \cup \cdots \cup C_k^-$.

An oriented matroid $\mathcal{M}$ on a set $E = \{e_1, e_2, \ldots, e_n\}$ is said to be *represented* by an $m \times n$ matrix A if $\mathcal{K}(\mathcal{M})$ is the collection of signed sets $(\{e_i : x_i > 0\}, \{e_i : x_i < 0\})$ for x satisfying $Ax = 0$.

A maximal subset of E that does not contain the underlying set of any circuit of $\mathcal{M}$ is called a *base* of $\mathcal{M}$. All bases of $\mathcal{M}$ have the same cardinality, called the *rank* of $\mathcal{M}$. Let $B(\mathcal{M})$ be the set of bases of $\mathcal{M}$, and let $\beta(\mathcal{M})$ be the set of ordered bases of $\mathcal{M}$. Las Vergnas showed (see [15]) that there is a map $\varepsilon : \beta(\mathcal{M}) \to \{-1, 1\}$ satisfying the following:

(E1) If β_1, β_2 are two orderings of a base B, then $\varepsilon(\beta_1) = \varepsilon(\beta_2)$ if β_2 can be obtained from β_1 by an even permutation.

(E2) If β_1 is an ordering of B_1, β_2 is an ordering of B_2, and β_1, β_2 agree in all but one position, then $\varepsilon(\beta_1) = \varepsilon(\beta_2)$ if each circuit C of M with $\underline{C} \subseteq B_1 \cup B_2$ has $B_1 \setminus B_2 \subseteq C^+$ and $B_2 \setminus B_1 \subseteq C^-$ or vice versa.

Furthermore, if $\varepsilon_1, \varepsilon_2$ satisfy the above, then $\varepsilon_1 = \pm\varepsilon_2$. A map ε satisfying (E1) and (E2) will be called a consistent assignment of signs to the bases of $\mathcal{M}$. We may drop the word consistent. In the case in which $\mathcal{M}$ of rank m is represented by an $m \times n$ matrix A, the bases of $\mathcal{M}$ correspond to nonsingular $m \times m$ submatrices of A, and one of the assignments of signs to bases of $\mathcal{M}$ assigns to each base the sign of the determinant of the corresponding $m \times m$ submatrix of A.

If B is a base of $\mathcal{M}$ and $f \in E \backslash B$, there is a unique circuit $C = C(B, f)$ with $\underline{C} \subseteq B \cup f$ and $f \in C^+$. This is called the fundamental circuit associated with B and f.

If F, G are disjoint subsets of E, define an oriented matroid $\mathcal{M}\backslash F/G$ on $E\backslash(F \cup G)$ by $\mathcal{K}(\mathcal{M}\backslash F/G) = \{(K^+\backslash G, K^-\backslash G) : K \in \mathcal{K}, \underline{K} \cap F = \emptyset\}$. We say that $\mathcal{M}\backslash F/G$ is obtained from $\mathcal{M}$ by *deleting* F and *contracting* G. We will write $\mathcal{M}\backslash q$ as an abbreviation for $\mathcal{M}\backslash\{q\}/\emptyset$.

A *point extension* of an oriented matroid $\mathcal{M}$ on a set E is an oriented matroid $\hat{\mathcal{M}}$ on $E \cup q$ so that $\hat{\mathcal{M}}\backslash q = \mathcal{M}$. We say then that q extends $\mathcal{M}$ to $\hat{\mathcal{M}}$. All of the point extensions $\hat{\mathcal{M}}$ considered in this paper are of the same rank as $\mathcal{M}$. Of particular importance to us are the *lexicographic extensions*, originally defined in [14]. Let $\mathcal{M}$ be an oriented matroid on a set E, and let $F = (f_1, f_2, \ldots, f_k)$ be a nonempty subset of E that does not contain the set underlying any circuit of $\mathcal{M}$. For every base B of $\mathcal{M}$ and every $i = 1, \ldots, k$ define $\check{C}(B, f_i) = (\emptyset, f_i)$ if $f_i \in B$, $\check{C}(B, f_i) = (C(B, f_i)^+\backslash f_i, C(B, f_i)^-)$ if $f_i \notin B$. Then define $C(B, q)$ to be the signed set $(q, \emptyset) \circ \check{C}(B, f_1) \circ \cdots \circ \check{C}(B, f_k)$. From [23] we get that the set $\mathcal{C} = \{C : C \text{ is a circuit of } \mathcal{M}\} \cup \{C(B, q) : B \text{ is a base of } \mathcal{M}\} \cup \{-C(B, q) : B \text{ is a base of } \mathcal{M}\}$ is the set of circuits of a point extension $\hat{\mathcal{M}}$ of $\mathcal{M}$, in which case we say $q = \text{lex}(f_1, f_2, \ldots, f_k)$ extends $\mathcal{M}$ to $\hat{\mathcal{M}}$. Todd showed that every circuit of $\hat{\mathcal{M}}$ containing q will then have at least $k + 1$ elements in its underlying set.

A *square oriented matroid* $\mathcal{M}$ is an oriented matroid on a set $E = S \cup T$ where $S = \{s_1, s_2, \ldots, s_n\}$, $T = \{t_1, t_2, \ldots, t_n\}$, $S \cap T = \emptyset$ and S is a base of $\mathcal{M}$. Oriented matroids represented by matrices of the form $[I, -M]$, for $M \in \mathbf{R}^{n \times n}$ are square, with the elements $s_1, s_2, \ldots, s_n$ corresponding to the columns of I. A subset A of such a set E is called *complementary* if $|\{s_i, t_i\} \cap A| \leq 1$ for $i = 1, \ldots, n$. If p extends a square oriented matroid $\mathcal{M}$ to $\hat{\mathcal{M}}$, the *oriented matroid complementarity problem* (OMCP) defined by $\hat{\mathcal{M}}$ is to find a circuit C of $\hat{\mathcal{M}}$ so that $C^- = \emptyset$ (C is *positive*), $p \in C^+$, and

$\underline{C}\backslash p$ is complementary.

We will work with three different nondegeneracy assumptions for our square oriented matroids:

(TND) All circuits of $\mathcal{M}$ have $n+1$ elements.

(ND) No circuits of $\mathcal{M}$ are complementary.

(WND) No positive circuits of $\mathcal{M}$ are complementary.

If $\mathcal{M}$ is represented by the $n\times 2n$ matrix $[I,-M]$, then $\mathcal{M}$ satisfies TND (totally nondegenerate) if every $n\times n$ submatrix of $[I,-M]$ is nonsingular, $\mathcal{M}$ satisfies ND (nondegenerate) if there is no nonzero $x\in\mathbf{R}^n$ so that $x_i(Mx)_i=0$ for $i=1,\ldots,n$, or equivalently, all principal minors of M are nonzero. $\mathcal{M}$ satisfies WND (weakly nondegenerate) if there is no nonzero $x\in\mathbf{R}^n$ with $x\geq 0$, $Mx\geq 0$, and $x_i(Mx)_i=0$ for $i=1,\ldots,n$, i.e., M is an R_0-matrix. It should be clear that TND $\Longrightarrow$ ND $\Longrightarrow$ WND.

3. The degree of a square oriented matroid. In this section we show that it is possible to define the degree of a square oriented matroid in terms of the solution set of the OMCP defined by an extension and that this definition is independent of the extension chosen.

DEFINITION 3.1. *Let $\mathcal{M}$ be a square oriented matroid of rank n. Let ε be the assignment of signs to bases of $\mathcal{M}$ that satisfies $\varepsilon(s_1,s_2,\ldots,s_n)=1$. For every complementary base B of $\mathcal{M}$, define the* index *of B,* $\mathrm{ind}(B)$, *to be* $(-1)^{|B\cap T|}\varepsilon(b_1,b_2,\ldots,b_n)$, *where $b_i\in\{s_i,t_i\}$ for $i=1,\ldots,n$.*

DEFINITION 3.2. *Let $\mathcal{M}$ be a square oriented matroid of rank n satisfying* WND. *Let p extend $\mathcal{M}$ to $\hat{\mathcal{M}}$, so that every positive circuit C of $\hat{\mathcal{M}}$ containing p with $\underline{C}\backslash p$ complementary has $n+1$ elements. Then the* degree *of $\mathcal{M}$ is the sum of the values of* $\mathrm{ind}(B)$ *for the bases B of $\mathcal{M}$ for which $C(B,p)$ solves the* OMCP *of $\hat{\mathcal{M}}$.*

To show that the degree is well defined, we pick a particular extension and show that the degree calculated with any other extension gives the same number. For a square oriented matroid $\mathcal{M}$ of rank n represented by a matrix $[I,-M]$, $q=\mathrm{lex}(s_1,s_2,\ldots,s_n)$ extends $\mathcal{M}$ to an oriented matroid represented by $[I,-M,r]$, where $r=(\varepsilon,\varepsilon^2,\ldots,\varepsilon^n)^T$, for all sufficiently small positive ε.

THEOREM 3.1. *Under the assumptions of Definition* 3.2, *the degree of $\mathcal{M}$ is well defined.*

Proof. Let $\mathcal{M}$ and $\hat{\mathcal{M}}$ be as in definition 3.2. Let $q=\mathrm{lex}(s_1,s_2,\ldots,s_n)$ extend $\hat{\mathcal{M}}$ to $\bar{\mathcal{M}}$. Using the ideas of [22] and [23], one can construct a graph for which the vertices correspond to the positive circuits C of $\bar{\mathcal{M}}$ that have $\underline{C}\backslash\{p,q\}$ complementary. Two vertices corresponding to circuits C_1 and C_2 are connected by an edge if $|\underline{C}_1\triangle\underline{C}_2|=2$. This graph turns out (see [23]) to be the union of disjoint paths and cycles, where every vertex corresponding to a circuit containing p and q has two neighbors, and every vertex corresponding to a circuit containing exactly one of p or q has one neighbor. We also have (see [23]) that:

(1) If $C(B_1,p)$ and $C(B_2,p)$ correspond to the two endpoints of a path, or if $C(B_1,q)$ and $C(B_2,q)$ correspond to the two endpoints of a path, then $\mathrm{ind}(B_1)=-\mathrm{ind}(B_2)$.

(2) If $C(B_1,p)$ and $C(B_2,q)$ correspond to the two endpoints of a path, then $\mathrm{ind}(B_1)=\mathrm{ind}(B_2)$.

This implies that if we add up the values of $\mathrm{ind}(B)$ for all B for which $C(B,p)$ solves the OMCP of $\hat{\mathcal{M}}$, we get the same number as when we sum up the values of $\mathrm{ind}(B)$ for all B so that $C(B,q)$ is positive and $\underline{C}(B,q)\backslash q$ is complementary. But these are

the solutions to the OMCP defined by $\check{\mathcal{M}} = \bar{\mathcal{M}}\backslash p$, the same oriented matroid as the extension of $\mathcal{M}$ by $q = \text{lex}(s_1, s_2, \ldots, s_n)$. □

Some ideas of the above proof go back to Lemke [25]. With oriented matroids one must take extra care, since two extensions may not be compatible. If p extends $\mathcal{M}$ to $\hat{\mathcal{M}}$ and q extends $\mathcal{M}$ to $\check{\mathcal{M}}$, there does not necessarily exist an oriented matroid $\bar{\mathcal{M}}$ so that $\bar{\mathcal{M}}\backslash q = \hat{\mathcal{M}}$ and $\bar{\mathcal{M}}\backslash p = \check{\mathcal{M}}$. Fortunately, this lexicographic extension is compatible with any other extension. The lexicographic extension is also useful because it guarantees that all of the circuits corresponding to vertices of the graph contain $n + 1$ elements.

The assumption that $\mathcal{M}$ satisfies WND is sufficient for the graph defined in the proof to have the structure described. When $\mathcal{M}$ does not satisfy WND, things can easily go wrong. For example, if $n = 1$ and $\mathcal{M}$ is represented by the matrix $[1, 0]$, then $\mathcal{M}$ does not satisfy WND as $(t_1, \emptyset)$ is a positive complementary circuit. Calculating the degree using the extension represented by $[1, 0, 1]$ gives 0 as the answer, while calculating it at $[1, 0, -1]$ gives 1. However, the assumption is not strictly necessary. Let $M = \left(\begin{smallmatrix} 0 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. Then the oriented matroid represented by $[I, -M]$ does not satisfy WND, but calculating the degree at any extension as in the definition yields 0. There does not seem to be any obvious simple necessary and sufficient condition for the degree to be well defined, but the assumption WND includes many important cases.

If $\mathcal{M}$ satisfying WND is represented by a matrix $[I, -M]$, then the degree of $\mathcal{M}$ is the same as the degree of the mapping determined by M, as defined in [11]. This mapping sends the standard basis vectors of $\mathbf{R}^n$ to themselves, sends the negative of the ith standard basis vector to the ith column of $-M$, for $i = 1, \ldots, n$ and is linear on each orthant. A consequence of the definition is that the mappings defined by two matrices M_1 and M_2, for which the oriented matroid represented by $[I, -M_1]$ is the same as that represented by $[I, -M_2]$, have the same degree.

Recent work by Stewart [21] (see also [7]) gives a method for calculating the degree of $\mathcal{M}$ with extensions $\hat{\mathcal{M}}$ for which there are solutions to the LCP of $\hat{\mathcal{M}}$ that have less than $n + 1$ elements. Lemma 3.1 gives an indication of how one can deal with extensions for which some of the solutions to the resulting OMCP have less than $n + 1$ elements.

LEMMA 3.1. *Let $\mathcal{M}$ be a square oriented matroid satisfying* WND. *If the degree of $\mathcal{M}$ is nonzero, then the* OMCP *defined by any extension of $\mathcal{M}$ has a solution.*

Proof. Let $\mathcal{M}$ be a square oriented matroid of nonzero degree. Let q extend $\mathcal{M}$ to $\hat{\mathcal{M}}$. If $(q, \emptyset)$ is a circuit of $\hat{\mathcal{M}}$, then this is a solution to the OMCP of $\hat{\mathcal{M}}$. If not, then there is an ordered base F of $\hat{\mathcal{M}}$ containing q in the first position. Let $p = \text{lex}(F)$ extend $\hat{\mathcal{M}}$ to $\bar{\mathcal{M}}$, and let $\check{\mathcal{M}} = \bar{\mathcal{M}}\backslash q$. Then $\check{\mathcal{M}}$ satisfies the conditions of Theorem 3.1, so its OMCP must have a solution. A solution $C(B, p)$ to the OMCP of $\check{\mathcal{M}}$ implies, by the definition of lexicographic extensions, that there is a circuit $C(B, q)$ of $\hat{\mathcal{M}}$ that solves the OMCP of $\hat{\mathcal{M}}$. □

It was observed in [11] that specifying the signs of the principal minors of a matrix M does not determine the degree of the LCP mapping defined by M. On the other hand, from the definition of lexicographic extensions one sees that if $\mathcal{M}$ is represented by $[I, -M]$ and $q = \text{lex}(s_1, s_2, \ldots, s_n)$ extends $\mathcal{M}$ to $\hat{\mathcal{M}}$, then the solutions to the OMCP of $\hat{\mathcal{M}}$, and hence the degree of $\mathcal{M}$, are determined by the circuits of $\mathcal{M}$ of the form $C(B, s_i)$, where B is a complementary base of $\mathcal{M}$ not containing s_i. By (E2) these circuits in turn are determined by the values of $\varepsilon(\beta)$ for orderings β of bases B that are either complementary or contain $\{s_i, t_i\}$ for exactly one subscript i (Here

we assume that $\varepsilon(s_1, s_2, \ldots, s_n) = 1$). In the representable case, these values of ε are determined by the determinants of square submatrices of M that either are principal or are defined by a set of rows I and a set of columns J with $|I \triangle J| = 2$.

4. Reorientation results. All of the square oriented matroids in this section are assumed to satisfy ND. We would like to obtain relationships between the degree of a square oriented matroid $\mathcal{M}$ and the degrees of its minors.

DEFINITION 4.1. *If F is a subset of $E = S \cup T$, then define the oriented matroid $\mathcal{M}_{\bar{F}}$ by $\mathcal{K}(\mathcal{M}_{\bar{F}}) = \{((K^+ \backslash F) \cup (K^- \cap F), (K^- \backslash F) \cup (K^+ \cap F)), K \in \mathcal{K}(\mathcal{M})\}$.*

The oriented matroid $\mathcal{M}_{\bar{F}}$ is said to be obtained from reorienting F. We write $\deg(\mathcal{M})$ for the degree of $\mathcal{M}$. If ε is an assignment of signs to bases of $\mathcal{M}$, then $\varepsilon_{\bar{F}}$ defined by $\varepsilon_{\bar{F}}(\beta) = \varepsilon(\beta)(-1)^{|\beta \cap F|}$ is an assignment of signs to bases of $\mathcal{M}_{\bar{F}}$.

DEFINITION 4.2. *Let $\mathcal{M}$ be a square oriented matroid. Let ε be an assignment of signs to ordered bases of $\mathcal{M}$. Define the assignment signs $\varepsilon \backslash s_i$ of signs to ordered bases of $\mathcal{M} \backslash t_i / s_i$ by $\varepsilon \backslash s_i(\beta) = \varepsilon(\bar{\beta})$, where $\bar{\beta}$ is obtained from β by inserting s_i in the ith position. Similarly, define $\varepsilon \backslash t_i$ for $\mathcal{M} \backslash s_i / t_i$ by $\varepsilon \backslash t_i(\beta) = \varepsilon(\bar{\beta})$, where $\bar{\beta}$ is obtained from β by inserting t_i into the ith position.*

LEMMA 4.1. *Let $\mathcal{M}$ be a square oriented matroid satisfying* ND. *Then*

(1) $\deg(\mathcal{M}) + \deg(\mathcal{M}_{\bar{s}_i}) = \deg(\mathcal{M} \backslash t_i / s_i)$;

(2) $\deg(\mathcal{M}) - \deg(\mathcal{M}_{\bar{t}_i}) = -\deg(\mathcal{M} \backslash s_i / t_i)(\varepsilon(\sigma_i))$

for $i = 1, 2, \ldots, n$, where σ_i is the ordering of $S \backslash s_i \cup t_i$ for which the subscripts are in increasing order and $\varepsilon(s_1, s_2, \ldots, s_n) = 1$.

Proof. We first prove (1). Let F be a complementary ordered base of $\mathcal{M}$ containing t_i in the first position. Let $q = \text{lex}(F)$ extend $\mathcal{M}$ to $\hat{\mathcal{M}}$. Note that if B is a complementary base of $\mathcal{M}$ containing t_i, then in $\hat{\mathcal{M}}$ we must have $t_i \in C(B, q)^-$ by the definition of lexicographic extensions. Thus all complementary bases B that solve the OMCP of $\hat{\mathcal{M}}$ contain s_i. The same is true for all bases B that solve the OMCP of $\hat{\mathcal{M}}_{\bar{s}_i}$, and no base can solve both the OMCP of $\hat{\mathcal{M}}$ and the OMCP of $\hat{\mathcal{M}}_{\bar{s}_i}$. For any base B that solves the OMCP of $\hat{\mathcal{M}}$ or $\hat{\mathcal{M}}_{\bar{s}_i}$, the base $B \backslash s_i$ of $\mathcal{M} \backslash t_i / s_i$ solves the OMCP of $\hat{\mathcal{M}} \backslash t_i / s_i$. If ε is the assignment of signs to ordered bases of $\mathcal{M}$ satisfying $\varepsilon(s_1, s_2, \ldots, s_n) = 1$, and β is a complementary ordered base of $\mathcal{M}$ containing s_i, then $\varepsilon(\beta) = -\varepsilon_{\bar{s}_i}(\beta) = (\varepsilon \backslash s_i)(\beta \backslash s_i)$. But $\varepsilon_{\bar{s}_i}$ is the opposite of the assignment of bases to ordered bases of $\hat{\mathcal{M}}_{\bar{s}_i}$ that assigns 1 to $(s_1, s_2, \ldots, s_n)$. Thus if β is an ordering of B, then the index of B in $\hat{\mathcal{M}}$ is the same as the index of B in $\hat{\mathcal{M}}_{\bar{s}_i}$, and this is the same as the index of $B \backslash s_i$ in $\hat{\mathcal{M}} \backslash t_i / s_i$. Thus (1) follows.

To prove (2), we analogously extend $\mathcal{M}$ to $\hat{\mathcal{M}}$ by $q = \text{lex}(F)$, where F is an ordered base of $\mathcal{M}$ with s_i in the first position. For a complementary ordered base β of $\mathcal{M}$ containing t_i we still have $\varepsilon(\beta) = -\varepsilon_{\bar{t}_i}(\beta)$, but now if ε assigns 1 to $(s_1, s_2, \ldots, s_n)$, so does $\varepsilon_{\bar{t}_i}$. We also have $\varepsilon(\beta) = \varepsilon \backslash t_i(\beta \backslash t_i)$, but here $\varepsilon \backslash t_i$ assigns 1 to $(s_1, s_2, \ldots, s_n) \backslash s_i$ only if $\varepsilon(\sigma_i) = 1$. The index formula counts $|B \cap T|$, so we must multiply by -1 to account for deleting t_i. □

If $\mathcal{M}$ is represented by $[I, -M]$, then: $\mathcal{M}_{\bar{t}_i}$ is represented by $[I, -M_1]$, where M_1 is obtained from M by negating the ith column; $\mathcal{M}_{\bar{s}_i}$ is represented by $[I, -M_2]$, where M_2 is obtained from M by negating the ith row; $\mathcal{M} \backslash t_i / s_i$ is represented by $[I, -M_3]$, where M_3 is obtained from M by deleting the ith row and column; $\mathcal{M} \backslash s_i / t_i$ is represented by $[I, -M_4]$, where M_4 is obtained from M by first pivoting on the ith diagonal element of M and then deleting the ith row and column of the resulting matrix. Finally, $\varepsilon(\sigma_i)$ is the sign of the ith diagonal element of $-M$.

When $n = 1$, there are only two square oriented matroids satisfying ND. These

are the oriented matroids represented by the matrices $[1,1]$ and $[1,-1]$. It is easy to check that if we define the *empty* oriented matroid $\mathcal{M} = (\emptyset, \{(\emptyset,\emptyset)\})$ to have degree 1, then Lemma 4.1 holds in the case $n = 1$.

THEOREM 4.1. *Let $\mathcal{M}$ be a square oriented matroid of rank n satisfying* ND. *Then if $N = \{1,2,\ldots,n\}$*

$$\sum_{I\subseteq N} \deg(\mathcal{M}_{\bar{S}_I}) = 1,$$

where $S_I = \{s_i : i \in I\}$.

Proof. The proof is by induction on n. By the definition of the degree of the empty oriented matroid, the statement is true for $n = 0$. For $n = 1$, the oriented matroid represented by the matrix $[1,-1]$ has degree 1, and the oriented matroid represented by $[1,1]$ has degree 0. If $\mathcal{M}$ is one of these two, then $\mathcal{M}_{\bar{s}_1}$ is the other. Suppose next that the statement is true for all square oriented matroids of rank $n-1$, and let $\mathcal{M}$ be a square oriented matroid of rank n. Then

$$\begin{aligned}\sum_{I\subseteq N} \deg(\mathcal{M}_{\bar{S}_I}) &= \sum_{J\subseteq(N\setminus n)} \deg(\mathcal{M}_{\bar{S}_J}) + \deg(\mathcal{M}_{\overline{S_J\cup s_n}}) \\ &= \sum_{J\subseteq(N\setminus n)} \deg(\mathcal{M}_{\bar{S}_J}\backslash t_n/s_n) = 1.\end{aligned}$$

The last equality is the inductive hypothesis. □

Since there are 2^n subsets of $\{1,2,\ldots,n\}$, it follows that the average value of the degree of a square oriented matroid satisfying ND is 2^{-n}. The implication for the matrix case is stated as a corollary.

COROLLARY 4.1. *Let P be a probability measure on $\mathbf{R}^{n\times n}$ for which* (1) *$P(ND) = 1$, where* ND *is the set of matrices in $\mathbf{R}^{n\times n}$ with all principal minors nonzero.*

(2) *If $G\subseteq\mathbf{R}^{n\times n}$ and H is obtained from G by negating the ith row of each element of G, then $P(G) = P(H)$ for $i = 1,2,\ldots,n$.*

Then the expected value of the degree of the mapping defined by a matrix chosen at random from $\mathbf{R}^{n\times n}$ according to P is 2^{-n}.

Due to the asymmetry of Lemma 4.1, Corollary 4.1 does not remain true when "row" is replaced by "column."

Theorem 4.1 implies that for at least one of the subsets I, the degree of $\mathcal{M}_{\bar{S}_I}$ must be nonzero. Matrices for which the corresponding maps have nonzero degree are known to be Q-matrices. (This also follows from Lemma 3.1.) This justifies the following corollary.

COROLLARY 4.2. *Let P be as in Corollary* 4.1, *and let Q be the set of Q-matrices in $\mathbf{R}^{n\times n}$. Then $P(Q) \geq 2^{-n}$.*

It is plausible that as n approaches infinity, $P(Q)$ goes to zero for P as above. It seems to be difficult to prove this.

One can get various other equations similar to those of Lemma 4.1. For example, we have the following lemma.

LEMMA 4.2. *Let $\mathcal{M}$ be a square oriented matroid of rank n satisfying* ND. *Then*

(1) $\deg(\mathcal{M}_{\bar{t}_i}) + \deg(\mathcal{M}_{\overline{s_i\cup t_i}}) = \deg(\mathcal{M}\backslash t_i/s_i)$,

(2) $\deg(\mathcal{M}_{\bar{s}_i}) - \deg(\mathcal{M}_{\overline{s_i\cup t_i}}) = \deg(\mathcal{M}\backslash s_i/t_i)(\varepsilon(\sigma_i))$, *and*

(3) $\deg(\mathcal{M}) + \deg(\mathcal{M}_{\bar{s}_i}) - \deg(\mathcal{M}_{\bar{t}_i}) - \deg(\mathcal{M}_{\overline{s_i\cup t_i}}) = 0$ *for $i = 1,2,\ldots,n$.*

Parts (1) and (2) are proved similarly to Lemma 4.1 and part (3) is a linear combination of the first parts of Lemmas 4.1 and 4.2, or a combination of their second parts.

The degree of a square oriented matroid is not always determined by those of its minors. For example, if $M_1 = \left(\begin{smallmatrix} -1 & 2 \\ 2 & -1 \end{smallmatrix}\right)$ and $M_2 = \left(\begin{smallmatrix} -1 & -2 \\ -2 & -1 \end{smallmatrix}\right)$, and $\mathcal{M}_1$ is represented by $[I, -M_1]$, $\mathcal{M}_2$ is represented by $[I, -M_2]$, then $\mathcal{M}_1\backslash t_i/s_i = \mathcal{M}_2\backslash t_i/s_i$ and $\mathcal{M}_1\backslash s_i/t_i = \mathcal{M}_2\backslash s_i/t_i$ for $i = 1, 2$. On the other hand, $\deg(\mathcal{M}_1) = -1$ while $\deg(\mathcal{M}_2) = 0$.

We would like to organize the study of equations such as those proved above by introducing the *reorientation vector* of a square oriented matroid. This is a vector with 2^{2n} components, indexed by the subsets of $E = S \cup T$ ordered lexicographically. For a square oriented matroid $\mathcal{M}$ of rank n, the component of the reorientation vector of $\mathcal{M}$ corresponding to a set $I \subseteq E$ is $\deg(\mathcal{M}_{\bar{I}})$. The reorientation vector is not defined if $\mathcal{M}$ does not satisfy ND, because if $\mathcal{M}$ has a complementary circuit, then this circuit is positive for some reorientation of $\mathcal{M}$.

PROBLEM 4.1. Characterize the set of reorientation vectors of square oriented matroids of rank n.

One hope is that a characterization of the set of reorientation vectors would indicate a relationship between this set and other sets of vectors in combinatorial geometry, such as the set of f-vectors of convex polytopes. This search is partly motivated by the fact that the tools from algebra instrumental in proving the upper bounds on the components of the f-vectors were also used by the author in proving special cases of the conjectured bound on the entries of the reorientation vector in [19]. The conjectured bound, given in [18], on the degree of an LCP mapping is

$$\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}.$$

A trivial observation to make is that for any $I \subseteq E$, we have $\mathcal{M}_{\bar{I}} = \mathcal{M}_{\overline{E\backslash I}}$. This cuts down the dimension of the linear span of the reorientation vectors to 2^{2n-1}. Theorem 4.1 and Lemma 4.2 give us more affine and linear equations satisfied by these vectors.

In the case that $\mathcal{M}$ satisfies TND, we can get help from oriented matroid theory. We first prove the following lemma, which is valid under a weaker nondegeneracy assumption.

LEMMA 4.3. *Let $\mathcal{M}$ be a square oriented matroid satisfying* WND. *If $\mathcal{K}(\mathcal{M})$ does not contain the signed set $(E, \emptyset)$ then* $\deg(\mathcal{M}) = 0$.

Proof. If $\mathcal{K}(\mathcal{M})$ does not contain the signed set $(E, \emptyset)$, then there is an element $e \in E$ so that no positive circuits of $\mathcal{K}(\mathcal{M})$ contain e. Furthermore, e is the first element of some ordered base F. Let $q = \text{lex}(F)$ extend $\mathcal{M}$ to $\hat{\mathcal{M}}$. Then $\hat{\mathcal{M}}$ has no positive circuits containing q. Hence we have that the OMCP of $\hat{\mathcal{M}}$ has no solutions and $\deg(\mathcal{M}) = 0$.

An oriented matroid $\mathcal{M}$ with $(E, \emptyset) \in \mathcal{K}(\mathcal{M})$ is said to be *acyclic*. If $\mathcal{M}$ is represented by $[I, -M]$ with M square and $\mathcal{M}$ is acyclic, then M is an S-matrix.

If $\mathcal{M}$ satisfies TND and $\mathcal{M}$ is square, it follows from [13] that $\mathcal{M}_{\bar{I}}$ is acyclic for exactly half of the subsets $I \subseteq E$. Furthermore, the collection of these subsets of E has the combinatorial structure of a "barely unlopsided set" (see [13]).

COROLLARY 4.3. *If $\mathcal{M}$ of rank greater than zero is square and satisfies* TND, *then at least half of the components of its reorientation vector are zero.*

We close this section by showing that the set of P-oriented matroids is characterized by its reorientation vectors. A square oriented matroid $\mathcal{M}$ is called a P-oriented matroid if for every circuit C of $\mathcal{M}$ there is a subscript i so that $\{s_i, t_i\} \subseteq C^+$ or $\{s_i, t_i\} \subseteq C^-$. See [14] for other equivalent characterizations of these oriented matroids. A matrix M for which $[I, -M]$ represents a P-oriented matroid $\mathcal{M}$ is a P-matrix.

THEOREM 4.2. *A square oriented matroid $\mathcal{M}$ of rank n is a P-oriented matroid if and only if for all subsets J of E we have* $\deg(\mathcal{M}_{\bar{J}}) = 1$ *if* $J = S_I \cup T_I$ *for some* $I \subseteq \{1, 2, \ldots, n\}$, $\deg(\mathcal{M}_{\bar{J}}) = 0$ *otherwise.*

Proof. Suppose first that $\mathcal{M}$ is a P-oriented matroid. From [23] it follows that the OMCP of every extension of $\mathcal{M}$ has a unique solution. In particular, if $q = \text{lex}(s_1, s_2, \ldots, s_n)$ extends $\mathcal{M}$ to $\check{\mathcal{M}}$, then the OMCP defined by $\hat{\mathcal{M}} = \check{\mathcal{M}}_{\bar{q}}$ has as its solution $C(S, q)$. Since $\text{ind}(S) = 1$, $\deg(\mathcal{M}) = 1$. If $J = S_I \cup T_I$ for some I, it is clear that $\mathcal{M}_{\bar{J}}$ is also a P-oriented matroid, hence its degree is also 1. If J is not of the form $S_I \cup T_I$ for any I, assume without loss of generality that $s_i \in J, t_i \notin J$. Let F be an ordered base of $\mathcal{M}$ containing s_i in the first position, and let $q = \text{lex}(F)$ extend $\mathcal{M}_{\bar{J}}$ to $\hat{\mathcal{M}}_{\bar{J}}$. As in the proof of Lemma 4.2, we see that all solutions B to the LCP of $\hat{\mathcal{M}}_{\bar{J}}$ must contain t_i. However, since the only index j for which $\{s_j, t_j\} \subseteq \underline{C}(B, s_i)$ for such a B is i, we must have $t_i \in C(B, s_i)^-$ by the P-oriented matroid property of $\mathcal{M}$. Thus the OMCP of $\hat{\mathcal{M}}_{\bar{J}}$ has no solution and $\deg(\mathcal{M}_{\bar{J}}) = 0$.

The opposite implication will be proved by induction on n, the rank of $\mathcal{M}$. The implication is true for $n = 0$ because we defined the degree of the empty oriented matroid to be 1 and it is a P-oriented matroid. The case $n = 1$ can be quickly checked. Suppose next that $\mathcal{M}$ is of rank $n > 1$, and that for all subsets J of E we have $\deg(\mathcal{M}_{\bar{J}}) = 1$ if J is of the form $S_I \cup T_I$, $\deg(\mathcal{M}_{\bar{J}}) = 0$ otherwise. Let $J = S_I \cup T_I$ for some $I \subseteq (N \backslash n)$. Then by Lemma 4.1, $\deg((\mathcal{M}_{\bar{J}}) \backslash t_n / s_n) = \deg(\mathcal{M}_{\bar{J}}) + \deg(\mathcal{M}_{\overline{J \cup s_n}}) = 1 + 0 = 1$. On the other hand, if J is not of the form $S_I \cup T_I$ for any I, and also $J \cap \{s_n, t_n\} = \emptyset$, then $\deg((\mathcal{M}_{\bar{J}}) \backslash t_n / s_n) = \deg(\mathcal{M}_{\bar{J}}) + \deg(\mathcal{M}_{\overline{J \cup s_n}}) = 0 + 0 = 0$ and $\deg((\mathcal{M}_{\bar{J}}) \backslash s_n / t_n) = \pm(\deg(\mathcal{M}_{\bar{J}}) - \deg(\mathcal{M}_{\overline{J \cup t_n}})) = \pm(0 - 0) = 0$. By induction we can say that $\mathcal{M} \backslash t_n / s_n$ is a P-oriented matroid. If we apply Theorem 4.1 to $(\mathcal{M}_{\bar{T}_I}) \backslash s_n / t_n$ where $I \subseteq (N \backslash n)$, we have that $\deg((\mathcal{M}_{\overline{S_I \cup T_I}}) \backslash s_n / t_n) = 1$. Thus we can also say by induction that $\mathcal{M} \backslash s_n / t_n$ is a P-oriented matroid. Here n was arbitrary, so we can replace n by any $i = 1, 2, \ldots, n$. Finally, if C is a circuit of $\mathcal{M}$, then there must be an element of E that is not in $\underline{C}$, say s_n. Then $C \in \mathcal{K}(\mathcal{M} \backslash s_n / t_n)$, which is a P-oriented matroid. Thus there is a subscript i in $\{1, 2, \ldots, n-1\}$ for which $\{s_i, t_i\} \subseteq C^+$ or $\{s_i, t_i\} \subseteq C^-$. This means that $\mathcal{M}$ is a P-oriented matroid. □

5. A degree zero Q-oriented matroid. A matrix M in $\mathbf{R}^{n \times n}$ is a Q-matrix if the LCP defined by M and a vector q has a solution for all $q \in \mathbf{R}^n$. It is well known that if the degree of the mapping defined by M is nonzero, then M is a Q-matrix. On the other hand, the discovery by Kelly and Watson [12] of a Q-matrix defining a map of degree zero, showed that the Q-matrix property is more difficult to establish than the calculation of the degree. The Kelly and Watson example gave a matrix on the boundary of the set of Q-matrices, showing that the set of Q-matrices is not open in $\mathbf{R}^{n \times n}$. The set of matrices in $\mathbf{R}^{n \times n}$ defining maps of nonzero degree is open. Kelly and Watson go on to ask if there are other interesting open sets of Q-matrices.

Part of the mystique concerning Q-matrices is generated by the fact that the only known method that can guarantee to tell if a matrix is a Q-matrix, due to Gale, requires the solution of n^{2^n} linear programs in the worst case, each with 2^n constraints. Recent work by Naiman and Stone (see [20]) has reduced the amount of work needed to a roughly $O(2^{3n^2})$ algorithm. Compare this to the obvious algorithm for calculating the degree by solving 2^n linear systems for some arbitrary right-hand side. The result is that one can calculate the degree by hand for 4×4 matrices quite easily, whereas showing that a 4×4 matrix is a Q-matrix often involves a "proof by picture" argument as in [12].

For a square oriented matroid $\mathcal{M}$ of rank n satisfying TND, the set of matrices $M \in \mathbf{R}^{n\times n}$ for which $[I, -M]$ represents $\mathcal{M}$ is an open set. (However, for $n \geq 5$, it is not necessarily connected! See [1].) Thus, if we can produce an oriented matroid $\mathcal{M}$ satisfying TND so that for any point extension $\hat{\mathcal{M}}$, the OMCP has a solution, we have an open set of Q-matrices.

DEFINITION 5.1. *A square oriented matroid $\mathcal{M}$ is called a* Q-oriented matroid *if for every point extension $\hat{\mathcal{M}}$ of $\mathcal{M}$ the* OMCP *of $\hat{\mathcal{M}}$ has a solution.*

If a matrix M is a Q-matrix, it is not necessarily true that the oriented matroid $\mathcal{M}$ represented by $[I, -M]$ is a Q-oriented matroid. $\mathcal{M}$ may have nonrepresentable extensions or representable extensions that cannot be represented by adding a column to $[I, -M]$, for which the OMCP has no solutions. An example of a Q-matrix M for which the oriented matroid $\mathcal{M}$ represented by $[I, -M]$ satisfies TND, but is not a Q-oriented matroid, is given in [17]. Thus the boundary of the set of Q-matrices can run through the interior of a set of matrices representing a particular TND oriented matroid of degree zero. It is not a priori clear that there must exist such a set entirely contained within the set of Q-matrices. Theorem 5.1 shows that there is such an example.

THEOREM 5.1. *There exists a Q-oriented matroid of degree zero.*

Proof. Let

$$M = \begin{pmatrix} -4 & 3 & 3 & 5 \\ 3 & -4 & 3 & 5 \\ 3 & 3 & -4 & 5 \\ 5 & 5 & 5 & -4 \end{pmatrix}.$$

Howe [10] showed that the mapping defined by M had degree zero. Howe also showed that M was a Q-matrix. Let $\tilde{\mathcal{M}}$ be the oriented matroid represented by $[I, -M]$. The proof of [10] can be translated into oriented matroidese to show that $\tilde{\mathcal{M}}$ is a Q-oriented matroid. We do this for the benefit of those not conversant with this language.

LEMMA 5.1. *If $\mathcal{M}$ is a square oriented matroid satisfying, for some i,*

(1) *for all $j \neq i$, $s_i \notin C(S, t_j)^-$*

(2) *$\mathcal{M}\backslash t_i/s_i$ is a Q-oriented matroid.*

Then if q extends $\mathcal{M}$ to $\hat{\mathcal{M}}$ with $s_i \notin C(S, q)^-$, the OMCP *of $\hat{\mathcal{M}}$ will have a solution.*

Proof of Lemma 5.1. Let C_i be a solution to the OMCP $\hat{\mathcal{M}}\backslash t_i/s_i$. Then there is a $K \in \mathcal{K}(\mathcal{M})$ with $K\backslash s_i = C_i$. If $s_i \in K^+$, then K is a solution to the OMCP $\hat{\mathcal{M}}$. Suppose that $s_i \in K^-$. If $T \cap \underline{K} = \emptyset$, then $K = C(S, q)$, contradicting $s_i \notin C(S, q)^-$. Otherwise, eliminate t_j for some $t_j \in T \cap K$ between K and $-C(S, t_j)$, obtaining $K' \in \mathcal{K}(\mathcal{M})$ with $s_i \in K'^-, q \in K'^+, |T \cap \underline{K'}| = |T \cap \underline{K}| - 1, K'^- \cap T = \emptyset$. Continue this way until $T \cap \underline{K}$ is empty, obtaining a contradiction. □

Howe also showed that the maps defined by the 3×3 principal minors of M are all of nonzero degree. It follows that the oriented matroids $\tilde{\mathcal{M}}\backslash t_i/s_i$ are Q-oriented matroids for $i = 1, 2, 3, 4$. Lemma 5.1 then shows that there only remains to find a solution to the OMCPs of extensions $\hat{\mathcal{M}}$ of $\tilde{\mathcal{M}}$ with $C(S, q) = (q, S)$. Let $\hat{\mathcal{M}}$ be such an extension. One can calculate that $\mathcal{K}(\tilde{\mathcal{M}})$ contains the positive circuits $(\{s_1, s_4, t_1, t_2, t_3\}, \emptyset), (\{s_2, s_4, t_1, t_2, t_3\}, \emptyset)$, and $(\{s_3, s_4, t_1, t_2, t_3\}, \emptyset)$. For $B = \{t_1, t_2, t_3, s_4\}$, these are the circuits $C(B, s_i)$, for $i = 1, 2, 3$. By successively eliminating s_i between $C(S, q)$ and $C(B, s_i)$ for $i = 1, 2, 3$, as in the proof of Lemma 5.1 above, one gets $C(B, q)$, with $\{t_1, t_2, t_3\} \subseteq C(B, q)^+$. If $s_4 \in C(B, q)^+$, then $C(B, q)$ will solve the OMCP of $\hat{\mathcal{M}}$. Suppose then that $s_4 \in C(B, q)^-$. We can also calculate that $\mathcal{K}(\mathcal{M})$ contains the circuits $C_1 = (\{s_1, s_2, t_3, t_4\}, t_1), C_2 = (\{s_1, s_2, s_4, t_3, t_4\}, \emptyset)$,

and $C_3 = (\{s_1, s_2, t_3, t_4\}, t_2)$. Eliminate t_1 between $C(B, q)$ and C_1 to get $K_1 = (\{s_1, s_2, t_2, t_3, t_4, q\}, s_4)$ of $\mathcal{K}(\hat{\mathcal{M}})$. Then eliminate s_4 between K_1 and C_2 to get $K_2 = (\{s_1, s_2, t_2, t_3, t_4, q\}, \emptyset) \in \mathcal{K}(\hat{\mathcal{M}})$. Finally, eliminate t_2 between K_2 and C_3 to get $(\{s_1, s_2, t_3, t_4, q\}, \emptyset)$, which solves the OMCP. □

The oriented matroid $\tilde{\mathcal{M}}$ did not satisfy TND, so we cannot say that the set of matrices M for which $[I, -M]$ represents $\tilde{\mathcal{M}}$ is an open set. However, one can get a *perturbation* of $\tilde{\mathcal{M}}$ that satisfies TND and that has the relevant properties of $\tilde{\mathcal{M}}$. The idea of perturbations was shown in the proof of Lemma 3.1, where q was replaced by p. Because q was in the first position of F, circuits of $\hat{\mathcal{M}}$ with $n+1$ elements, containing q become circuits of $\tilde{\mathcal{M}}$ with p replacing q. We can start with $\tilde{\mathcal{M}}$ and replace t_i by t_i' this way successively for $i = 1, 2, \ldots, n$. The resulting oriented matroid will satisfy TND. All of the relevant circuits of $\tilde{\mathcal{M}}$ used in the proof contained $n+1 = 5$ elements, hence they will also be circuits of the perturbed oriented matroid. Thus the set of matrices M for which $[I, -M]$ represents this perturbed oriented matroid will be an open set of Q-matrices defining maps of degree zero.

6. Remarks. Theorems 3.1 and 5.1 were written in an earlier version of this paper that was not submitted for publication. Comments of M. J. Todd were very helpful for the improvement of this paper. The anonymous referees also provided very helpful comments.

REFERENCES

[1] J. Bokowski and A. G. de Oliveira, *Simplicial convex 4-polytopes do not have the isotopy property*, Portugal. Math., 47 (1990), pp. 309–318.

[2] R. G. Bland and M. Las Vegnas, *Orientability of matroids*, J. Combin. Theory Ser. B, 24 (1978), pp. 94–123.

[3] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[4] B. C. Eaves and H. Scarf, *The solution of systems of piecewise linear equations*, Math. Oper. Res., 1 (1976), pp. 1–27.

[5] J. Folkman and J. Lawrence, *Oriented matroids*, J. Combin. Theory Ser. B, 25 (1978), pp. 199–236.

[6] K. Fukuda and T. Terlaky, *Linear complementarity and oriented matroids*, J. Oper. Res. Soc. Japan, 35 (1992) pp. 45–61.

[7] M. S. Gowda, *A degree formula of Stewart*, Research Report 91-13, Dept. of Mathematics, University of Maryland, Baltimore County, Catonsville, 1991.

[8] M. S. Gowda and J.-S. Pang, *Stability analysis of variational inequalities and nonlinear complementarity problems, via the mixed linear complementarity problem and degree theory*, Research Report 91-21, University of Maryland at Baltimore County, Catonsville, 1991.

[9] C. D. Ha, *Application of degree theory in stability of the complementarity problem*, Math. Oper. Res., 12 (1987), pp. 368–376.

[10] R. Howe, *On a class of linear complementarity problems of variable degree*, in Homotopy methods and global convergence, B. C. Eaves, F. J. Gould, H. O. Peitgen, and M. J. Todd, eds., Plenum Press, New York, 1983, pp. 155–177.

[11] R. Howe and R. Stone, *Linear complementarity and the degree of mappings*, in Homotopy methods and global convergence, B. C. Eaves, F. J. Gould, H. O. Peitgen, and M. J. Todd, eds., Plenum Press, New York, 1983, pp. 179–223.

[12] L. M. Kelly and L. T. Watson, *Q-matrices and spherical geometry*, Linear Algebra Appl., 25 (1979), pp. 175–189.

[13] J. Lawrence, *Lopsided sets and orthant intersection by convex sets*, Pacific J. Math., 104 (1983), pp. 155–173.

[14] M. Las Vergnas, *Extensions ponctuelles d'une geometrie combinatoire orientee*, in Problemes

combinatoires et theorie des graphes, Actes du Colloque International C.N.R.S., No. 260, Orsay 1976, Paris, 1978, pp. 263–268.

[15] M. LAS VERGNAS, *Bases in oriented matroids*, J. Combin. Theory Ser. B, 25 (1978), pp. 283–289.

[16] A. MANDEL, *Topology of oriented matroids*, Ph.D. thesis, University of Waterloo, Ontario, 1982.

[17] W. D. MORRIS, JR., *Counterexamples to Q-matrix conjectures*, Linear Algebra Appl., 111 (1988), pp. 135–145.

[18] ———, *On the maximum degree of an* LCP *map*, Math. Oper. Res., 15 (1990), pp. 423–429.

[19] ———, *The maximum number of complementary facets of a simplicial polytope*, Disc. Appl. Math., 15 (1992), pp. 293–298.

[20] D. Q. NAIMAN AND R. E. STONE, Private communication, 1991.

[21] D. E. STEWART, *A degree theory approach to degeneracy of* LCP*s*, Linear Algebra Appl., to appear.

[22] M. J. TODD, *Orientation in complementary pivot algorithms*, Math. Oper. Res., 1 (1976), pp. 54–66.

[23] ———, *Complementarity in oriented matroids*, SIAM J. Alg. Discrete Meth., 5 (1984), pp. 467–485.

[24] ———, *Linear and quadratic programming in oriented matroids*, J. Combin. Theory Ser. B, 39 (1985), pp. 105–133.

[25] C. E. LEMKE, *Bimatrix equilibrium points and mathematical programming*, Management Sci., 11 (1965), pp. 681–689.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 1007–1014, July 1994

020

VARIATION OF THE UNITARY PART OF A MATRIX*

RAJENDRA BHATIA† AND KALYAN MUKHERJEA‡

Abstract. The derivative of the map that takes an invertible matrix A to the unitary factor U in the polar decomposition $A = UP$ is evaluated. The same is done for the map that takes A to the unitary factor Q in the QR decomposition $A = QR$. These results lead to perturbation bounds for these maps. Other applications of the method developed are discussed.

Key words. polar decomposition, QR decomposition, Cholesky factorisation, perturbation, manifold, tangent space, unitarily invariant norm, singular value, Fréchet derivative

AMS subject classifications. 15A45, 65F99

1. Introduction.

Let $\mathbf{M}(n)$ be the space of all $n \times n$ (complex) matrices; let $\mathbf{GL}(n)$ be the group consisting of all invertible matrices and let $\mathbf{U}(n)$ be the subgroup of unitary matrices. Every matrix A has a *polar decomposition* $A = UP$, where $U \in \mathbf{U}(n)$ and P is positive semidefinite. The positive part P, written as $|A|$, is unique and is equal to $(A^*A)^{1/2}$. If $A \in \mathbf{GL}(n)$ then the polar part U is also unique, since $U = AP^{-1}$.

Let $F : \mathbf{GL}(n) \longrightarrow \mathbf{U}(n)$ be the map $F(UP) = U$, which takes an invertible matrix to its polar part. Our first result, Theorem 2.1 below, gives an explicit expression for the Fréchet derivative of this map. As corollaries we obtain the value of the norm of this derivative with respect to any unitarily invariant norm on $\mathbf{M}(n)$, and then a perturbation bound for the polar part.

Another expression for the derivative of F has been obtained by Barrlund [1]. Using this and some results on Hadamard products, Mathias [12] has obtained the perturbation bound (13) derived below. Our coordinate-free approach to these questions is in line with some of our earlier work [3], [6], and [2, Chaps. 4, 5]. This approach has two merits. First, it is adaptable to more general contexts such as the KAK decomposition in semisimple Lie groups. We do not pursue that direction in this paper. Second, it works well for other matrix decompositions like the QR factorisation and the Cholesky factorisation. We illustrate this in later sections of this paper. Results similar to these have been obtained by Stewart [13] and, more recently, by Sun [15]. Here our approach clarifies some of the issues, unifies the work on these different questions, and clearly brings out the similarities and the differences between them.

We will denote by $|||\cdot|||$ any norm on $\mathbf{M}(n)$ that is *unitarily invariant*, i.e., a norm that satisfies the condition $|||UAV||| = |||A|||$ for all $A \in \mathbf{M}(n)$ and $U, V \in \mathbf{U}(n)$. Basic properties of such norms may be found in [2]. The singular values of A will be denoted as $s_1(A) \geq s_2(A) \geq \cdots \geq s_n(A)$. The *operator bound norm*, also called the *spectral norm* in the numerical analysis literature, will be denoted by $||\cdot||$ and the *Frobenius norm* by $||\cdot||_F$. We have

$$||A|| = s_1(A),$$

* Received by the editors September 30, 1992; accepted for publication (in revised form) January 31, 1993.

† Indian Statistical Institute, New Delhi 110016, India (rbh@isid.ernet.in). The work of this author was supported by a Department of Atomic Energy research grant.

‡ Indian Statistical Institute, Calcutta 700035, India.

$$||A||_F = \left[\sum_j s_j^2(A)\right]^{1/2}.$$

If $\mathcal{T}$ is a transformer, i.e., a linear map on the space $\mathbf{M}(n)$, then for any norm $|||\cdot|||$ on $\mathbf{M}(n)$, we define

$$|||\mathcal{T}||| = \sup\{|||\mathcal{T}(X)||| : |||X||| = 1\}.$$

We will use some elementary facts of calculus on manifolds that the reader may find in texts such as [7].

2. Variation of the unitary part. Let $\mathbf{T}_A\mathbf{GL}(n)$ be the tangent space to the manifold $\mathbf{GL}(n)$ at a point A in it. Since $\mathbf{GL}(n)$ is an open subset of $\mathbf{M}(n)$ we have $\mathbf{T}_A\mathbf{GL}(n) = \mathbf{M}(n)$. This is a special instance of the correspondence between a Lie group and its Lie algebra. Here the Lie algebra corresponding to the group $\mathbf{GL}(n)$ is $gl(n) = \mathbf{M}(n)$. The Lie algebra corresponding to the group $\mathbf{U}(n)$ is $u(n)$, the set of all skew-Hermitian matrices. This is the tangent space to $\mathbf{U}(n)$ at the point I. The tangent space to $\mathbf{U}(n)$ at a point U is $\mathbf{T}_U\mathbf{U}(n) = U \cdot u(n) = \{US : S \in u(n)\}$. The derivative of F at a point $A = UP$ of $\mathbf{GL}(n)$, denoted by $DF(UP)$, is a linear map from $\mathbf{M}(n)$ to $U \cdot u(n)$.

Let $h(n)$ denote the space of all Hermitian matrices. We have $h(n) = i \cdot u(n)$. We have a vector space decomposition

$$\mathbf{M}(n) = u(n) + h(n), \tag{1}$$

in which every matrix splits uniquely as

$$X = S + H, \tag{2}$$

where

$$S = \frac{X - X^*}{2}, \qquad H = \frac{X + X^*}{2}. \tag{3}$$

We can now state our first main result.

THEOREM 2.1. *Let $F : \mathbf{GL}(n) \longrightarrow \mathbf{U}(n)$ be the map defined above as $F(UP) = U$. Let X be any element of $\mathbf{M}(n)$ and let $X = S + H$ be its splitting into skew-Hermitian and Hermitian parts. Then the value of the derivative $DF(UP)$ on the tangent vector UX is given by*

$$DF(UP)(UX) = 2U \int_0^\infty e^{-tP} S e^{-tP} dt. \tag{4}$$

Proof. Let $\mathbf{P}(n)$ be the set of all $n \times n$ positive definite matrices. This is an open subset of the real vector space $h(n)$. Hence for every $P \in \mathbf{P}(n)$ the tangent space $\mathbf{T}_P\mathbf{P}(n) = h(n)$.

Let $\Psi : \mathbf{U}(n) \times \mathbf{P}(n) \longrightarrow \mathbf{GL}(n)$ be the map $\Psi(U, P) = UP$ and let Φ be the inverse map $\Phi(UP) = (U, P)$. Then writing $\Phi = (\Phi_1, \Phi_2)$, we have $F = \Phi_1$.

The derivative $D\Psi(U, P)$ is a linear map with domain $\mathbf{T}_U\mathbf{U}(n) + \mathbf{T}_P\mathbf{P}(n) = U \cdot u(n) + h(n)$ and range $\mathbf{M}(n) = U \cdot u(n) + U \cdot h(n)$. By definition, this derivative is evaluated as

$$D\Psi(U, P)(US, H) = \frac{d}{dt}[\Psi(Ue^{tS}, P + tH)]_{t=0} = USP + UH \tag{5}$$

for all $S \in u(n), H \in h(n)$.

Now note that for small values of t, $P+tH$ is positive for any $H \in h(n)$, and hence we have $\Phi_1(UP+tUH) = \Phi_1(UP) = U$. So the kernel of $D\Phi_1(UP)$ contains $U \cdot h(n)$. In fact, $\ker D\Phi_1(UP) = U \cdot h(n)$, since Φ is a diffeomorphism from $\mathbf{GL}(n)$ onto $\mathbf{U}(n) \times \mathbf{P}(n)$ and each of $u(n)$ and $h(n)$ has half the dimension of $\mathbf{M}(n)$. So we need to compute the value of $D\Phi_1(UP)$ only on tangent vectors of the form $US, S \in \mathrm{u}(n)$. Let

$$D\Phi(UP)(US) = (UM, N), \qquad M \in \mathrm{u}(n), N \in h(n). \tag{6}$$

Since $\Phi = \Psi^{-1}$, we have using (5)

$$US = D\Psi(U,P)(UM,N) = UMP + UN. \tag{7}$$

We want to determine M from this equation. So, we must solve the equation

$$MP + N = S.$$

Taking adjoints we have

$$-PM + N = -S.$$

From these two equations we obtain

$$MP + PM = 2S. \tag{8}$$

This is the familiar Lyapunov equation and its solution (see [9], [10]) is

$$M = 2\int_0^\infty e^{-tP} S e^{-tP} dt. \tag{9}$$

Equation (4) now follows from (6) and (9). □

COROLLARY 2.1. *For every unitarily invariant norm* $|||\cdot|||$ *on* $\mathbf{M}(n)$ *we have*

$$|||DF(UP)||| = ||P^{-1}|| = s_n^{-1}(A). \tag{10}$$

Proof. This follows from (4) by a familiar argument that we repeat for the reader's convenience.

Since the norm is unitarily invariant, we have

$$|||DF(UP)(UX)||| \leq 2\int_0^\infty |||e^{-tP} S e^{-tP}|||dt. \tag{11}$$

Then, since $|||BCD||| \leq ||B|| \cdot |||C||| \cdot ||D||$ for all B, C, D, we have

$$\begin{aligned} |||e^{-tP} S e^{-tP}||| &\leq ||e^{-tP}|| \cdot |||S||| \cdot ||e^{-tP}|| \\ &\leq e^{-2ts_n(A)}|||S||| \\ &\leq e^{-2ts_n(A)}|||X||| \end{aligned} \tag{12}$$

using the fact $|||S||| \leq |||X|||$.

From (11) and (12) we obtain

$$|||DF(UP)||| = \sup_{|||X|||=1} |||DF(UP)(UX)||| \leq s_n^{-1}(A).$$

Choosing $X = \imath I/|||I|||$ one sees that this is actually an equality. □

Using the mean value theorem, we obtain from Corollary 2.2.

COROLLARY 2.2. *Let A_0 and A_1 be two elements of $\mathbf{GL}(n)$ with polar parts U_0 and U_1, respectively. Assume that the line segment $A(t) = (1-t)A_0 + tA_1, 0 \leq t \leq 1$, joining A_0 and A_1 lies inside $\mathbf{GL}(n)$. Then for every unitarily invariant norm*

$$|||U_0 - U_1||| \leq \max_{0\leq t\leq 1} ||A(t)^{-1}||.|||A_0 - A_1|||. \tag{13}$$

These statements can be expressed in another language by saying that in any unitarily invariant norm, the *condition* of the function F at any point A of $\mathbf{GL}(n)$ is given by $s_n^{-1}(A)$.

We should remark that the solution of (8) can also be expressed as a Hadamard product [10], [11]; from this we can obtain estimates like ours either directly or by converting this formula to the integral expression (9). We have chosen the integral form of the solution because it might be useful in analysing infinite dimensional problems as well. An effective use of such integrals was made earlier in [5].

3. The QR decomposition. Every square complex matrix A can be written as a product $A = QR$ where Q is unitary and R is upper triangular. If A is invertible then so is R. Furthermore, we can choose the diagonal entries of R to be positive and with this added restriction this product decomposition is unique for every $A \in GL(n)$. This decomposition called the *QR decomposition* is extremely important in numerical analysis. See [14] for details.

We will now analyse the variation of the unitary part in this decomposition in the same way as for the polar decomposition.

Let $\mathbf{B}(n)$ denote the set of all upper triangular matrices with positive diagonal entries and let $b(n)$ be the set of all upper triangular matrices with real diagonal entries. Then $b(n)$ is a real vector space and $\mathbf{B}(n)$ is an open subset of it. So, the tangent space $\mathbf{T}_R\mathbf{B}(n)$ to $\mathbf{B}(n)$ at any point R of it is the space $b(n)$. (One may note here that $\mathbf{B}(n)$ is a Lie group and $b(n)$ is its Lie algebra.)

The QR decomposition associates with every element A of $\mathbf{GL}(n)$ a unique element Q of $\mathbf{U}(n)$ and a unique element R of $\mathbf{B}(n)$. Let $F : \mathbf{GL}(n) \longrightarrow \mathbf{U}(n)$ now be the map $F(QR) = Q$. The derivative of F at $A = QR$ is a linear map from $\mathbf{M}(n)$ to $Q{\cdot}\mathbf{U}(n)$.

The subspaces $u(n)$ and $b(n)$ are complementary to each other in $\mathbf{M}(n)$ and we have a vector space decomposition

$$\mathbf{M}(n) = u(n) + b(n). \tag{14}$$

This decomposition is not as familiar as the one in (1) and it has some different features. If a matrix X splits as

$$X = K + T \tag{15}$$

in the above decomposition then we must have the following relations between the entries of these matrices

$$k_{jj} = \imath \mathrm{Im}\, x_{jj} \quad \text{for all } j, \quad k_{ij} = -\overline{x}_{ji} \quad \text{for } j > i, \quad k_{ij} = x_{ij} \quad \text{for } i > j,$$

$$t_{jj} = \mathrm{Re}\, x_{jj} \quad \text{for all } j, \quad t_{ij} = x_{ij} + \overline{x}_{ji} \quad \text{for } j > i, \quad t_{ij} = 0 \quad \text{for } i > j. \tag{16}$$

Whereas, in the case of the decomposition (1) the projections onto both the components are norm-reducing for every unitarily invariant norm (just use the triangle inequality), this is not the case for the decomposition (14). Instead, we have for the Frobenius norm the following lemma.

LEMMA 3.1. *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the complementary projection operators in $\mathbf{M}(n)$ corresponding to the decomposition* (14). *Then*

$$||\mathcal{P}_1||_F = ||\mathcal{P}_2||_F = \sqrt{2}. \tag{17}$$

Proof. From (15) and (16) one can easily see that $||K||_F^2 \leq 2||X||_F^2$ and $||T||_F^2 \leq 2||X||_F^2$. The first inequality becomes an equality when $X = \left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)$, the second when $X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. □

Remark 3.1. If instead of the Frobenius norm the operator norm is used then the norms of the projections $\mathcal{P}_1$ and $\mathcal{P}_2$ grow with the dimension n. To see this, note that if X is Hermitian then

$$T = 2\Delta(X) - \operatorname{diag} X, \tag{18}$$

where Δ is the *triangular truncation* operator, i.e., for any matrix $A, \Delta(A)$ is the matrix obtained from A by replacing the entries below the main diagonal by zeros. It is well known that the norm $||\Delta||$ grows as $\log n$. For example, if X is the $n \times n$ Hermitian matrix whose diagonal entries are zero and whose off-diagonal entries are $x_{ij} = \sqrt{-1}/(i-j)$, then $||X|| \leq \pi$ and $||\Delta(X)|| \geq \frac{4}{5}\log n$. (See [8, p. 39].) On the other hand, $||\operatorname{diag} X|| \leq ||X||$. So $||\mathcal{P}_2||$ must grow at least as $\log n$. Hence so must $||\mathcal{P}_1||$.

Returning to the map $F(QR) = Q$, let us see how far an analysis similar to the one in §2 takes us. Now define $\Psi : \mathbf{U}(n) \times \mathbf{B}(n) \longrightarrow \mathbf{GL}(n)$ to be the map $\Psi(Q, R) = QR$ and let Φ be its inverse map $\Phi(QR) = (Q, R)$. If Φ is written as $\Phi = (\Phi_1, \Phi_2)$ then $F = \Phi_1$. The derivative $D\Psi(Q, R)$ is a linear map whose domain is $\mathbf{T}_Q\mathbf{U}(n) + \mathbf{T}_R\mathbf{B}(n) = Q \cdot u(n) + b(n)$, and whose range is $\mathbf{M}(n) = Q \cdot u(n) + Q \cdot b(n)$. The derivative is evaluated as

$$D\Psi(Q, R)(QK, T) = \frac{d}{dt}[\Psi(Qe^{tK}, R + tT)]_{t=0} = QKR + QT \tag{19}$$

for all $K \in u(n), T \in b(n)$.

If $R \in \mathbf{B}(n)$ and $T \in b(n)$ then for small values of t, $R + tT$ is in $\mathbf{B}(n)$. By the uniqueness of the QR factorisation, $\Phi_1(QR + tQT) = \Phi_1(QR) = Q$. Hence the space $Q \cdot b(n)$ is contained in $\ker D\Phi_1(QR)$. But, then counting their dimensions we can conclude that $Q \cdot b(n) = \ker D\Phi_1(QR)$. So we need to compute the values of $D\Phi_1(QR)$ only on tangent vectors of the form $QK, K \in u(n)$. Let

$$D\Phi(QR)(QK) = (QM, Y), \quad \text{where } M \in u(n), \qquad Y \in b(n). \tag{20}$$

Since $\Phi = \Psi^{-1}$, we have from (19) and (20)

$$QK = QMR + QY. \tag{21}$$

To determine M from this we need to solve the equation

$$MR + Y = K. \tag{22}$$

Here the similarity with the analysis in §2 ends. In (3) P and N were selfadjoint, so taking adjoints we could achieve a major simplification by eliminating the redundant variable N. We cannot do that here. However, we can still obtain an expression for M from (22). Rewrite this equation as

$$M + YR^{-1} = KR^{-1}.$$

Note that $M \in \mathrm{u}(n)$ and $YR^{-1} \in b(n)$. So $M = \mathcal{P}_1(KR^{-1})$ in the notation used earlier. We have thus proved the following theorem.

THEOREM 3.1. *Let $F : \mathbf{GL}(n) \longrightarrow \mathbf{U}(n)$ be the map defined as $F(QR) = Q$. Let X be any element of $\mathbf{M}(n)$ and let $X = K + T$ be its splitting in the decomposition $\mathbf{M}(n) = \mathrm{u}(n) + b(n)$. Then the value of the derivative $DF(QR)$ on the tangent vector QX is given by*

$$DF(QR)(QX) = Q\mathcal{P}_1(KR^{-1}), \tag{23}$$

where $\mathcal{P}_1$ is the projection operator in $\mathbf{M}(n)$ projecting onto $\mathrm{u}(n)$ along the complementary space $b(n)$.

Note that the quantities occurring in the above formula can be explicitly computed from the relations (16).

COROLLARY 3.1. *For every matrix $A = QR$ in $\mathbf{GL}(n)$, we have*

$$||DF(QR)||_F \leq \sqrt{2}||R^{-1}|| = \sqrt{2}||A^{-1}||. \tag{24}$$

Proof. Use Theorem 3.1, Lemma 3.1, the unitary invariance of the Frobenius norm, and the inequality $||ST||_F \leq ||S||_F||T||$ that is valid for any two matrices S and T. □

Using the mean value theorem we obtain the following corollary.

COROLLARY 3.2. *Let $A_0 = Q_0R_0$ and $A_1 = Q_1R_1$ be any two elements of $\mathbf{GL}(n)$. Suppose that the line segment $A(t) = (1-t)A_0 + tA_1$, $0 \leq t \leq 1$, joining A_0 and A_1 lies entirely inside $\mathbf{GL}(n)$. Then*

$$||Q_0 - Q_1||_F \leq \sqrt{2} \max_{0\leq t\leq 1} ||A(t)^{-1}|| \ ||A_0 - A_1||_F. \tag{25}$$

We should remark that from (23) we could surely derive some estimates for $|||DF(QR)|||$ for any unitarily invariant norm. These would, however, involve $|||\mathcal{P}_1|||$ and for this we have good estimates only in the case of the Frobenius norm.

4. The Cholesky factorisation. A common feature of our analysis of the polar decomposition and the QR decomposition is that we replaced the study of the map Φ, which takes a matrix to its factors, by that of its inverse map Ψ. This, being a multiplication map, is easier to handle. A similar idea is useful in the perturbation analysis of the Cholesky factorisation.

Every positive definite matrix A has a unique factorisation $A = R^*R$, where R is an upper triangular matrix with positive diagonal entries. This is called the *Cholesky factorisation.*

In our notation, we now have a map $\Phi : \mathbf{P}(n) \longrightarrow \mathbf{B}(n)$ defined as $\Phi(A) = R$, where R is the Cholesky factor of A. The inverse map is $\Psi(R) = R^*R$. The derivative $D\Psi(R)$ is a linear map from the tangent space $\mathbf{T}_R\mathbf{B}(n) = b(n)$ to the tangent space $\mathbf{T}_A\mathbf{P}(n) = h(n)$. This derivative is evaluated as

$$D\Psi(R)(T) = \frac{d}{dt}\ [\Psi(R + tT)]_{t=0} = R^*T + T^*R, \tag{26}$$

for every $T \in b(n)$.

Now, for $H \in h(n)$ let

$$D\Phi(A)(H) = T, \quad \text{where } T \in b(n). \tag{27}$$

Then since $\Phi = \Psi^{-1}$, we must have

$$R^*T + T^*R = H. \tag{28}$$

To estimate $|||D\Phi(A)|||$, we need to estimate T in terms of H and R. Rewrite (28) as

$$TR^{-1} + (TR^{-1})^* = (R^*)^{-1}HR^{-1}. \tag{29}$$

Since $TR^{-1} \in b(n)$, we have from (29)

$$||TR^{-1}||_F \leq \frac{1}{\sqrt{2}}||(R^*)^{-1}HR^{-1}||_F \leq \frac{1}{\sqrt{2}}||R^{-1}||^2||H||_F.$$

Since $||T||_F \leq ||TR^{-1}||_F||R||$, this gives

$$||T||_F \leq \frac{1}{\sqrt{2}}||R||\ ||R^{-1}||^2\ ||H||_F. \tag{30}$$

From (27) and (30), we get

$$||D\Phi(A)||_F \leq \frac{1}{\sqrt{2}}||R||\ ||R^{-1}||^2 = \frac{1}{\sqrt{2}}||A||^{1/2}\ ||A^{-1}||. \tag{31}$$

For the map Ψ, we could write from (26)

$$|||D\Psi(R)||| = \sup_{|||T|||=1} |||R^*T + T^*R||| \leq 2||R||, \tag{32}$$

for every unitarily invariant norm.

Inequalities (31) and (32) can be used to write perturbation bounds for Φ and Ψ as before.

Finally, we remark that from results of §§2 and 3, we can obtain some information about the variation of the positive part P in the polar decomposition and the upper triangular part R in the QR decomposition.

Note. In a sequel to this paper [4], the above analysis has been carried further to obtain perturbation bounds for several other matrix decompositions.

REFERENCES

[1] A. BARRLUND, *Perturbation bounds on the polar decomposition*, BIT, 30 (1989), pp. 101–113.

[2] R. BHATIA, *Perturbation Bounds for Matrix Eigenvalues*, Longman Scientific and Technical, Essex, UK, 1987.

[3] ———, *Analysis of spectral variation and some inequalities*, Trans. Amer. Math. Soc., 272 (1982), pp. 323–331.

[4] ———, *Matrix factorizations and their perturbations*, Linear Algebra Appl., 197 (1994), pp. 245–276.

[5] R. BHATIA, C. DAVIS, AND A. MCINTOSH, *Perturbation of spectral subspaces and solution of linear operator equations*, Linear Algebra Appl., 52 (1983), pp. 45–67.

[6] R. BHATIA AND K. K. MUKHERJEA, *On the rate of change of spectra of operators*, Linear Algebra Appl., 27 (1979), pp. 147–157.

[7] R. L. BISHOP AND R. J. CRITTENDEN, *Geometry of Manifolds*, Academic Press, New York, 1964.

[8] K. R. DAVIDSON, *Nest Algebras*, Longman Scientific and Technical, Essex, UK, 1988.
[9] E. HEINZ, *Beiträge zur Störungstheorie der Spectralzerlegung*, Math. Ann., 123 (1951), pp. 415–438.
[10] R. A. HORN, *The Hadamard product*, Proc. Symp. Appl. Math., Vol. 40, Amer. Math. Soc., 1990.
[11] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.
[12] R. MATHIAS, *Perturbation bounds for the polar decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 588–597.
[13] G. W. STEWART, *Perturbation bounds for the QR factorization of a matrix*, SIAM J. Numer. Anal., 14 (1977), pp. 509–518.
[14] G. W. STEWART AND J. -G. SUN, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[15] J. -G. SUN, *Perturbation bounds for the Cholesky and QR factorizations*, BIT, 31 (1991), pp. 341–352.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 1015–1017, July 1994

021

AN UPPER BOUND FOR THE REAL PART OF NONMAXIMAL EIGENVALUES OF NONNEGATIVE IRREDUCIBLE MATRICES*

SHMUEL FRIEDLAND† AND LEONID GURVITS‡

Abstract. Let A be a nonnegative irreducible matrix. In this note an upper bound is given for the real part of an eigenvalue of A that is different from its spectral radius.

Key words. nonnegative matrices, nonmaximal eigenvalues

AMS subject classifications. A18, A42, A48

Introduction. Let $M_n(\mathbf{R})$ denote the algebra of $n \times n$ real valued matrices. Assume that $A = (a_{ij})_1^n \in M_n(\mathbf{R})$ is a nonnegative irreducible matrix. The Perron–Frobenius theorem yields that $\rho(A)$, the spectral radius of A, is an algebraically simple eigenvalue of A with the corresponding positive right and left eigenvectors

$$Au = \rho(A)u, \quad A^T v = \rho(A)v, \quad 0 < u, v \in \mathbf{R}^n. \tag{1}$$

Arrange the eigenvalues of A in the following order

$$\rho(A) = \lambda_n(A) > \Re(\lambda_{n-1}(A)) \geq \cdots \geq \Re(\lambda_1(A)). \tag{2}$$

A is called diagonally symmetric if there exists a diagonal matrix D with positive diagonal entries and a nonnegative irreducible symmetric C so that $A = D^{-1}C$. Note that if A is diagonally symmetric, then all the eigenvalues of A are real and A is diagonable. The results of [Fri, §4] yield that any nonnegative irreducible diagonally symmetric matrix A satisfies

$$\rho(A) - \lambda_{n-1}(A) \geq \frac{1}{2}(\rho(A) - \max_{1 \leq i \leq n} a_{ii})\epsilon(A, u, v)^2. \tag{3}$$

Here

$$\begin{aligned} \epsilon(A, u, v) &= \inf_{\emptyset \neq U \subset \{1,\ldots,n\}, \operatorname{card}(U) \leq \lfloor \frac{n}{2} \rfloor} \frac{\sum_{i \in U, j \in \{1,\ldots,n\} \setminus U} a_{ij} v_i u_j + a_{ji} v_j u_i}{\sum_{i \in U} 2(\rho(A) - a_{ii}) v_i u_i} \\ &= \inf_{\emptyset \neq U \subset \{1,\ldots,n\}, \operatorname{card}(U) \leq \lfloor \frac{n}{2} \rfloor} \frac{\sum_{i \in U, j \in \{1,\ldots,n\} \setminus U} a_{ij} v_i u_j + a_{ji} v_j u_i}{\sum_{i \in U, 1 \leq j = i \leq n} a_{ij} v_i u_j + a_{ji} v_j u_i}. \end{aligned} \tag{4}$$

The aim of this paper to prove the following extension of (3).

THEOREM. *Let $A = (a_{ij})_1^n$ be a nonnegative irreducible matrix with the positive right and left eigenvectors u and v, respectively, satisfying* (1). *Arrange the eigenvalues of A in the order* (2). *Let $\epsilon(A, u, v)$ be defined by* (4). *Then*

$$\rho(A) - \Re(\lambda_{n-1}(A)) \geq \frac{1}{2}(\rho(A) - \max_{1 \leq i \leq n} a_{ii})\epsilon(A, u, v)^2. \tag{5}$$

*Received by the editors April 6, 1992; accepted for publication (in revised form) January 26, 1993.

†Department of Mathematics, Statistics, and Computer Science, University of Illinois, Chicago, Illinois 60680 (U12735@UICVM.BITNET).

‡Robotics Research Laboratory, Courant Institute of Mathematical Sciences, New York University, New York, New York 10012. Current address, Learning Systems Dept., Siemens Corporation Research, Princeton, New Jersey 08540 (gurvits@scr.siemens.com).

Assume that A is an irreducible stochastic matrix. Then choose $u = e = (1, \dots, 1)^T$ and $v = \pi = (\pi_1, \dots, \pi_n)^T$ to be the stationary probability vector of the Markov chain corresponding to A. As $\rho(A) = 1$ we get (5) with

$$\epsilon(A, e, \pi) = \inf_{\emptyset \neq U \subset \{1,\dots,n\}, \operatorname{card}(U) \leq \lfloor \frac{n}{2} \rfloor} \frac{\sum_{i \in U, j \in \{1,\dots,n\} \setminus U} a_{ij}\pi_i + a_{ji}\pi_j}{\sum_{i \in U} 2(1 - a_{ii})\pi_i}. \tag{6}$$

Let A be an irreducible permutation matrix, i.e., A has one cycle. Hence, the n eigenvalues of A are the nth roots of unity. In particular, $\Re(\lambda_{n-1}(A)) = \cos \frac{2\pi}{n}$. Note that A is orthogonal and hence a normal matrix. In this case we let $u = v = e$. A straightforward computation shows that $\epsilon(A, e, e) = \frac{1}{n-1}$. Hence the inequality (5) in this case reduces to

$$1 - \cos \frac{2\pi}{n} \geq \frac{1}{2(n-1)^2}, \qquad n \geq 2.$$

In the case where a stochastic matrix A corresponds to a time reversible Markov chain, i.e., A is diagonably symmetric, the inequality (5) is due to Sinclair and Jerrum [S-J, Lemma 3.3].

Proof of the Theorem. Let $D = \operatorname{diag}(d_1, \dots, d_n)$ be a diagonal matrix with positive diagonal entries. Set $\hat{A} = DAD^{-1} = (d_i a_{ij} d_j^{-1})_1^n$. As A and $\hat{A}$ are similar A and $\hat{A}$ have the same spectrum. We next note that

$$\hat{A}\hat{u} = \rho(A)\hat{u}, \quad \hat{A}^T\hat{v} = \rho(A)\hat{v}, \quad \hat{u} = Du, \quad \hat{v} = D^{-1}v.$$

It now follows that $\epsilon(A, u, v) = \epsilon(\hat{A}, \hat{u}, \hat{v})$. Thus, it suffices to prove (5) for $\hat{A}$. Choose the unique D with positive diagonal entries so that $\hat{u} = \hat{v} = w$. Hence, without loss of generality, we may assume that $Aw = A^T w = \rho(A)w$, $w > 0$. Set $B = (A + A^T)/2 = (b_{ij})_1^n$, $b_{ii} = a_{ii}$, $i = 1, \dots, n$. Thus, B is a nonnegative irreducible symmetric matrix. As w is a positive eigenvector of B, we deduce that $\rho(B) = \rho(A)$. Furthermore, it is straightforward to check that $\epsilon(B, w, w) = \epsilon(A, w, w)$. Theorem 4.5 in [Fri] states that

$$\rho(B) - \lambda_{n-1}(B) \geq \frac{1}{2}(\rho(B) - \max_{1 \leq i \leq n} b_{ii})\chi(B, w)^2,$$

where

$$\chi(B, w) = \inf_{\emptyset \neq U \subset \{1,\dots,n\}, \operatorname{card}(U) \leq \lfloor \frac{n}{2} \rfloor} \frac{\sum_{i \in U, j \in \{1,\dots,n\} \setminus U} b_{ij} w_i w_j}{\sum_{i \in U, 1 \leq j = i \leq n} b_{ij} w_i w_j}.$$

As $Bw = \rho(B)w$ and B is symmetric, it is straightforward to show that $\chi(B, w) = \epsilon(B, w, w)$. Thus, to prove (5) it suffices to show that

$$\lambda_{n-1}(B) \geq \Re(\lambda_{n-1}(A)). \tag{7}$$

Let $w^\perp$ be the orthogonal complement of $\operatorname{span}(w)$ in $\mathbf{R}^n$. Thus, $Aw^\perp, A^T w^\perp, Bw^\perp \subset w^\perp$. Choose an orthonormal basis of $\mathbf{R}$ of the form $w_1 = w/\sqrt{w^T w}, w_2, \dots, w_n$. That is $w_2, \dots, w_n$ is an orthonormal basis of $w^\perp$. In the new basis $w_1, \dots, w_n$ the matrices A, B are represented by the matrices $(\rho(A)) \oplus A_1, (\rho(A)) \oplus B_1$ such that the following equalities hold:

$$A_1, B_1 \in M_{n-1}(\mathbf{R}), \quad B_1 = \frac{A_1 + A_1^T}{2}, \quad \lambda_j(A_1) = \lambda_j(A), \quad \lambda_j(B_1) = \lambda_j(B), \quad j = 1, \dots, n-1.$$

Recall that $\nu(A_1) = \{z = x^*A_1x : x \in \mathbf{C}^{n-1}, x^*x = 1\}$ is the numerical range of A_1. It is well known that $\nu(A_1)$ contains all the eigenvalues of A_1. As $\Re(x^*A_1x) = x^*B_1x$ we deduce that $\Re(\nu(A_1)) = \nu(B_1)$. Since B_1 is a real symmetric matrix its numerical range is $[\lambda_1(B_1), \lambda_{n-1}(B_1)]$. In particular, $\Re(\lambda_{n-1}(A_1)) \leq \lambda_{n-1}(B_1)$. The proof of the theorem is completed. □

REFERENCES

[Fri] S. FRIEDLAND, *Lower bounds for the first eigenvalue of certain M-matrices associated with graphs*, Linear Algebra Appl., 172 (1992), pp. 71–84.

[S-J] A. SINCLAIR AND M. JERRUM, *Approximate counting, uniform generation and rapidly mixing Markov chains*, Graph-theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci., 314, Springer-Verlag, Berlin, New York, 1988, pp. 134–138.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 1018–1034, July 1994

022

BLOCK DOWNDATING OF LEAST SQUARES SOLUTIONS*

L. ELDÉN† AND H. PARK‡

Abstract. This paper introduces new algorithms that extend the LINPACK downdating algorithm for a single row downdating, to downdating of a block of rows in an efficient way. The method of the corrected seminormal equations is then applied to the LINPACK-type block downdating algorithm to produce accurate downdated solutions. A sensitivity analysis of the Cholesky block downdating problem is presented. Based on this analysis, a hybrid algorithm is developed that has the advantages of the lower computational cost of the LINPACK-type algorithm and the higher accuracy of the corrected seminormal equation (CSNE) block downdating algorithm. Numerical test results comparing the accuracy of these three new block downdating algorithms for the recursive least squares sliding window method are presented.

Key words. block downdating, seminormal equations, iterative refinement, least squares, level 3 BLAS

AMS subject classifications. 65F20, 65F25

1. Introduction.

In linear least squares problems, we need to solve

$$\min_w \|Xw - s\|_2, \quad X \in \mathbf{R}^{p\times n}, \quad p > n. \tag{1.1}$$

If $\operatorname{rank}(X) = n$ and the QR decomposition of the data matrix $(X\ s)$ is

$$Q^T(X\ s) = \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{p\times(n+1)}, \tag{1.2}$$

where $Q \in \mathbf{R}^{p\times p}$ is orthogonal, then the least squares solution w is obtained from

$$Rw = u, \tag{1.3}$$

and the residual vector r and its norm satisfy

$$r = s - Xw, \qquad \|r\|_2 = |\rho|.$$

Frequently, one knows the factorization in (1.2) and wishes to find the solution to a modified problem

$$\min_w \|\tilde{X}w - \tilde{s}\|_2,$$

where a block of k new observations $(Y\ y) \in \mathbf{R}^{k\times(n+1)}$ is added (*block updating*):

$$\tilde{X} = \begin{pmatrix} X \\ Y \end{pmatrix}, \qquad \tilde{s} = \begin{pmatrix} s \\ y \end{pmatrix},$$

* Received by the editors September 14, 1992; accepted for publication (in revised form) March 10, 1993.

† Department of Mathematics, Linköping University, S–581 83 Linköping, Sweden (laeld@math.liu.se).

‡ Computer Science Department, University of Minnesota, Minneapolis, Minnesota 55455 (hpark@cs.umn.edu). The work of this author was supported in part by National Science Foundation grant CCR-9209726, and by contract DAAL02-89-C-0038 between the Army Research Office and the University of Minnesota for the Army High Performance Computing Research Center.

or a block of k old observations $(Z\ z) \in \mathbf{R}^{k\times(n+1)}$ is removed (*block downdating*):

$$X = \begin{pmatrix} Z \\ \tilde{X} \end{pmatrix}, \qquad s = \begin{pmatrix} z \\ \tilde{s} \end{pmatrix}.$$

Throughout this paper, we assume that the data matrices have full rank, i.e.,

$$\operatorname{rank}(X) = \operatorname{rank}(\tilde{X}) = n.$$

Often the modified problem involves both an updating and a downdating. From (1.3) we see that the solution to the modified problem can be obtained by modifying the R factor of the corresponding augmented matrix $(\tilde{X}\ \tilde{s})$. If R and $\tilde{R}$ are the R factors of X and $\tilde{X}$, respectively, then for updating we have

$$\tilde{R}^T\tilde{R} = R^TR + Y^TY$$

and for downdating

$$\tilde{R}^T\tilde{R} = R^TR - Z^TZ.$$

The case when $k = 1$ has been considered in several papers. For information concerning updating, see [13, p. 596]. Downdating for $k = 1$ has been studied, e.g., in [1], [4], [7], [10], [12], [15], [19]. Downdating a block using a variation of the Householder transformation is treated in [5], [6], [16].

In this paper we consider the downdating problem when $k > 1$, which we refer to as *block downdating*. The *block updating* problem is easy in the sense that a backward stable algorithm can be obtained by a straightforward generalization of the algorithm for the case $k = 1$ [16].

The LINPACK downdating algorithm due to Saunders [17] has been analyzed in [19]. A generalization of the LINPACK algorithm to block downdating can be found in [18]. Recently we developed accurate downdating methods for $k = 1$ based on the LINPACK algorithm combined with iterative refinement [4]. In this paper, we introduce generalizations of these methods for block downdating and compare their accuracies and computational complexities. In §2, we discuss some properties of the block downdating problem. In §3, we analyze the sensitivity of the Cholesky block downdating problem. Then we present two algorithms generalizing the LINPACK and CSNE algorithms for single row downdating [4], to handle block downdating in §§4 and 5, respectively. As in the case when only one row is downdated, it is possible to compromise between the LINPACK algorithm that is faster but less accurate when the downdating is ill conditioned, and the CSNE algorithm that has a higher computational complexity but better stability properties. Thus a hybrid method, which is an intermediary between the LINPACK-type algorithm and the CSNE algorithm with some significant virtues of both, is described in §6. Finally in §7, some numerical experiments are presented that show that the hybrid algorithm produces far more accurate solutions than the LINPACK-type algorithm when the problem is ill conditioned, with a modest increase in computational complexity.

2. Block downdating. Assume that the matrix $(X\ \ s) \in \mathbf{R}^{p\times(n+1)}$ has the QR decomposition

$$(X\ \ s) = \begin{pmatrix} Z & z \\ \tilde{X} & \tilde{s} \end{pmatrix} = Q \begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix}, \tag{2.1}$$

where $(\,Z \quad z\,) \in \mathbf{R}^{k\times(n+1)}$ is the block of k rows to be deleted and

$$p \geq k+n, \qquad n > k.$$

We first show that removing the block $(\,Z \quad z\,)$ is equivalent to updating the QR factorization of

$$(\,E_k \quad X \quad s\,) = \begin{pmatrix} I_k & Z & z \\ 0 & \tilde{X} & \tilde{s} \end{pmatrix},$$

where $E_k = \begin{pmatrix} I_k \\ 0 \end{pmatrix}$ and I_k is the $k \times k$ identity matrix. From (2.1), it follows that

$$Q^T(\,E_k \quad X \quad s\,) = \begin{pmatrix} Q_1 & R & u \\ q^T & 0 & \rho \\ Q_2 & 0 & 0 \end{pmatrix},$$

where $(Q_1^T \; q \; Q_2^T) \in \mathbf{R}^{k\times p}$ denotes the first k rows of Q. We can now determine an orthogonal matrix U that makes $Q^T(\,E_k \quad X \quad s\,)$ upper triangular, i.e.,

$$U^TQ^T(\,E_k \quad X \quad s\,) = U^T\begin{pmatrix} Q_1 & R & u \\ q^T & 0 & \rho \\ Q_2 & 0 & 0 \end{pmatrix} = \begin{pmatrix} I_k & V & f \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.2}$$

for some $V \in \mathbf{R}^{k\times n}$ and $f \in \mathbf{R}^{k\times 1}$. Then we have

$$\hat{Q}^T\begin{pmatrix} I_k & Z & z \\ 0 & \tilde{X} & \tilde{s} \end{pmatrix} = \begin{pmatrix} I_k & V & f \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.3}$$

where $\hat{Q} = QU$. Equating the first k columns on both sides of (2.3), we obtain $\hat{Q}^TE_k = E_k$, so the first k rows in $\hat{Q}$ are equal to those of E_k^T. Hence, $\hat{Q}$ must have the form

$$\begin{pmatrix} I_k & 0 \\ 0 & \tilde{Q} \end{pmatrix},$$

and it follows that $(V \; f) = (Z \; z)$. Dropping the first k rows and columns from (2.3) gives the downdated QR decomposition

$$(\,\tilde{X} \; \tilde{s}\,) = \tilde{Q}\begin{pmatrix} \tilde{R} & \tilde{u} \\ 0 & \tilde{\rho} \\ 0 & 0 \end{pmatrix}.$$

The new algorithms described in this paper are based on the above derivation. An important fact in *downdating the QR decomposition* is that the downdating transformation U is determined based on the first k rows of the square orthogonal factor Q. Thus, the first k rows $(\,Q_1^T \quad q \quad Q_2^T\,)$ of Q must be known or recovered to determine the downdating transformation U in (2.2). We consider only the case when the full orthogonal matrix Q is not available.

3. Sensitivity of the Cholesky block downdating problem. The sensitivity of the Cholesky downdating problem has been treated previously in [19], [14] for the case of a single row downdating ($k = 1$). We now discuss the sensitivity of the Cholesky block downdating problem to perturbations, with the purpose of motivating the hybrid algorithm that we present in §6. A more detailed investigation of sensitivity analysis is presented in [9]. The problem is formulated as that of downdating a Cholesky decomposition

$$\tilde{R}^T\tilde{R} = R^TR - Z^TZ, \tag{3.1}$$

and we first consider perturbations of the type

$$\bar{R}^T\bar{R} = (R+E)^T(R+E) - Z^TZ,$$

where E is the perturbation matrix, satisfying $||E||_2 \leq \epsilon$.

LEMMA 3.1. *Assume that $R^TR - Z^TZ$ is positive definite, and thus the Cholesky decomposition*

$$\tilde{R}^T\tilde{R} = R^TR - Z^TZ$$

exists. Then there is a matrix $\tilde{C} \in \mathbf{R}^{n\times n}$, such that

$$\tilde{R}^T\tilde{R} = R^TR - Z^TZ = R^T\tilde{C}^T\tilde{C}R, \tag{3.2}$$

where the singular values, $\sigma_i(\tilde{C})$, $1 \leq i \leq n$, of $\tilde{C}$ satisfy

$$\begin{aligned}\sigma_i(\tilde{C}) &= 1, \qquad i = 1,\ldots,n-k,\\ 1 \geq \sigma_{n-k+1}(\tilde{C}) &\geq \sigma_{n-k+2}(\tilde{C}) \geq \cdots \geq \sigma_n(\tilde{C}) > 0.\end{aligned}$$

Proof. We have

$$\tilde{R}^T\tilde{R} = R^T(I - Q_1Q_1^T)R,$$

where $Q_1 = R^{-T}Z^T \in \mathbf{R}^{n\times k}$. Since $\tilde{R}^T\tilde{R}$ is assumed to be positive definite, the same must be true of $I - Q_1Q_1^T$ and $\tilde{C}$ can be taken as the Cholesky factor of $I - Q_1Q_1^T$. Since $\tilde{C}^T\tilde{C}$ is a perturbation of the identity matrix of rank at most k, it has at least $n-k$ eigenvalues equal to one and the remaining eigenvalues are one minus the nonzero eigenvalues of $Q_1Q_1^T$. Since the eigenvalues of $Q_1Q_1^T$ are nonnegative, the corresponding eigenvalues of $\tilde{C}$ must lie in the half open interval $(0, 1]$. □

We can now prove a perturbation theorem for the Cholesky block downdating problem.

THEOREM 3.2. *Assume that $R^TR - Z^TZ$ is positive definite, and thus that the Cholesky decomposition $\tilde{R}^T\tilde{R} = R^TR - Z^TZ$ exists. Further assume that for a perturbation matrix E, satisfying $||E||_2 \leq \epsilon$, the Cholesky decomposition*

$$\bar{R}^T\bar{R} = (R+E)^T(R+E) - Z^TZ,$$

exists. Then there is a matrix $\bar{C} \in \mathbf{R}^{n\times n}$, such that

$$\bar{R}^T\bar{R} = R^T\bar{C}^T\bar{C}R,$$

and the eigenvalues of $\bar{C}^T\bar{C}$ satisfy

$$\lambda_i(\bar{C}^T\bar{C}) = \sigma_i^2(\tilde{C}) + \eta_i, \qquad i = 1,\ldots,n,$$

where $\sigma_i(\tilde{C})$ are the singular values of the matrix $\tilde{C}$ defined in (3.2), *the η_i are bounded by*

$$|\eta_i| \le \epsilon \left(\frac{2}{\sigma_n(R)} + \frac{\epsilon}{\sigma_n^2(R)} \right), \qquad i = 1, \ldots, n, \tag{3.3}$$

and $\sigma_n(R)$ is the smallest singular value of R.

Proof. We have

$$\bar{R}^T \bar{R} = (R+E)^T (R+E) - Z^T Z = \tilde{R}^T \tilde{R} + R^T (F^T + F + F^T F) R,$$

where $F = ER^{-1}$. From Lemma 3.1, we have $\tilde{R}^T \tilde{R} = R^T \tilde{C}^T \tilde{C} R$, for some matrix $\tilde{C}$, and $\bar{R}^T \bar{R} = R^T \bar{C}^T \bar{C} R$, where

$$\bar{C}^T \bar{C} = \tilde{C}^T \tilde{C} + F^T + F + F^T F.$$

Using classical perturbation theory for eigenvalues of symmetric matrices [13, p. 411], we get

$$|\lambda_i(\bar{C}^T \bar{C}) - \sigma_i^2(\tilde{C})| \le 2||F||_2 + ||F||_2^2 \le 2\epsilon ||R^{-1}||_2 + \epsilon^2 ||R^{-1}||_2^2.$$

Since $||R^{-1}||_2 = \sigma_n^{-1}$, (3.3) follows. □

The same inequality (3.3) can be obtained for a perturbation in Z using the technique of the above proof and the relations $Q_1 = R^{-T} Z^T$ and $||Q_1||_2 \le 1$; see [9].

The theorem shows the importance of the magnitude of the singular values $\sigma_i(\tilde{C})$, and it implies that $\bar{C}$ and $\tilde{C}$ can deviate in norm by a large amount. This can be seen as follows. Since, for nonnegative x and y, the inequality $x|x-y| \le |x+y||x-y|$ holds, we get from (3.3),

$$|\sigma_i(\bar{C}) - \sigma_i(\tilde{C})| \le \frac{\epsilon}{\sigma_i(\tilde{C})} \left(\frac{2}{\sigma_n} + \frac{\epsilon}{\sigma_n^2} \right).$$

Therefore, since $||\bar{C} - \tilde{C}||_2 \ge \max |\sigma_i(\bar{C}) - \sigma_i(\tilde{C})|$, we see that $||\bar{C} - \tilde{C}||$ can be as large as $\epsilon/\sigma_n(\tilde{C})(2/\sigma_n(R) + \epsilon/\sigma_n^2(R))$. Furthermore, if $\sigma_n(\tilde{C}) \le \sqrt{2/\sigma_n(R) + \epsilon/\sigma_n^2(R)}$, then we can expect that $\sigma_n(\bar{C})$ and $\sigma_n(\tilde{C})$ do not agree to any significant figures.

It is seen that we can take

$$\kappa_{\text{down}} = \max_i \{\sigma_i^{-2}(\tilde{C})\} = 1/\sigma_n^2(\tilde{C}), \tag{3.4}$$

as a measure of the conditioning of the block downdating problem. This is a generalization of the results by Stewart for a single row downdating [19].

The arguments in [19] that we referred to above are concerned with the singular values of $\tilde{R}$ (and not $\tilde{C}$) for the special case of a single row downdating ($k = 1$). It is shown in [19] that if *any* singular value of R is reduced (to a singular value of $\tilde{R}$) by a considerable amount, then the downdating problem is ill conditioned. Therefore, to ascertain the downdating conditioning, all of the singular values of R and $\tilde{R}$ must be examined, and it is inefficient to numerically estimate the conditioning this way.

In contrast, the singular values of $\tilde{C}$ give clear information about the reduction of quantities from R to $\tilde{R}$, i.e., a small $\sigma_i(\tilde{C})$ represents a quantity that has been reduced from one in R, and thus it signals that the downdating problem is ill conditioned. Later, in §6 we demonstrate how downdating condition estimation based on κ_{down} can be implemented efficiently.

If κ_{down} is large, then the downdating problem is ill conditioned. Also, if the condition number of R is large, then the computation of the matrix Q_1 via $R^{-T}Z^T$ is sensitive to errors, and so is the downdating problem. In (3.3), $\sigma_n(R)$ appears in the denominator, and if $\sigma_n(R)$ is small, then the bound in (3.3) for $|\eta_i|$ is large. However, the following example shows that the downdating problem can be ill conditioned, even if $\sigma_n(R)$ is not small.

Example. Let

$$X = \begin{pmatrix} \tau & 0 & 0 \\ 0 & \tau & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where $\tau >> 1$, and let Z consist of the first two rows of X. Then

$$R = \begin{pmatrix} \sqrt{\tau^2+1} & 0 & 0 \\ 0 & \sqrt{\tau^2+1} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and $\sigma_3 = 1$. From

$$Q_1 = \begin{pmatrix} \tau/\sqrt{\tau^2+1} & 0 \\ 0 & \tau/\sqrt{\tau^2+1} \\ 0 & 0 \end{pmatrix},$$

it follows that $c_1 = 1$, $c_2 = c_3 = 1/\sqrt{\tau^2+1}$, and $\kappa_{\text{down}} = \tau^2 + 1$, which indicates that this downdating problem is ill conditioned. In fact, downdating from R to $\tilde{R}$ is here equivalent to computing the diagonal elements in $\tilde{R}$ from the formula

$$\tilde{r} = \sqrt{r^2 - \tau^2}, \qquad r^2 = \tau^2 + 1.$$

Consider the perturbed problem

$$\bar{r} = \sqrt{(r+e)^2 - \tau^2}, \quad |e| \le \epsilon,$$

where ϵ is small. A simple computation shows that

$$\bar{r} \approx \tilde{r}\left(1 + \frac{r}{\tilde{r}^2}\, e\right) \approx \tilde{r}(1 + \tau e),$$

which shows that the problem is ill conditioned. If $\tau > 1/\sqrt{\mu}$, where μ is the unit round-off of the floating point system, then the downdating will fail completely, since $\tilde{R}$ will be computed as

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where all information from the first two rows of $\tilde{X}$ is lost.

For the case $k = 1$ we have $\kappa_{\text{down}} = 1/\sigma_1(\tilde{C})^2$, which is the same as $1/(1 - ||q_1||_2^2)$, where q_1 is the vector consisting of the first n elements of the first row of the orthogonal matrix Q, see §2. This is essentially the quantity discussed in [19] and [4] as an indicator of ill-conditioning for the downdating problem in the special case $k = 1$.

4. Generalization of the LINPACK algorithm. In §2, we showed that to downdate the first k rows of the data matrix, we need the first k rows, $(\,Q_1^T \quad q \quad Q_2^T\,) \in \mathbf{R}^{k\times p}$, of the orthogonal matrix Q. From (2.1), we have

$$(\,Z \quad z\,) = (\,Q_1^T \quad q \quad Q_2^T\,)\begin{pmatrix} R & u \\ 0 & \rho \\ 0 & 0 \end{pmatrix} = (\,Q_1^T \quad q\,)\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix}.$$

It follows that Q_1 and q can be computed by solving the triangular system

$$\begin{pmatrix} R^T & 0 \\ u^T & \rho \end{pmatrix}\begin{pmatrix} Q_1 \\ q^T \end{pmatrix} = \begin{pmatrix} Z^T \\ z^T \end{pmatrix},$$

which gives $Q_1 = R^{-T}Z^T$. Using the relation $u^TQ_1 = u^TR^{-T}Z^T = w^TZ^T$, where w is the solution of the least squares problem (1.1), we obtain

$$q = (z - Zw)/\rho, \qquad (\rho \neq 0). \tag{4.1}$$

If $\rho = 0$ then q cannot be computed from (4.1). However, it is seen below that the computations can be arranged so that the assumption $\rho \neq 0$ is not needed.

Now we need to determine a matrix Q_2 that satisfies

$$Q_1^TQ_1 + qq^T + Q_2^TQ_2 = I_k. \tag{4.2}$$

The orthogonal transformation U in (2.2) can be chosen in such a way that Q_2 is first reduced to upper triangular form without changing R, u, or ρ. Therefore *it is sufficient to determine an upper triangular matrix* $\bar{\Gamma} \in \mathbf{R}^{k\times k}$ *that satisfies*

$$Q_1^TQ_1 + qq^T + \bar{\Gamma}^T\bar{\Gamma} = I_k.$$

This can be done simply by computing a Cholesky decomposition

$$\bar{\Gamma} = \mathrm{chol}(I_k - Q_1^TQ_1 - qq^T),$$

where $\bar{\Gamma}$ is upper triangular and $\bar{\Gamma}^T\bar{\Gamma} = I_k - Q_1^TQ_1 - qq^T$. Next we determine a product of plane rotations U such that

$$U^T\begin{pmatrix} Q_1 & R & u \\ q^T & 0 & \rho \\ \bar{\Gamma} & 0 & 0 \end{pmatrix} = \begin{pmatrix} I_k & Z & z \\ 0 & \tilde{R} & \tilde{u} \\ 0 & 0 & \tilde{\rho} \end{pmatrix}. \tag{4.3}$$

We can initially choose an orthogonal matrix J that affects only rows from $(n+1)$ to $(n+k+1)$ of (4.3)

$$J\begin{pmatrix} q^T & \rho \\ \bar{\Gamma} & 0 \end{pmatrix} = \begin{pmatrix} \Gamma & h \\ 0 & \tilde{\rho} \end{pmatrix}, \tag{4.4}$$

where $\Gamma \in \mathbf{R}^{k\times k}$ is upper triangular. Then, since $\Gamma^T\Gamma = qq^T + \bar{\Gamma}^T\bar{\Gamma}$, we can obtain the upper triangular matrix Γ from

$$\Gamma = \mathrm{chol}(I_k - Q_1^TQ_1). \tag{4.5}$$

From (4.1) and (4.4), we obtain

$$h = \Gamma^{-T}(z - Q_1^Tu), \qquad \tilde{\rho} = (\rho^2 - h^Th)^{1/2}. \tag{4.6}$$

Note that q in (4.1) is not used, so the assumption that $\rho \neq 0$ is not needed. If $I_k - Q_1^T Q_1$ is not positive definite, then the downdating procedure would break down. This is equivalent to the case when $\tilde{X}^T \tilde{X}$ is not positive definite. Consider the Cholesky downdating problem

$$\tilde{R}^T \tilde{R} = R^T R - Z^T Z = R^T (I_n - Q_1 Q_1^T) R,$$

where $Q_1 = R^{-T} Z^T$. The matrix $\tilde{R}^T \tilde{R}$ is positive definite if and only if all the eigenvalues of $I_n - Q_1 Q_1^T$ are positive, in which case all the eigenvalues of $I_k - Q_1^T Q_1$ are also positive (the former matrix has additional $n - k$ eigenvalues that are equal to one). Since we have assumed that $\operatorname{rank}(\tilde{X}) = n$, $I_k - Q_1^T Q_1$ is positive definite and Γ is nonsingular. Thus, under the assumption that $\operatorname{rank}(\tilde{X}) = n$ the algorithm is well defined. Summarizing these results we get the following algorithm.

ALGORITHM BDLIN: LINPACK-type algorithm for block downdating

Given R, u, ρ, w, and $(Z \;\; z)$, the following algorithm computes the downdated quantities $\tilde{R}, \tilde{u}, \tilde{\rho}$ and $\tilde{w}$:

1. (a) Compute Q_1 and Γ from

$$R^T Q_1 = Z^T, \qquad \Gamma := chol(I_k - Q_1^T Q_1).$$

   (b) Compute h from

$$\Gamma^T h = (z - Q_1^T u).$$

2. Apply a product of Givens rotations, $\hat{U}$, such that

$$\begin{pmatrix} I_k & Z & z \\ 0 & \tilde{R} & \tilde{u} \end{pmatrix} := \hat{U}^T \begin{pmatrix} Q_1 & R & u \\ \Gamma & 0 & h \end{pmatrix}.$$

3. Compute the new solution $\tilde{w}$ and the residual norm from

$$\tilde{R}\tilde{w} = \tilde{u}, \qquad \tilde{\rho} := (\rho^2 - h^T h)^{1/2}.$$

In Step 2, we assume that the sequence of Givens rotations in the adjacent planes is generated in the order of annihilating the elements $(n+1,1), (n,1), \ldots, (2,1), (n+2,2), (n+1,2), \ldots, (3,2)$, etc., of $\binom{Q_1}{\Gamma}$. Assuming that $k < n$, this algorithm takes the total of approximately $2.5n^2k$ flops (one flop is taken to be one multiplication and one addition). For $k = 1$, the Saunders LINPACK algorithm requires $3n^2$ flops; thus applying it k times for downdating k rows would require about $3n^2k$ flops. However, by a simple modification of the Saunders LINPACK algorithm for the $k = 1$ case, we can avoid solving a triangular system each time and reduce the complexity of single row downdating to $2.5n^2$. With this modification, the above algorithm has about the same computational complexity as applying LINPACK algorithm for a single row downdating k times.

5. Block downdating using seminormal equations. As in the case of $k = 1$ [4], we can improve the accuracy of the solution vector for the downdated least squares problem by applying the method of CSNE. The seminormal equations (SNE) for solving a least squares problem $\min_w \|Xw - s\|_2$ are defined as

$$R^T R w = X^T s,$$

where R is the upper triangular factor in the QR decomposition of X, which we denote as

$$R = qr(X).$$

Applying one step of iterative refinement on the solution computed from the SNE, we have the method of CSNE

$$\begin{aligned} R^T R w &= X^T s, \qquad r = s - Xw, \\ R^T R \delta w &= X^T r, \quad w^c = w + \delta w, \quad r^c = r - X\delta w. \end{aligned} \tag{5.1}$$

We assume that *all computations* are performed in *single precision.* For details, see [3].

The following theorem shows how the CSNE method can be used to recover the rows in the orthogonal matrix Q more accurately based on which the downdating transformation is determined.

THEOREM 5.1. *Let V be the solution to*

$$\min_V \|E_k - XV\|_F, \qquad E_k = \begin{pmatrix} I_k \\ 0 \end{pmatrix}. \tag{5.2}$$

Then the R factor of $(X\ E_k)$ is

$$\begin{pmatrix} R & Q_1 \\ 0 & \Gamma \end{pmatrix}, \tag{5.3}$$

where the upper triangular matrix $\Gamma \in \mathbf{R}^{k\times k}$ is the same as the upper triangular factor of $qr(E_k - XV)$. The matrix V satisfies the equation $Q_1 = RV$.

Applying the CSNE method to (5.2), we obtain

$$R^T Q_1 = Z^T, \quad RV = Q_1, \quad T = E_k - XV, \tag{5.4}$$

$$R^T \delta Q_1 = X^T T, \qquad Q_1^c = Q_1 + \delta Q_1, \tag{5.5}$$

$$R\delta V = \delta Q_1, \qquad T^c = T - X\delta V \tag{5.6}$$

and more accurate Γ (5.3) can be computed as the R factor in the QR decomposition of T^c. We can also apply the CSNE method to downdate the *augmented* upper triangular factor for solving the least squares problem

$$\min_{Y,f} \|E_k - (X\ s)\begin{pmatrix} Y \\ f^T \end{pmatrix}\|_F. \tag{5.7}$$

In this case, we have

$$\begin{pmatrix} R & u & Q_1 \\ 0 & \rho & q^T \\ 0 & 0 & \bar{\Gamma} \end{pmatrix} = qr((X \quad s \quad E_k)), \tag{5.8}$$

where $u \in \mathbf{R}^{n\times 1}$, $\rho \in \mathbf{R}$, $q \in \mathbf{R}^{k\times 1}$, and $\bar{\Gamma} \in \mathbf{R}^{k\times k}$. First, from (5.7) and (5.8), we obtain

$$\begin{pmatrix} R^T & 0 \\ u^T & \rho \end{pmatrix}\begin{pmatrix} Q_1 \\ q^T \end{pmatrix} = \begin{pmatrix} Z^T \\ z^T \end{pmatrix},$$

which gives

$$Q_1 = R^{-T}Z^T, \qquad q = (z - Q_1^T u)/\rho = (z - Zw)/\rho \tag{5.9}$$

with the assumption that $\rho \neq 0$. This assumption is eliminated in the following. Next we solve

$$\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix} \begin{pmatrix} Y \\ f^T \end{pmatrix} = \begin{pmatrix} Q_1 \\ q^T \end{pmatrix},$$

which gives

$$f = q/\rho, \qquad Y = R^{-1}(Q_1 - uf^T) = V - wf^T.$$

The residual matrix T_a for the augmented problem (5.7) is

$$T_a = E_k - (X\ s)\begin{pmatrix} Y \\ f^T \end{pmatrix} = E_k - XV - (s - Xw)q^T/\rho = T - (s - Xw)q^T/\rho$$

and

$$\bar{\Gamma} = qr(T_a).$$

The equation

$$\begin{pmatrix} R^T & 0 \\ u^T & \rho \end{pmatrix} \begin{pmatrix} \delta Q_{1a} \\ \delta q^T \end{pmatrix} = \begin{pmatrix} X^T \\ s^T \end{pmatrix} T_a \tag{5.10}$$

is analogous to the first equation in (5.5). Since $X^T(s - Xw) = 0$, we have $X^T T_a = X^T T$, and therefore $\delta Q_{1a} = \delta Q_1$, where δQ_1 is defined by (5.5). From (5.10) we get

$$\delta q = T_a^T(s - Xw)/\rho.$$

Similarly, from

$$\begin{pmatrix} R & u \\ 0 & \rho \end{pmatrix} \begin{pmatrix} \delta Y \\ \delta f^T \end{pmatrix} = \begin{pmatrix} \delta Q_1 \\ \delta q^T \end{pmatrix},$$

which is analogous to the first equation in (5.6), we obtain

$$\delta f = \delta q/\rho, \quad \delta Y = \delta V - w\delta q^T/\rho, \quad T_a^c = T_a - X\delta V - (s - Xw)\delta q^T/\rho.$$

As in the generalization of the LINPACK algorithm for block downdating, we can choose an orthogonal matrix J to make $J\left(\begin{smallmatrix} q^T & \rho \\ \bar{\Gamma} & 0 \end{smallmatrix}\right)$ upper triangular, i.e.,

$$J\begin{pmatrix} q^T & \rho \\ \bar{\Gamma} & 0 \end{pmatrix} = \begin{pmatrix} \Gamma & h \\ 0 & \tilde{\rho} \end{pmatrix}, \tag{5.11}$$

which gives

$$h = \rho\Gamma^{-T}q, \qquad \tilde{\rho}^2 = \rho^2 - h^T h = \rho^2(1 - q^T\Gamma^{-1}\Gamma^{-T}q). \tag{5.12}$$

By applying the Sherman–Morrison formula [13] to

$$qq^T + \bar{\Gamma}^T\bar{\Gamma} = \Gamma^T\Gamma, \tag{5.13}$$

we have

$$q^T\Gamma^{-1}\Gamma^{-T}q = q^T\bar{\Gamma}^{-1}\bar{\Gamma}^{-T}q/(1+q^T\bar{\Gamma}^{-1}\bar{\Gamma}^{-T}q) \tag{5.14}$$

and

$$\tilde{\rho}^2 = \rho^2(1-q^T\Gamma^{-1}\Gamma^{-T}q) = \rho^2/(1+q^T\bar{\Gamma}^{-1}\bar{\Gamma}^{-T}q) = \rho^2||\Gamma^{-T}q||_2^2/||\bar{\Gamma}^{-T}q||_2^2. \tag{5.15}$$

We now summarize the above derivations in the following algorithm.

ALGORITHM BDCSNE: Block downdating using CSNE.

Given R, u, ρ, w, and the data $(X\ s)$ the following algorithm deletes the first k rows $(Z\ z)$ of $(X\ s)$ and computes the downdated quantities $\tilde{R}, \tilde{u}, \tilde{\rho}$ and $\tilde{w}$:

1. Compute Q_1, V, and T from

$$(a)\ R^TQ_1 = Z^T, \quad (b)\ RV = Q_1, \quad (c)\ T := E_k - XV.$$

2. (a) Update Q_1, V, and T:

$$\begin{aligned} R^T\delta Q_1 &= X^TT, & Q_1 &:= Q_1 + \delta Q_1, \\ R\delta V &= \delta Q_1, & T &:= T - X\delta V. \end{aligned}$$

(b) Compute a QR decomposition of T to determine Γ:

$$\Gamma = qr(T).$$

3. Set $h := 0$, $\tilde{\rho} := 0$
   $r = s - Xw$; $\rho = ||r||_2$
   If $fl(1+\rho) \neq 1$,
   (a) compute the normalized residual: $r := (s - Xw)/\rho$,
   (b) modify T: $q := E_k^Tr$, $T := T - rq^T$,
   (c) update q and T: $\delta q := T^Tr$, $q := q + \delta q$, $T := T - r\delta q^T$,
   (d) compute h from $\Gamma^Th = \rho q$,
   (e) compute $\bar{\Gamma} = qr(T)$,
   (f) determine y from $\bar{\Gamma}^Ty = q$, and compute $\tilde{\rho} = ||h||_2/||y||_2$.
4. Determine an orthogonal matrix U^T as a product of Givens rotations such that

$$\begin{pmatrix} I_k & Z & z \\ 0 & \tilde{R} & \tilde{u} \end{pmatrix} := U^T \begin{pmatrix} Q_1 & R & u \\ \Gamma & 0 & h \end{pmatrix}.$$

5. Compute the new solution $\tilde{w}$ from

$$\tilde{R}\tilde{w} = \tilde{u}.$$

Applying the CSNE algorithm for a single row downdating requires approximately $4pn + 4.5n^2$ flops [4]. Thus, if we downdate k rows by applying the CSNE algorithm for a single row downdating k times, the computational complexity becomes about $4kpn + 4.5kn^2$ flops. Assuming that $k < n$ and $n < p$, the above algorithm takes a total of approximately $(3k+1)pn + 4kn^2$ flops. This algorithm is rich in level 3 BLAS [8] operations: solution of triangular systems with k right-hand sides, matrix–matrix multiplications. Therefore, it should execute efficiently on vector and parallel computers.

6. A hybrid algorithm. For a single row downdating, the CSNE algorithm gives much better accuracy than the LINPACK algorithm when the downdating problem is ill conditioned [4], but the computational complexity of the CSNE algorithm is considerably higher than that of the LINPACK algorithm. We have recently developed a hybrid algorithm for a single row downdating [4], and have shown that it produces accurate solutions that are comparable to those of the CSNE algorithm with lower computational cost. In the hybrid algorithm, the CSNE algorithm is used if the downdating is ill conditioned and the LINPACK algorithm is used otherwise. Thus, the hybrid algorithm will be a competitive alternative if a good indication of the conditioning of the downdating problem is available so that the iterative refinement is used only when it is necessary. We now introduce a hybrid algorithm for block downdating that is a combination of the block LINPACK-type algorithm and the block CSNE algorithm we presented in the previous sections.

Before we develop the hybrid algorithm, we state the following two properties of the block downdating that are essential in our hybrid algorithm.

First, from the sensitivity analysis of the block downdating problem presented in §3, we know that if any of the singular values of $\tilde{C}$ is small, where $\tilde{C}$ is the Cholesky factor of $I - Q_1 Q_1^T$, then the downdating problem is ill conditioned. Since Γ, which is the Cholesky factor of $I - Q_1^T Q_1$, has the same singular values as $\tilde{C}$, apart from a number of singular values equal to one, we can determine the downdating conditioning by estimating the singular values of Γ.

Second, since we have

$$X = \begin{pmatrix} Z \\ \tilde{X} \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{6.1}$$

where $Z \in \mathbf{R}^{k \times n}$, applying a permutation on the first k rows of the matrices X and Q does not change the block downdating problem mathematically. In other words, the result of downdating a block of the first k rows is not affected by a permutation on these k rows. Thus, removing the block $(\, Z \quad z \,)$ is also equivalent to updating the QR decomposition of

$$(\, P_k \quad X \quad s \,) = \begin{pmatrix} \Pi_k & Z & z \\ 0 & \tilde{X} & \tilde{s} \end{pmatrix}, \qquad P_k = \begin{pmatrix} \Pi_k \\ 0 \end{pmatrix},$$

where $\Pi_k \in \mathbf{R}^{k \times k}$ is any permutation matrix. Then from (2.1), it follows that

$$Q^T (\, P_k \quad X \quad s \,) = \begin{pmatrix} Q_1 \Pi_k & R & u \\ q^T \Pi_k & 0 & \rho \\ Q_2 \Pi_k & 0 & 0 \end{pmatrix}.$$

Proceeding as in §2, we find an orthogonal downdating transformation that makes $Q^T (\, P_k \quad X \quad s \,)$ upper triangular.

In our hybrid algorithm, we incorporate a diagonal pivoting strategy when we compute the upper triangular matrix Γ as a Cholesky factor of $I_k - Q_1^T Q_1$ (see Step 1(a) in Algorithm BDLIN): in the first step of the Cholesky decomposition algorithm, we permute the rows and columns of $I_k - Q_1^T Q_1$ so that the largest diagonal element is moved to position (1,1), and similarly in the subsequent steps. This is mathematically (but not numerically) equivalent to computing the matrix R in the QR decomposition with column pivoting, see [13, §5.4]. By using diagonal pivoting, we permute and partition the block $(\, Z \quad z \,)$ into two parts so that we downdate the better conditioned

first block of rows by the less costly algorithm, and apply one step of the iterative refinement to the ill-conditioned block of the rest of the rows to improve the accuracy.

Specifically, assume that after i steps of the Cholesky decomposition procedure on $I_k - Q_1^T Q_1$, we have computed the partial decomposition

$$\Pi^{(i)^T} \cdots \Pi^{(1)^T} (I_k - Q_1^T Q_1) \Pi^{(1)} \cdots \Pi^{(i)} = \begin{pmatrix} \Gamma_{11}^{(i)} & \Gamma_{12}^{(i)} \\ 0 & \Gamma_{22}^{(i)} \end{pmatrix}^T \begin{pmatrix} \Gamma_{11}^{(i)} & \Gamma_{12}^{(i)} \\ 0 & \Gamma_{22}^{(i)} \end{pmatrix},$$

where $\Pi^{(i)}$'s are permutation matrices and $\Gamma_{11}^{(i)} \in \mathbf{R}^{i \times i}$ is upper triangular. The purpose of the pivoting is to find the maximal triangular matrix $\Gamma_{11}^{(i)}$ for which the corresponding block of rows of $(\, Z \quad z \,)$ constitute a well-conditioned downdating problem. We use the incremental condition estimator (ICE) [2] to get an estimate $\hat{\sigma}^{(i)}$ of the smallest singular value of $\Gamma_{11}^{(i)}$ in each step of the Cholesky procedure, and if $\hat{\sigma}^{(i)} > \text{tol}$ for some given tolerance "tol," we conclude that the downdating problem so far is well-conditioned enough to be handled using Algorithm BDLIN. The first time when $\hat{\sigma}^{(i)} > \text{tol}$ and $\hat{\sigma}^{(i+1)} < \text{tol}$, the downdating problem is partitioned into two blocks between rows i and $i+1$, and we apply Algorithm BDLIN to the block of first i rows after the permutation determined from the pivoting and Algorithm BDCSNE to the rest.

ALGORITHM BDHYB: Hybrid block downdating algorithm.

Given R, u, ρ, w, and $(X \; s)$, the following algorithm deletes the first k rows $(Z \; z)$ of $(X \; s)$ and computes the downdated quantities $\tilde{R}, \tilde{u}, \tilde{\rho}$ and $\tilde{w}$:

1. Compute Q_1 from

$$R^T Q_1 = Z^T.$$

2. Compute a Cholesky decomposition with diagonal pivoting of $I_k - Q_1^T Q_1$ until the last index i such that $\hat{\sigma}^{(i)}(\Gamma_{11}) > \text{tol}$ is found, where $\Gamma_{11} \in R^{i \times i}$, $1 \le i \le k$, and

$$\Pi_k^T (I_k - Q_1^T Q_1) \Pi_k = \begin{pmatrix} \Gamma_{11} & \bar{\Gamma}_{12} \\ 0 & \bar{\Gamma}_{22} \end{pmatrix}^T \begin{pmatrix} \Gamma_{11} & \bar{\Gamma}_{12} \\ 0 & \bar{\Gamma}_{22} \end{pmatrix}.$$

Permute the rows of X and s (including Z and z), and the columns of Q_1:

$$X(1:k,:) := \Pi_k^T X(1:k,:); \quad s(1:k) := \Pi_k^T s(1:k), \quad Q_1 := Q_1 \Pi_k$$

3. If $i = k$ (use LINPACK-type algorithm for the whole block),
   Perform Steps 1(b)–3 of Algorithm BDLIN
   else if $i = 0$ (use CSNE algorithm for the whole block),
   Perform Steps 1(b)–5 of Algorithm BDCSNE
   else
   (a) (Separate the data that are needed for downdating the first i rows of $(Z \; z)$.)

$$Q_1 := Q_1(:, 1:i), \quad \Gamma := \Gamma_{11}, \quad z := z(1:i).$$

   (b) (Downdate rows 1 to i of $(Z \; z)$)
   Perform Steps 1(b) – 3 of Algorithm BDLIN, giving downdated $\check{R}$, $\check{u}$, $\check{\rho}$, and $\check{w}$.

(c) (Downdate rows $i+1$ to k of Z)
Take $\check{R}$, $\check{u}$, $\check{\rho}$, and $\check{w}$ from step (b), and rows $i+1$ to k of Z as input and perform Algorithm BDCSNE.

We remark that it is also possible to replace Step 1 in Algorithm BDHYB by Step 1 in Algorithm BDCSNE. Then Γ_{11} can be obtained from a QR decomposition with column pivoting [13, §5.4] of the residual matrix T instead of a Cholesky decomposition with diagonal pivoting of $I_k - Q_1^T Q_1$ in Step 2, since $T^T T = I_k - Q_1^T Q_1$. This variant should have slightly better stability properties than the usual LINPACK-type block downdating algorithm since the data matrix X is used in the computation of T. We do not pursue this here, however, since Algorithm BDHYB performed well in our tests and since it is more efficient.

The computational complexity of Algorithm BDHYB is between $2.5kn^2$ (when the LINPACK-type algorithm is applied for the whole block) and $3kpn + 4kn^2$ (when the CSNE-type algorithm is used for the whole block), and in general, it depends on the index where the block of rows is partitioned. If p is much larger than n, then we can expect the hybrid algorithm to be considerably faster than the CSNE algorithm. The ICE only requires $3i^2/2$ approximately, where i is the number of rows to be downdated by Algorithm BDLIN. Hence its contribution to the overall computational complexity is minimal. The numerical experimental results presented in the next section show that the accuracy of the above algorithm is far superior to that of Algorithm BDLIN and the additional cost is modest.

7. Numerical experiments. In a sliding window method, a least squares solution is computed based on the p latest rows of an observation matrix A, where p is the number of rows in the window matrix [1]. In each step, a new block of k rows of observations, is updated into the QR decomposition, and an existing block of k rows of the data matrix is downdated from the decomposition, on a first in, first out basis.

Numerical tests for the recursive least squares method using the sliding window method have been performed in Pro-Matlab with IEEE double precision floating point arithmetic to compare the accuracy of the block downdating algorithms that have been presented. The solution obtained from the QR decomposition of the window matrix was used as a reference. In each figure, we present the relative error in Euclidean norm in the downdated solution vector produced by algorithms BDLIN, BDCSNE, and BDHYB. A measure of the conditioning of the downdating of the whole block of k rows, $1/\sigma_k^2(\Gamma)$, and the spectral condition number κ_X of the window matrix to be downdated are shown. To illustrate the conditioning of the subproblem that is treated by Algorithm BDLIN in the hybrid algorithm, we give estimates $1/(\hat{\sigma}^{(i)})^2$ (obtained from ICE) of $1/\sigma_i^2(\Gamma_{11}^{(i)})$. The digits in the plot denote the size of the row block that was downdated in the hybrid algorithm using Algorithm BDLIN. We have used the following criterion in the hybrid method: if i is the last index such that $\hat{\sigma}^{(i)} > \text{tol}$, then Algorithm BDCSNE is applied to the part that starts from the row $i+1$. The approximation $\hat{\sigma}^{(i)}$ of the smallest singular values of $\Gamma_{11}^{(i)}$ was computed using ICE [2] and tol was chosen to be 0.5.

The following two test problems are similar to those in [4] and [10], which we used earlier to compare the accuracy of several downdating algorithms when $k = 1$. They were also used in the context of adaptive condition number estimation in [11].

Test I. A random matrix $A \in \mathbf{R}^{210\times 12}$ was constructed with elements taken from a uniform distribution in $(0,1)$. An outlier equal to $6 \cdot 10^3$ was added in position (34,3). The right-hand side vector b was taken to be $b = Ax_0 + b_r$, where b_r has

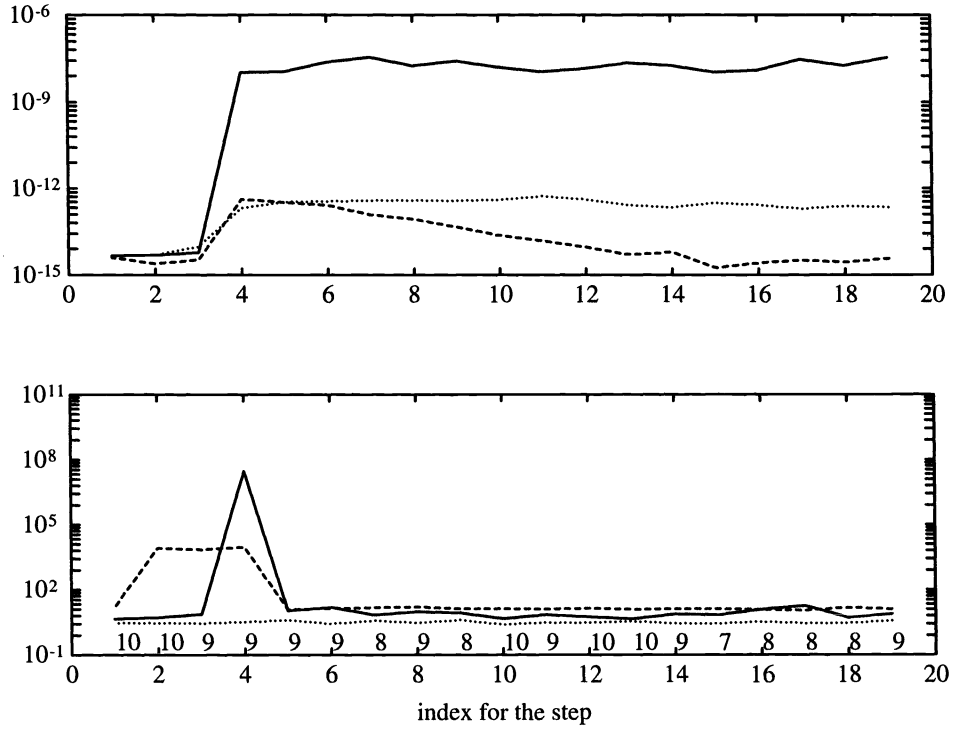


FIG. 7.1. *Test* I. *The upper graph shows the relative error in Euclidean norm in the downdated solution vector by the* BDLIN (*solid line*), BDCSNE (*dashed*), *and* BDHYB (*dotted*) *algorithms. The lower graph shows the condition number* κ_X *of the window matrix to be downdated* (*dashed*), $1/\sigma_k^2(\Gamma)$ (*solid line*), *and the* ICE *estimate* $1/(\hat{\sigma}^{(i)})^2$ (*dotted*). *The digits in the graph show the number of rows* (*in the block of* 10 *rows*) *treated by Algorithm* BDLIN *in the hybrid method.*

random elements uniformly distributed in $(0, 10^{-6})$ and x_0 is 12×1 vector with ones as its components. The window size p is 20 and ten rows are added and deleted each time, i.e., $k = 10$.

The results are shown in Fig. 7.1. It is seen that the relative error in the solution using Algorithm BDLIN is considerably magnified in the ill-conditioned downdating step and that it remains on that high level even if the subsequent downdating problems are well conditioned. The BDCSNE and BDHYB algorithms are much less affected by the ill-conditioned downdating and the errors remain on a low level throughout.

The digits in the plot show that in Algorithm BDHYB the major part of each downdating was performed using Algorithm BDLIN. In fact, over 88% of the total number of rows was downdated using Algorithm BDLIN. This shows that the hybrid algorithm can be much more efficient than Algorithm BDCSNE and can produce solutions that are far more accurate than those from Algorithm BDLIN.

Test II. A 76×6 matrix was constructed by taking a 38×6 Hilbert matrix as the first 38 rows and the same rows in reversed order as the 38 last rows. Then a perturbation from a uniform distribution in $(0, 10^{-5})$ was added to each matrix element. The right-hand side was constructed as in Test I, $p = 8$ and $k = 4$.

The results are shown in Fig. 7.2. Throughout this test, the downdating problem

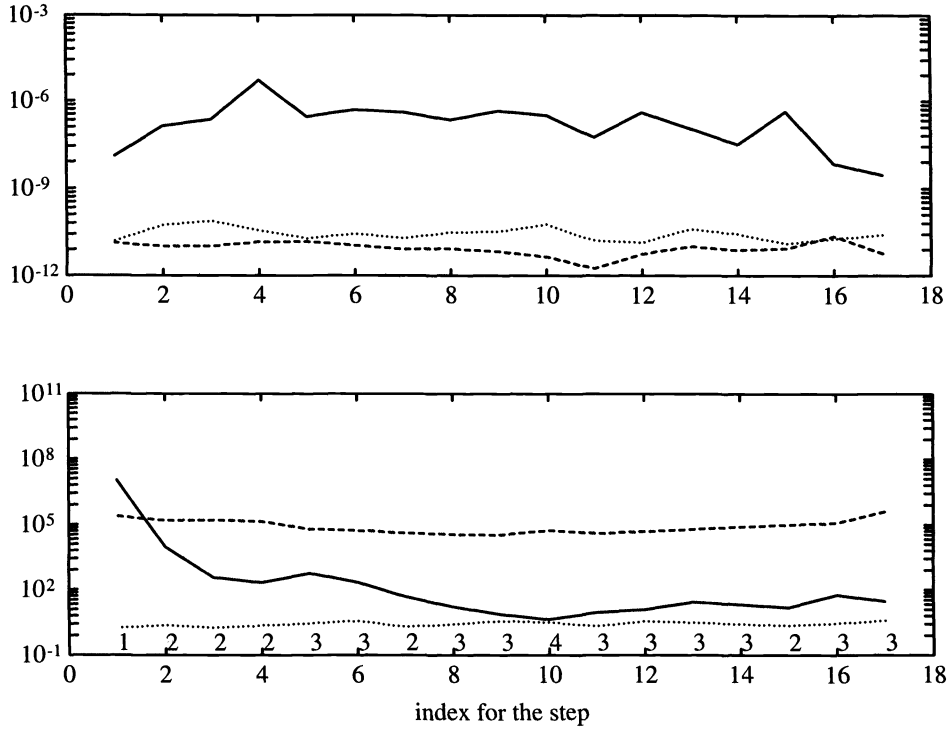


FIG. 7.2. *Test* II. *Modified Hilbert matrix with perturbations from a uniform distribution in* $(0, 10^{-5})$.

is rather ill conditioned, but even so, the CSNE branch of the hybrid algorithm is used only for one third of the total number of rows downdated. It is remarkable that Algorithm BDLIN performs so much worse than the others. This is probably due to the fact that the window matrix is very ill conditioned, which leads to large errors in the computed approximations of Q_1. In Algorithm BDCSNE, this vector is refined and much better accuracy is attained.

8. Conclusion. In this paper, we have considered generalizations of the LINPACK and CSNE algorithms for single row downdating to handle block downdating. The block downdating algorithms are rich in level 3 BLAS operations and this makes them amenable to efficient implementation on vector and parallel computers. The sensitivity of the block downdating problem to perturbations has been analyzed and a method of estimating the conditioning of the downdating problem has been devised. Based on this analysis we have developed a hybrid method in which the more efficient LINPACK-type algorithm is used for the well-conditioned part of the block downdate and the more expensive CSNE algorithm is used for the ill-conditioned part.

Preliminary numerical experiments indicate that the block downdating LINPACK-type and CSNE algorithms have properties that are similar to those of their single row counterparts, i.e., the latter is much more accurate (but considerably more expensive) than the former. The hybrid method has accuracy comparable to that of the CSNE algorithm, and it is almost as fast as the LINPACK-type algorithm. Therefore

it appears to be very promising for applications, where both speed and accuracy are essential.

Acknowledgments. The second author would like to thank the Department of Mathematics, Linköping University, Linköping, Sweden, where part of this work was carried out in August 1992. The first author acknowledges a discussion about block downdating with Adam Bojanczyk at the initial stage of this work. We would also like to thank the referees for their helpful comments.

REFERENCES

[1] S. T. ALEXANDER, C.-T. PAN, AND R. J. PLEMMONS, *Analysis of a recursive least squares hyperbolic rotation algorithm for signal processing*, Linear Algebra Appl., 98 (1988), pp. 3–40.

[2] C. H. BISCHOF, *Incremental condition estimation*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 312–332.

[3] Å. BJÖRCK, *Stability analysis of the method of semi-normal equations for least squares problems*, Linear Algebra Appl., 88/89 (1987), pp. 31–48.

[4] Å. BJÖRCK, H. PARK, AND L. ELDÉN, *Accurate downdating of least squares solutions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 550–570.

[5] A. BOJANCZYK AND A. O. STEINHARDT, *Stabilized hyperbolic Householder transformations*, IEEE Trans. Acoust., Speech Signal Process., ASSP-37 (1989), pp. 1286–1288.

[6] A. W. BOJANCZYK, J. G. NAGY, AND R. J. PLEMMONS, *Block* RLS *using row Householder transformations*, Linear Algebra Appl., 188 (1993), pp. 31–61.

[7] J. DANIEL, W. B. GRAGG, L. KAUFMAN, AND G. W. STEWART, *Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[8] J. J. DONGARRA, J. DUCROZ, I. DUFF, AND S. HAMMARLING, *A set of level* 3 *basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[9] L. ELDÉN AND H. PARK, *Perturbation Analysis for Block Downdating of a Cholesky Decomposition*, Numer. Math., to appear.

[10] L. ELDÉN, *Downdating QR decompositions*, in Proc. Conf. Mathematics in Signal Processing, 1988, The Institute of Mathematics and Its Application, Essex, U.K., 1990.

[11] W. R. FERNG, G. H. GOLUB, AND R. J. PLEMMONS, *Adaptive Lanczos methods for recursive condition estimation*, J. Numer. Algebra, 1 (1991), pp. 1–20.

[12] G. H. GOLUB AND G. P. STYAN, *Numerical computations for univariate linear models*, J. Statist. Comput. Simulations, 2 (1973), pp. 253–274.

[13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[14] C. T. PAN, *A perturbation analysis of the problem of downdating a Cholesky factorization*, Linear Algebra Appl., 183(19xx), pp. 103–115.

[15] C. T. PAN AND R. J. PLEMMONS, *Least squares modifications with inverse factorizations: Parallel implications*, Comput. Appl. Math., 27 (1989), pp. 109–127.

[16] C. RADER AND A. STEINHARDT, *Hyperbolic Householder transformations*, IEEE Trans. Acoust. Speech Signal Process., ASSP-34 (1986), pp. 1589–1602.

[17] M. A. SAUNDERS, *Large-Scale Linear Programming Using The Cholesky Factorization*, Tech. report CS252, Computer Science Department, Stanford University, Stanford CA, 1972.

[18] R. SCHREIBER AND W.-P. TANG, *On systolic arrays for updating the Cholesky factorization*, BIT, 26 (1986), pp. 451–466.

[19] G. W. STEWART, *The effects of rounding error on an algorithm for downdating a Cholesky factorization*, J. Inst. Math. Appl., 23 (1979), pp. 203–213.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 1035–1043, July 1994

023

AN ATTAINABLE LOWER BOUND FOR THE BEST NORMAL APPROXIMATION*

LAJOS LÁSZLÓ†

Abstract. Lower bounds for the distance of a complex $n \times n$ matrix A from the variety of normal matrices are established. The weaker version gives a lower bound of the form $\mathrm{dep}(A)/\sqrt{n}$, where $\mathrm{dep}(A)$ is Henrici's "departure from normality." Recall that $\mathrm{dep}(A)$ itself is an upper bound for the distance at issue. The tighter bound contains n diagonal sums coming from the Schur form, hence its computational cost is larger; however, it is attainable. The main result is showing this property. To this end some lemmas concerning normal and triangular matrices are needed, and a set of triangular and (closest) normal matrices with properties of independent interest is introduced.

Key words. best normal approximation, departure from normality, lower bound, extremal normal matrices, Schur decomposition

AMS subject classifications. 15A60, 15A42

1. Introduction.

Let A be a complex matrix of order n, and denote by $\nu(A)$ its distance from the set of normal matrices, i.e., let

$$\nu(A) = \nu_F(A) = \min \{\|A - N\|_F : \ N \text{ is normal}\}.$$

There exist a number of measures of nonnormality with several inequalities holding between them, e.g.,

(i) the "departure from normality" defined by Henrici [4, p. 27]:

$$\mathrm{dep}(A) = \{\|A\|_F^2 - \sum |\lambda_i|^2\}^{1/2},$$

where $\{\lambda_i\}$ are the eigenvalues of A; or

(ii) the commutator of A and A^*, more precisely,

$$\mathrm{com}(A) = \|A^*A - AA^*\|_F^{1/2}.$$

The latter one is suitable for estimating $\nu(A)$ both from below and above; see (C1), (C3), (C5) in Elsner–Paardekooper [2, p. 111]:

$$\mathrm{com}^2(A) \ / \ (4\|A\|_2) \ \leq \ \nu(A) \ \leq ((n^3 - n)/12)^{1/4}\mathrm{com}(A).$$

However, this is unsatisfactory in some sense as stated by Higham [5, p. 16], "Unfortunately the lower and upper bounds differ by orders of magnitude when $\nu(A)/\|A\|_F$ is small."

In light of this fact, it is remarkable that there exist lower and upper bounds for $\nu(A)$ in terms of $\mathrm{dep}(A)$—without the trouble mentioned:

$$\mathrm{dep}(A)/\sqrt{n} \ \leq \ \nu(A) \ \leq \mathrm{dep}(A). \tag{1}$$

*Received by the editors June 15, 1992; accepted for publication (in revised form) April 21, 1993.

†Department of Numerical Analysis, Eötvös Loránd University, Múzeum krt. 6-8, 1088 Budapest, Hungary (llaszlo@ludens.elte.hu).

The second inequality here is an easy consequence of the Schur decomposition theorem (see, e.g., Elsner–Paardekooper [2, p. 115], or Ruhe [8, p. 587]), while the first one is new, see Theorem 2.

In fact we will obtain a tighter lower bound, which is attainable, i.e., the coefficients involved are optimal.

THEOREM 1. *Let A be $n \times n$ with a Schur decomposition*

$$A = WRW^*$$

and let N be its closest normal matrix in the Frobenius norm. Then

$$\|R. * T\|_F \leq \|A - N\|_F \leq \|R. * E\|_F$$

with . denoting elementwise (Hadamard) product and T and E are upper triangular Toeplitz matrices with first row*

$$t_1 = (0, 1/n, 2/(n+1), \ldots, 1/2)^{1/2} \quad \text{and} \quad (0, 1, 1, \ldots, 1),$$

respectively. No single element of t_1 can be replaced by a larger number.

Remark. Observe that the theorem is formulated in terms of a *fixed* Schur decomposition; hence the lower bound $\|R. * T\|_F$ does not depend on A only, but also on R, the Schur form used. (As an illustrative 3×3 example for the nonunicity of the Schur decomposition, see [6].) However, the following weakened version gives an underestimate, independent on R.

THEOREM 2. *Let A be $n \times n$ with eigenvalues $\{\lambda_i\}$. Then* (1) *holds.*

Remark. Recall that relations between $\mathrm{dep}(A)$ and $\mathrm{com}(A)$ also are available; see (C1),(C2) in [2, p. 111]. We prove Theorems 1 and 2 in §4.

2. Notations. We mention in advance that the subscript F at the matrix norms is omitted since the Frobenius norm is exclusively used. The order of all square matrices occurring through this paper is denoted by n, while $n \geq 2$, respectively, $n \geq 3$ in certain cases. The classes of right triangular, unitary, and normal matrices are denoted by $\mathcal{R}$, $\mathcal{U}$, $\mathcal{N}$, omitting the index n for the sake of simplicity. Matrix elements are referred to with the corresponding lower case and indices if not otherwise stated.

Any $R \in \mathcal{R}$ can be partitioned into the diagonals

$$r(m) = (r_{1,m+1}, r_{2,m+2}, \ldots, r_{n-m,n})^T, \qquad 0 \leq m \leq n-1.$$

In particular, the main diagonal—as the vector of the eigenvalues—is denoted by λ, i.e.,

$$\lambda = r(0).$$

We use four auxiliary matrices in conjunction with elementwise operations (including square rooting). They are defined in §§1 and 3, more precisely: E and T in Theorem 1; S in Lemma 1; M in Lemma 2.

3. Preliminary lemmas. The first two statements of this section are true for arbitrary normal matrices. (We use for them the notation Z instead of N to avoid writing "n_{ij}" for the entries.)

LEMMA 1. *If $Z \in \mathcal{N}$, then*

$$\|Z. * S\|^2 = \|Z. * S^T\|^2,$$

where S is the upper triangular Toeplitz matrix with first row

$$(0, 1, 2, \ldots, n-1)^{1/2}.$$

Proof. The equality of the main diagonal elements in $ZZ^* = Z^*Z$ yields

$$\sum_{j=1}^{n} |z_{ij}|^2 = \sum_{j=1}^{n} |z_{ji}|^2, \qquad 1 \le i \le n.$$

By summation for $1 \le i \le k$ and subtracting the equal terms, we have

$$\sum_{i=1}^{k} \sum_{j=k+1}^{n} |z_{ij}|^2 = \sum_{i=1}^{k} \sum_{j=k+1}^{n} |z_{ji}|^2, \qquad 1 \le k \le n-1.$$

Now we add these equalities for $1 \le k \le n-1$, then we rearrange both sides in accordance with the scheme

$$\sum_{k=1}^{n-1} \sum_{i=1}^{k} \sum_{j=k+1}^{n} p_{ij} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (j-i) p_{ij},$$

and we change the indices on the right to get

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (j-i)|z_{ij}|^2 = \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} (i-j)|z_{ij}|^2. \qquad \square$$

The normal, *very* nearly triangular matrices play a significant role in the treatment.

DEFINITION 1. $\mathcal{N}_0 = \{ Z \in \mathcal{N} : z_{ij} = 0 \text{ if } i > j \text{ and } (i,j) \ne (n,1) \}$.

For example,

$$Z = \begin{pmatrix} 1 & 4 & 3 \\ 0 & -2 & 4 \\ 5 & 0 & 1 \end{pmatrix} \in \mathcal{N}_0.$$

In Lemma 2 the class $\mathcal{N}_0$ is characterized. We denote elementwise division by $./$, so that $A./B = (a_{ij}/b_{ij})$.

LEMMA 2. $\|Z./M\| \le \|Z\|$ *with equality for* $Z \in \mathcal{N}_0$, *where* M *is the upper triangular Toeplitz matrix with first row*

$$(1, (n-1)/n, (n-1)/(n+1), \ldots, (n-1)/(2n-3), 1/2)^{1/2}.$$

Proof. We have

$$\begin{aligned} \|Z\|^2 - \|Z./M\|^2 &= \sum_{i=1}^{n} \sum_{j=1}^{n} |z_{ij}|^2 - \sum_{i=1}^{n} \sum_{j=i}^{n} (n+j-i-1)/(n-1)\ |z_{ij}|^2 \\ &= \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} |z_{ij}|^2 - \sum_{i=1}^{n} \sum_{j=i}^{n} (j-i)/(n-1)\ |z_{ij}|^2. \end{aligned}$$

Applying Lemma 1 to the second term, we get

$$\|Z\|^2 - \|Z./M\|^2 = \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} (n+j-i-1)/(n-1)\ |z_{ij}|^2.$$

Since here every term is nonnegative, the first statement is true. On the other hand, this double sum contains only terms with indices $i > j$, while the coefficient of $|z_{ij}|^2$ is positive except when $(i,j) = (n,1)$; hence the second statement holds as well. □

The following definition is based on [7, Thm. 4(i)], using the equivalence of the problems

$$\max\{\|\operatorname{diag} UAU^*\|^2 :\ U \in \mathcal{U}\} \tag{2}$$

and

$$\min\{\|A - N\|^2 :\ N \in \mathcal{N}\} \tag{3}$$

as investigated in [1] and [3].

DEFINITION 2. *For $R \in \mathcal{R}$, $Z \in \mathcal{N}$ let*

$$\Delta(R,Z) = \|R - Z\|^2 - \|R. * T\|^2.$$

Remark. Because of the theorem mentioned, this quantity is always nonnegative, since any upper bound "ub" for (2) automatically yields a lower bound "lb" for (3). This can be seen by observing that

$$\|A - U^*DU\|^2 = \|UAU^* - D\|^2 \geq \|UAU^*\|^2 - \|\operatorname{diag} UAU^*\|^2$$

$$\geq \|A\|^2 - \text{ub} = \|R\|^2 - \text{ub} = \text{lb}$$

holds for any diagonal D and unitary U, i.e., for any $N = U^*DU \in \mathcal{N}$; R is an arbitrary but fixed Schur form of A as in Theorem 1. Consequently, by taking the upper bound $\text{ub} = \|R. * M\|^2$ from [7, Thm. 4(i)], we immediately get $\text{lb} = \|R. * T\|^2$ and, at the same time, $\|A - \mathcal{N}\|^2 = \|R - \mathcal{N}\|^2 \geq \text{lb}$.

Remark. Lemma 2 and inequality $\|R. * M\| \leq \|R\|$ can be extended to a chain of inequalities

$$\|Z./M\|^2 \leq \|Z\|^2 \leq \|R. * M\|^2 \leq \|R\|^2,$$

where

(i) the first $\leq$ is true for every $Z \in \mathcal{N}$,

(ii) the third $\leq$ is true for every $R \in \mathcal{R}$,

(iii) the second $\leq$ holds if Z is the best normal approximation to R, since then

$$\|R - Z\|^2 = \|R\|^2 - \|Z\|^2, \quad \|R. * T\|^2 = \|R\|^2 - \|R. * M\|^2, \quad \text{and} \quad \Delta(R,Z) \geq 0.$$

DEFINITION 3. *$R \in \mathcal{R}$ is called* extremal *if there exists $Z \in \mathcal{N}$ such that*

$$\Delta(R,Z) = 0.$$

If, moreover, the restrictions

$$\|r(m)\|^2 \neq 0, \qquad 0 \leq m \leq n-1$$

also hold, then R is perfectly extremal.

Remark. If R is extremal, then Z at issue is *necessarily* its best normal approximation, i.e., the closest normal matrix to R.

Our last preparing lemma yields a useful representation for $\Delta(R, Z)$.

LEMMA 3. $\Delta(R, Z) = \|R. * M - Z./M\|^2 + \|Z\|^2 - \|Z./M\|^2$.

Proof. By Definition 2 we have

$$\begin{aligned}
\Delta(R,Z) &= \sum_{i=1}^{n}\sum_{j=i}^{n}|r_{ij}-z_{ij}|^2 + \sum_{j=1}^{n-1}\sum_{i=j+1}^{n}|z_{ij}|^2 \\
&\quad - \sum_{i=1}^{n}\sum_{j=i}^{n}(j-i)/(n+j-i-1)\;|r_{ij}|^2 \\
&= \sum_{i=1}^{n}\sum_{j=i}^{n}(n-1)/(n+j-i-1)\;|r_{ij}-(n+j-i-1)/(n-1)\;z_{ij}|^2 \\
&\quad + \sum_{j=1}^{n-1}\sum_{i=j+1}^{n}|z_{ij}|^2 - \sum_{i=1}^{n}\sum_{j=i}^{n}(j-i)/(n-1)\;|z_{ij}|^2 \\
&= \Sigma_1 + \Sigma_2 - \Sigma_3,
\end{aligned}$$

where we have applied elementary algebraic transformations. Let

$$\Sigma_0 = \sum_{i=1}^{n}\sum_{j=i}^{n}|z_{ij}|^2,$$

then

$$\begin{aligned}
\Delta(R,Z) &= \Sigma_1 + (\Sigma_2 + \Sigma_0) - (\Sigma_3 + \Sigma_0) \\
&= \|R. * M - Z./M\|^2 + \|Z\|^2 - \|Z./M\|^2,
\end{aligned}$$

and the lemma is proved. □

Remark. By virtue of Lemmas 2 and 3 extremal matrices can be characterized by using the equivalence

$$\Delta(R,Z) = 0 \iff R. * M = Z./M \quad \text{and} \quad Z \in \mathcal{N}_0.$$

4. Perfectly extremal matrices: proof of the main result. Before constructing perfectly extremal matrices of any order n, we distinguish between the cases $n = 2$ and $n \geq 3$.

(i) In case of $n = 2$ every $R \in \mathcal{R}$ is extremal. For, if

$$R = \begin{pmatrix} \lambda_1 & r \\ 0 & \lambda_2 \end{pmatrix} \in \mathcal{R},$$

then

$$R. * M = \begin{pmatrix} \lambda_1 & r/\sqrt{2} \\ 0 & \lambda_2 \end{pmatrix} \in \mathcal{R},$$

and

$$N = \begin{pmatrix} \lambda_1 & r/2 \\ \varepsilon r/2 & \lambda_2 \end{pmatrix} \in \mathcal{N}$$

for a suitable complex ε with $|\varepsilon| = 1$. (If $z = (\lambda_1 - \lambda_2)\bar{r} \neq 0$ then $\varepsilon = z/\bar{z}$; otherwise $\varepsilon = 1$ can be taken). Hence we have

$$\|R - N\|^2 = 1/2\,|r|^2 = \|R\|^2 - \|R. * M\|^2 = \|R. * T\|^2,$$

thus R is extremal.

(ii) In case of $n \geq 3$ one would have a simple way of proving Theorem 1, namely, constructing n extremal $n \times n$ matrices such that

$$\|r(m)\|^2 \neq 0 \quad \text{and} \quad \|r(i)\|^2 = 0, \quad 0 \leq i \leq n-1, \quad i \neq m$$

holds for the mth one. However, this idea is not executable, as the following example /$n = 4, m = 2$/ shows: among the matrices with pattern

$$\begin{pmatrix} 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

there is no extremal matrix! Thus, constructing perfectly extremal matrices cannot be avoided.

Proof of Theorem 1. The second inequality is well known: it states that

$$\|A - \mathcal{N}\| = \|R - \mathcal{N}\| \leq \|R - \operatorname{diag} R\|.$$

Since the first inequality is a reformulation of [7, Thm. 4(i), p. 297], it suffices to prove that it can be attained. To this aim we choose $U = I$, i.e., $A = R \in \mathcal{R}$, and assume $n \geq 3$. There will be defined perfectly extremal matrices of

$$k = [(n-1)/2]$$

free parameters, by means of the following "big steps."

I. We give a matrix $R \in \mathcal{R}$ of k free parameters.

II. We give a matrix $Z \in \mathcal{N}_0$ in terms of the above free parameters.

In detail the following instructions should be kept.

I. (R1) R is real;

(R2) R is persymmetric, i.e.,

$$r_{ij} = r_{n-j+1,n-i+1}, \qquad 1 \leq i, j \leq n;$$

(R3) The elements of R "within the upper triangle" are zero, i.e.,

$$r_{ij} = 0, \qquad 1 < i < j < n;$$

(R4) $r_{11} = r_{nn} = n - 2$; $r_{ii} = -2$, $2 \leq i \leq n-1$; $r_{1n} = 2n$;

(R5) $0 \neq r_{1i}$, $2 \leq i \leq k+1$ are free parameters;

(R6) $(n + i - 2)r_{1,n+1-i} = (2n - 1 - i)r_{1i}$, $2 \leq i \leq n-1$.

II. (Z1) Z is real;

(Z2) Z is persymmetric;

(Z3) $z_{ij} = 0$, $1 < i < j < n$, and $Z \in \mathcal{N}_0$;

(Z4) $z_{11} = z_{nn} = n - 2$; $z_{ii} = -2$, $2 \leq i \leq n-1$; $z_{1n} = n$;

(Z5) $z_{1i} = r_{1i}(n-1)/(n+i-2)$, $2 \leq i \leq k+1$;

(Z6) $z_{1,n+1-i} = z_{1i}$, $2 \le i \le n-1$; and $z_{n1} = (\sum_{i=2}^{n} z_{1i}^2)^{1/2}$.

It is easy to check that I and II uniquely determine $R \in \mathcal{R}$ and $Z \in \mathcal{N}_0$ in terms of the parameters (R5), while (Z5) implies that

$$R. * M = Z./M.$$

Hence, from the last remark of the previous section it follows that R is extremal, and owing to (R5), R can be assumed to be perfect. The theorem is proved. □

Proof of Theorem 2. The following chain of (in)equalities yields a lower bound for $\|R. * T\|^2$:

$$\begin{aligned}
\|R. * T\|^2 &= \sum_{m=0}^{n-1} m/(n+m-1)\ \|r(m)\|^2 \\
&= \|R\|^2 - \|\lambda\|^2 - \sum_{m=1}^{n-1} (n-1)/(n+m-1)\ \|r(m)\|^2 \\
&\ge \|R\|^2 - \|\lambda\|^2 - (n-1)/n \sum_{m=1}^{n-1} \|r(m)\|^2 \\
&= \|R\|^2 - \|\lambda\|^2 - (n-1)/n\ (\|R\|^2 - \|\lambda\|^2) \\
&= 1/n\ (\|R\|^2 - |\lambda\|^2) \\
&= 1/n\ (\|A\|^2 - |\lambda\|^2) \\
&= \mathrm{dep}^2(A)\,/\,n,
\end{aligned}$$

using $\lambda = r(0)$ and $\|A\|^2 = \|R\|^2$. Applying the first relation in Theorem 1 gives the first inequality of (1). The second inequality is equivalent to that of Theorem 1, by observing that $\mathrm{dep}(A) = \|R. * E\|$. The proof is complete. □

Remark. Observe that the first inequality of (1) corresponds to the "simple version" [7, Thm. 3(iv)]. Therefore it can be proved also by rewriting that theorem in terms of the best normal approximation, i.e., from (2) into (3), to get

$$\|A - N\|^2 \ge \|A\|^2 - \left(\frac{1}{n} \sum_{i=1}^{n} |\lambda_i|^2 + \frac{(n-1)}{n} \|A\|^2 \right) = \mathrm{dep}^2(A)/n.$$

Example. The following choice of the free parameters yields a quite special "canonical" pair $\{R, Z\}$:

$$r_{1i} = n + i - 2, \quad z_{1i} = n - 1, \quad 2 \le i \le n-1.$$

For illustration, we display the ninth order special extremal triangular and normal matrices R_9 and Z_9. (This case is distinguished by the fact that the left bottom element in Z_9 is integer, too.)

$$R_9 = \begin{pmatrix}
7 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 18 \\
0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 15 \\
0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 14 \\
0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 13 \\
0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 12 \\
0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 11 \\
0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7
\end{pmatrix},$$

$$Z_9 = \begin{pmatrix} 7 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 9 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 8 \\ 23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \end{pmatrix}.$$

Example. (The case $n = 3$). The matrix $R_3 = R_3(p)$ below is extremal for any real value of the free parameter p; moreover, it is perfect if and only if $p \neq 0$:

$$R_3(p) = \begin{pmatrix} 1 & p & 6 \\ 0 & -2 & p \\ 0 & 0 & 1 \end{pmatrix}.$$

The best normal approximation to $R_3(p)$ is

$$Z_3(p) = \begin{pmatrix} 1 & 2p/3 & 3 \\ 0 & -2 & 2p/3 \\ q & 0 & 1 \end{pmatrix},$$

where $q = (9 + 4p^2/9)^{1/2}$.

Observe that the "canonical" pair $\{R_3,\ Z_3\}$ defined in the above example can be obtained for $p = 3$.

Remark. The matrices constructed in the theorem are real. This makes one think that all matrices involved in the problem of *extremality* can be assumed to be real. However, this conception is fundamentally false: the *optimal* unitary U, corresponding to a perfectly extremal R, is in case of $n \geq 3$ *necessarily complex*!

To see this, we must consider parallel with

$$\nu(A) \ = \ \min\{\|A - N\| : \ N = U\Lambda U^*,\ \Lambda : \text{diag},\ U : \text{unitary}\},$$

also the quantity

$$\nu^{\text{real}}(A) \ = \min\{\|A - N\| : \ N = U\Lambda U^T,\ \Lambda : \text{diag},\ U : \text{real orthogonal}\}.$$

Then we have

$$\|R. * E\|/\sqrt{2} \leq \nu^{\text{real}}(A)$$

for any $A = R \in \mathcal{R}$. (This follows from [7, Remark 3] and [7, Lemma 1(e)], using the equivalence of the problems (2) and (3).) At the same time, if

$$\|r(m)\|^2 \neq 0$$

holds for at least one m, $1 \leq m \leq n - 1$, then obviously

$$\|R. * T\| < \|R. * E\|/\sqrt{2},$$

showing that the lower bound $\|R.*T\|$ cannot be attained for real orthogonal matrices.

We illustrate this phenomenon by the help of 3×3 matrices.

Example. Consider the matrices

$$R = \begin{pmatrix} 1 & 6 & 6 \\ 0 & -2 & 6 \\ 0 & 0 & 1 \end{pmatrix} \in \mathcal{R}, \qquad N = \begin{pmatrix} 1 & 4 & 3 \\ 0 & -2 & 4 \\ 5 & 0 & 1 \end{pmatrix} \in \mathcal{N}_0.$$

Both matrices are real. Furthermore, $\Delta(R, N) = 0$, consequently, R is extremal (in fact, perfectly extremal) and N is its best normal approximation. By the considerations above, the minimum in $\nu^{\text{real}}(R)$ is not attainable for real orthogonal matrices, or equivalently, the eigenvectors of N cannot be pure real. This can also be checked by calculating its characteristic polynomial

$$\det(\lambda I - N) = \lambda^3 - 18\lambda - 108 = (\lambda - 6)\{(\lambda + 3)^2 + 9\}.$$

REFERENCES

[1] R. L. CAUSEY, *On Closest Normal Matrices*, Tech. Report CS-10, Dept. of Computer Science, Stanford University, Stanford, CA, 1964.

[2] L. ELSNER AND M. H. C. PAARDEKOOPER, *On measures of nonnormality of matrices*, LAA, 92 (1987), pp. 107–124.

[3] R. GABRIEL, *Matrizen mit maximaler Diagonale bei unitärer Similarität*, J. Reine Angew. Math., 307/308 (1979), pp. 31–52.

[4] P. HENRICI, *Bounds for iterates, inverses, spectral variation and fields of values of non-normal matrices*, Numer. Math., 4 (1962), pp. 24–40.

[5] N. J. HIGHAM, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. J. C. Gover and S. Barnett, eds., Oxford University Press, Oxford, 1989, pp. 1–27.

[6] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[7] L. LÁSZLÓ, *Upper bounds for matrix diagonals*, Linear and Multilinear Algebra, 30 (1991), pp. 283–301.

[8] A. RUHE, *Closest normal matrix finally found!*, BIT, 27 (1987), pp. 585–598.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1045–1060, October 1994

A PERTURBATION ANALYSIS OF THE GENERALIZED SYLVESTER EQUATION $(AR - LB, DR - LE) = (C, F)^*$

BO KÅGSTRÖM†

Abstract. Perturbation and error bounds for the generalized Sylvester equation $(AR-LB, DR-LE) = (C, F)$ are presented. An explicit expression for the normwise relative backward error associated with an approximate solution of the generalized Sylvester equation is derived and conditions when it can be much greater than the relative residual are given. This analysis is applicable to any method that solves the generalized Sylvester equation. A condition number that reflects the structure of the problem and a normwise forward error bound based on $\mathrm{Dif}^{-1}[(A, D), (B, E)]$ and the residual are derived. The structure-preserving condition number can be arbitrarily smaller than a Dif^{-1}-based condition number. The normwise error bound can be evaluated robustly and at moderate cost by using a reliable Dif^{-1} estimator. A componentwise LAPACK-style forward error bound that can be stronger than the normwise error bound is also presented. A componentwise approximate error bound that can be evaluated to a much lower cost is also proposed. Finally, some computational experiments that validate and evaluate the perturbation and error bounds are presented.

Key words. generalized Sylvester equation, backward error, condition number, perturbation bounds, error bounds.

AMS subject classifications. primary, 65F05, 65G05.

1. Introduction.

In this paper we study the sensitivity of and derive perturbation and error bounds for the generalized Sylvester equation

$$\begin{aligned} AR - LB &= C, \\ DR - LE &= F, \end{aligned} \tag{1}$$

where L and R are unknown $m \times n$ matrices, (A, D), (B, E), and (C, F) are given pairs of $m \times m, n \times n$, and $m \times n$ matrices, respectively, with real (or complex) entries. If we choose D and E to be the identity matrices and F as the zero matrix then (1) reduces to the (standard) Sylvester equation $AR - RB = C$. Using Kronecker products the matrix equation (1) can be written as a $2mn \times 2mn$ linear system of equations [6]

$$\begin{bmatrix} I_n \otimes A & -B^T \otimes I_m \\ I_n \otimes D & -E^T \otimes I_m \end{bmatrix} \begin{bmatrix} \mathrm{col}(R) \\ \mathrm{col}(L) \end{bmatrix} = \begin{bmatrix} \mathrm{col}(C) \\ \mathrm{col}(F) \end{bmatrix}, \tag{2}$$

where the column vector $\mathrm{col}(X)$ denotes an ordered stack of the columns of a matrix X from left to right starting with the first column. We write the system (2) in compact form as

$$Zx = b. \tag{3}$$

The coefficient matrix Z in (2) is $2mn \times 2mn$, which for moderate m and n is already quite a large matrix. So this equivalent formulation is mainly of interest for theoretical purposes.

* Received by the editors March 26, 1993; accepted for publication (in revised form) July 30, 1993. The work of this author was supported by the Swedish Board of Industrial and Technical Development grant NUTEK 89-02578P.

† Department of Computing Science, Umeå University, S-901 87 Umeå, Sweden (bokg@cs.umu.se).

One important application of the generalized Sylvester equation originates from computing stable eigendecompositions of matrix pencils [6]. It can be formulated in terms of a block-diagonalizing equivalence transformation $P^{-1}(M - \lambda N)Q$, where

$$M - \lambda N \equiv \begin{bmatrix} A & -C \\ 0 & B \end{bmatrix} - \lambda \begin{bmatrix} D & -F \\ 0 & E \end{bmatrix}. \tag{4}$$

We want to find (L, R) such that

$$\begin{bmatrix} I_m & -L \\ 0 & I_n \end{bmatrix} (M - \lambda N) \begin{bmatrix} I_m & R \\ 0 & I_n \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} - \lambda \begin{bmatrix} D & 0 \\ 0 & E \end{bmatrix}. \tag{5}$$

The first m columns of the transformation matrices in (5) (P^{-1} and Q, respectively) span a pair of eigenspaces (deflating subspaces) associated with $\lambda(A, D)$ [21]. By solving for (L, R) in (1) we also get a pair of complementary eigenspaces (deflating subspaces associated with $\lambda(B, E)$) from the last n columns of P^{-1} and Q, respectively. One can show that (1) has a unique solution if and only if the *regular* pencils $A - \lambda D$ and $B - \lambda E$ have disjoint spectra [20]. If these pencils have common spectra or are *singular* (i.e., $\det(A - \lambda D) \equiv 0$ or $\det(B - \lambda E) \equiv 0$ for each λ), the generalized Sylvester equation will not generally be consistent. An important quantity that measures the sensitivity of these eigenspaces is the *separation of the matrix pairs* (A, D) and (B, E) [20], [21],

$$\mathrm{Dif}[(A, D), (B, E)] = \inf_{\|(L,R)\|_F = 1} \|(AR - LB, DR - LE)\|_F. \tag{6}$$

The relationship with the generalized Sylvester equation is that $\mathrm{Dif}[(A, D), (B, E)] > 0$ (Dif for short) if and only if (1) has a unique solution. Furthermore, it can be shown [6] that

$$\mathrm{Dif}^{-1}[(A, D), (B, E)] = \|Z^{-1}\|_2 = \sigma_{\min}(Z)^{-1}, \tag{7}$$

where $\sigma_{\min}(Z)$ is the smallest singular value of Z.

A new direct method for reordering eigenvalues in the generalized real Schur form of a regular matrix pair is based on solving sequences of generalized Sylvester equations where m, n are 1 or 2 [14]. In [16] and [17], algorithms for computing an additive decomposition of a (generalized) transfer matrix are presented. The problem, computing the stable projection with respect to a specified region Γ in the complex plane of a transfer matrix given by its generalized state space realization, comprises both a reordering of eigenvalues and a block-diagonalization as described above.

Recently, Higham [13] presented a perturbation analysis of the standard Sylvester equation $AR - RB = C$. By taking full account of the structure of the Sylvester equation, he derives expressions for the backward error of an approximate solution $\hat{R}$ and a condition number that measures the sensitivity of a solution to perturbations in the data (A, B, C). One important result from [13] is that a small value of the residual $A\hat{R} - \hat{R}B - C$ does not necessarily yield a small backward error. The main purpose of this paper is to generalize these results and extend the analysis to the generalized Sylvester equation.

An alternative form of a generalized Sylvester equation with applications in control theory is

$$AXB^T + CXD^T = E, \tag{8}$$

where A and C are $m \times m$, B and D are $n \times n$, and E and the desired solution X are $m \times n$ [7]. The matrix equation (8) has a unique solution if and only if (A, C) and $(-D, B)$ are regular matrix pairs with disjoint spectra [5]. By introducing $R = XB^T$ and $L = CX$, (8) can be recast in the form (1)

$$\begin{aligned} AR + LD^T &= E, \\ CR - LB^T &= 0. \end{aligned}$$

The solvability condition for (8) (which is similar to the solvability condition for (1)) implies that at least one of B and C must be nonsingular, so X can be resolved from L or R. However, if both B and C are ill conditioned with respect to inversion (or one of them is singular and the other is ill conditioned), it is recommended to solve (8) directly [7].

The rest of the paper is outlined as follows. In §1.1 we collect the notation used. In §2 we briefly review algorithms for solving the generalized Sylvester equation and discuss residual bounds for an approximate solution $(\hat{L}, \hat{R})$. Section 3 is devoted to a normwise backward error analysis. An explicit expression for the normwise relative backward error associated with an approximate solution of (1) is derived and conditions when it can be much greater than the relative residual are given. This analysis is of course applicable to any method that solves the generalized Sylvester equation. In §4 we derive a condition number for the generalized Sylvester equation that reflects the structure of the problem and a normwise forward error bound based on $\mathrm{Dif}^{-1}[(A, D), (B, E)]$ and the residual. The structure-preserving condition number can be arbitrarily smaller than a Dif^{-1}-based condition number. The normwise error bound can be evaluated robustly and at moderate cost by using the Dif^{-1} estimators in [18]. Section 5 presents a componentwise forward error bound that can be stronger than the normwise error bound. This forward error bound can be converted to a LAPACK-style [1] error bound as for the standard Sylvester equation [13]. We also propose a componentwise approximate error bound that can be evaluated to a much lower cost. Finally, in §6 we present and discuss some computational experiments that validate and evaluate the perturbation and error bounds.

1.1. Notation. The following notation is used in the paper. $\lambda(A, B)$ denotes the spectrum of a regular matrix pair (A, B) or pencil $A - \lambda B$. If $B = I$ we only use $\lambda(A)$. $\|A\|_2$ denotes the spectral norm (2-norm) of a matrix A induced by the Euclidean vector norm. $\|A\|_F$ denotes the Frobenius (or Euclidean) matrix norm. $\|A\|_M = \max_{i,j} |a_{ij}|$, i.e., the maximum of the absolute values of the matrix entries. $\kappa(A) = \|A\|_2 \|A^+\|_2$, where A^+ is the pseudoinverse of A, and denotes the condition numbers of a matrix A with respect to the 2-norm. $\sigma(A)$ denotes the set of singular values of a matrix A. Especially, $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ denote the largest and smallest singular values of A, respectively. For a square matrix A, we have that $\|A\|_2 = \sigma_{\max}(A)$ and $\|A^{-1}\|_2 = \sigma_{\min}(A)^{-1}$. $A \otimes B$ denotes the Kronecker product of two matrices A and B whose (i, j)th block element is $a_{ij}B$. A^T denotes the transpose of A. A^H denotes the conjugate transpose of A. $\overline{A}$ denotes the conjugate of A. $|A|$ and $|x|$ denote the matrix and the vector whose elements are $|a_{ij}|$ and $|x_i|$, respectively. Inequalities such as $|A| \leq |B|, |x| \leq |y|$ are interpreted componentwise. $D = \mathrm{diag}(x)$ denotes a diagonal matrix wih $d_{ii} = x_i$.

2. Algorithms and residual bounds. In [18], algorithms for solving (1) are presented that are generalizations of the Schur method [3] and the Hessenberg–Schur method [8] for solving $AR - RB = C$. Both methods are based on orthogonal equiva-

lence transformations (unitary transformations if the matrix entries are complex) and involve the following four steps.

1. Transform (A, D) and (B, E) to simpler form:

$$(A_1, D_1) := (P^T A Q, P^T D Q),$$
$$(B_1, E_1) := (U^T B V, U^T E V).$$

2. Modify the right-hand sides (C, F):

$$C_1 := P^T C V, \qquad F_1 := P^T F V.$$

3. Solve the transformed system for (L_1, R_1):

$$\begin{aligned} A_1 R_1 - L_1 B_1 &= C_1, \\ D_1 R_1 - L_1 E_1 &= F_1. \end{aligned} \tag{9}$$

4. Transform the solution back to the original system:

$$L := P L_1 U^T, \qquad R := Q R_1 V^T.$$

In the generalized Schur method [18] (A_1, D_1) and (B_1, E_1) are in generalized real Schur form with A_1 and B_1 (upper) quasi triangular and D_1 and E_1 (upper) triangular. A quasi triangular matrix is block triangular with 1×1 and 2×2 diagonal blocks. The 2×2 blocks correspond to pairs of complex conjugate eigenvalues of the associated matrix pencil and the ratios of the 1×1 diagonal blocks are the real eigenvalues. (In the generalized complex Schur form A_1 and B_1 will be (upper) triangular too, which simplifies the discussion below.) Suppose the transformed matrix equation (9) is partitioned according to the diagonal block structure of A_1 and B_1. Let A_{ii} of size $a \times a, a = 1, 2$ and B_{jj} of size $b \times b, b = 1, 2$ denote the diagonal blocks of A_1 and B_1, respectively, and let p, q be the number of diagonal blocks of A_1 and B_1. Then (9) is solved by the GS algorithm that can be written compactly as [18]

$$\begin{aligned} A_{ii} R_{ij} - L_{ij} B_{jj} &= C_{ij} - \left(\textstyle\sum_{k=i+1}^{p} A_{ik} R_{kj} - \sum_{k=1}^{j-1} L_{ik} B_{kj} \right) \equiv G_{ij}, \\ D_{ii} R_{ij} - L_{ij} E_{jj} &= F_{ij} - \left(\textstyle\sum_{k=i+1}^{p} D_{ik} R_{kj} - \sum_{k=1}^{j-1} L_{ik} E_{kj} \right) \equiv H_{ij}, \end{aligned} \tag{10}$$

for $j = 1, 2, \ldots, q$ and $i = p, p-1, \ldots, 1$. In total we solve $p \cdot q$ small subsystems (10). Each of them can be written as a linear system

$$\begin{bmatrix} I_b \otimes A_{ii} & -B_{jj}^T \otimes I_a \\ I_b \otimes D_{ii} & -E_{jj}^T \otimes I_a \end{bmatrix} \begin{bmatrix} \operatorname{col}(R_{ij}) \\ \operatorname{col}(L_{ij}) \end{bmatrix} = \begin{bmatrix} \operatorname{col}(G_{ij}) \\ \operatorname{col}(H_{ij}) \end{bmatrix} \tag{11}$$

of size 2, 4, or 8. Note that even if A_{ii} and B_{jj} are upper triangular, the subsystems (11) cannot generally be transformed to upper triangular form only by permutations. This is in contrast to the Schur method for $AR - RB = C$.

In the generalized Hessenberg–Schur method [18], (A, D) is only transformed to a generalized Hessenberg-triangular form, where A_1 is (upper) Hessenberg and D_1 is (upper) triangular. The solution (L_1, R_1) is computed by solving a sequence of banded linear subsystems.

In both methods we use Gaussian elimination with partial pivoting to solve the generalized Sylvester equation (subsystem (11)) in each step. A rounding error analysis of the generalized Schur method is presented in [18]. The conclusion is that the

method is weakly stable [4], meaning that the relative errors in the computed solution $(\hat{L}, \hat{R})$ are small for well-conditioned problems. More precisely, the relative errors in $(\hat{L}, \hat{R})$ are proportional to a condition number times a smooth function of the relative machine precision. The analysis in [18] applies standard results on backward error analysis for products of orthogonal matrices (steps 1, 2, and 4) and for Gaussian elimination with partial pivoting for solving (9) in step 3 as $Z_1 x_1 = b_1$ (3). Since steps 1, 2, and 4 are backward stable processes, in general, the computed solution $\hat{x}$ of (3) satisfies (e.g., see [9])

$$(Z + \Delta Z)\hat{x} = b, \quad |\Delta Z| \leq 2mnu(3|Z| + 5P^T|\hat{L}_Z|\,|\hat{U}_Z|), \tag{12}$$

where $\hat{L}_Z, \hat{U}_Z$ are the computed LU factors, P the permutation matrix, and u is the unit roundoff (Z is $2mn \times 2mn$). From (12) we can derive the following normwise bounds on the residual of (1) for $\hat{x}$:

$$\begin{aligned} \|C - A\hat{R} + \hat{L}B\|_F &\leq c_{m,n}\rho_{m,n}u\left(\|A\|_F\|\hat{R}\|_F + \|\hat{L}\|_F\|B\|_F\right) \\ &\leq c_{m,n}\rho_{m,n}u\,(\|A\|_F + \|B\|_F)\max(\|\hat{L}\|_F, \|\hat{R}\|_F) \end{aligned} \tag{13}$$

and

$$\begin{aligned} \|F - D\hat{R} + \hat{L}E\|_F &\leq c_{m,n}\rho_{m,n}u\left(\|D\|_F\|\hat{R}\|_F + \|\hat{L}\|_F\|E\|_F\right) \\ &\leq c_{m,n}\rho_{m,n}u\,(\|D\|_F + \|E\|_F)\max(\|\hat{L}\|_F, \|\hat{R}\|_F), \end{aligned} \tag{14}$$

where $c_{m,n}$ is a modest function in the dimensions m and n, $\rho_{m,n}$ is the growth factor in Gaussian elimination. One interpretation of the bounds (13), (14) is that the relative residuals are bounded by a modest multiple of $c_{m,n}\rho_{m,n}u$. The size of the relative residuals is mainly determined by the (maximum) growth factor that measures how large the numbers become during the elimination processes. In practice, $\rho_{m,n}$ is usually of order 10 but it can also be as large as 2^{k-1}, where k is the size of the linear system solved. A similar error analysis can be performed for the generalized Hessenberg–Schur method that also results in residual bounds similar to (13), (14).

A LAPACK-style [1] block algorithm for solving (9) is under development [15]. Gaussian elimination with complete pivoting and tests for underflow and overflow are used to solve the subsystems (11). This will hopefully prohibit a large growth factor (in the residual bounds) despite the fact that examples have been found with growth factors greater than the problem size [10].

3. A normwise backward error analysis. Let $(\hat{L}, \hat{R})$ denote an approximate solution of the generalized Sylvester equation (1). The normwise backward error of $(\hat{L}, \hat{R})$ is defined by

$$\begin{aligned} \eta(\hat{L}, \hat{R}) \equiv \min\{\,\epsilon : &(A + \Delta A)\hat{R} - \hat{L}(B + \Delta B) = C + \Delta C, \\ &(D + \Delta D)\hat{R} - \hat{L}(E + \Delta E) = F + \Delta F, \\ &\|(\Delta A, \Delta D)\|_F \leq \epsilon\alpha, \|(\Delta B, \Delta E)\|_F \leq \epsilon\beta, \|(\Delta C, \Delta F)\|_F \leq \epsilon\gamma\}. \end{aligned} \tag{15}$$

It holds that $\eta(\hat{L}, \hat{R})$ is a measure of the distance to the closest perturbed generalized Sylvester equation that has $(\hat{L}, \hat{R})$ as the exact solution. By choosing $\alpha = \|(A, D)\|_F, \beta = \|(B, E)\|_F, \gamma = \|(C, F)\|_F$, $\eta(\hat{L}, \hat{R})$ corresponds to the normwise relative backward error with respect to the Frobenius norm. The perturbed generalized Sylvester equation in the definition (15) can be written as

$$\begin{aligned} \Delta A\hat{R} - \hat{L}\Delta B - \Delta C &= C - (A\hat{R} - \hat{L}B) \equiv R_1, \\ \Delta D\hat{R} - \hat{L}\Delta E - \Delta F &= F - (D\hat{R} - \hat{L}E) \equiv R_2, \end{aligned} \tag{16}$$

where (R_1, R_2) denotes the residual corresponding to $(\hat{L}, \hat{R})$. From (15) we can bound R_i as

$$\|R_i\|_F \leq (\alpha\|\hat{R}\|_F + \beta\|\hat{L}\|_F + \gamma)\eta(\hat{L}, \hat{R}). \tag{17}$$

Since $\|(R_1, R_2)\|_F = (\|R_1\|_F^2 + \|R_2\|_F^2)^{\frac{1}{2}}$, it follows that a small backward error of the generalized Sylvester equation implies a small relative residual. In the following analysis we show that a small relative residual will not always give a small (relative) backward error $\eta(\hat{L}, \hat{R})$.

We use a similar technique as in [13] and start to rewrite (16) by using Kronecker products:

$$\begin{aligned}(\hat{R}^T \otimes I_m)\mathrm{col}(\Delta A) - (I_n \otimes \hat{L})\mathrm{col}(\Delta B) - \mathrm{col}(\Delta C) &= \mathrm{col}(R_1),\\ (\hat{R}^T \otimes I_m)\mathrm{col}(\Delta D) - (I_n \otimes \hat{L})\mathrm{col}(\Delta E) - \mathrm{col}(\Delta F) &= \mathrm{col}(R_2),\end{aligned}$$

or equivalently

$$\mathcal{H}z = r, \qquad \mathcal{H} \equiv \begin{bmatrix} \hat{H} & 0 \\ 0 & \hat{H} \end{bmatrix}, \tag{18}$$

where

$$\hat{H} = \begin{bmatrix} \alpha(\hat{R}^T \otimes I_m) & -\beta(I_n \otimes \hat{L}) & -\gamma I_{mn} \end{bmatrix}, \tag{19}$$

and

$$z = \begin{bmatrix} \mathrm{col}(\Delta A)/\alpha \\ \mathrm{col}(\Delta B)/\beta \\ \mathrm{col}(\Delta C)/\gamma \\ \mathrm{col}(\Delta D)/\alpha \\ \mathrm{col}(\Delta E)/\beta \\ \mathrm{col}(\Delta F)/\gamma \end{bmatrix}, \qquad r = \begin{bmatrix} \mathrm{col}(R_1) \\ \mathrm{col}(R_2) \end{bmatrix}.$$

The system (18) is underdetermined where $\mathcal{H}$ is of size $2mn \times (2m^2 + 2n^2 + 2mn)$. If $\gamma \neq 0$, $\hat{H}$ is of full (row) rank and (18) has a minimum 2-norm solution

$$z = \mathcal{H}^+ r. \tag{20}$$

It follows from (18), (20) and the definition of $\eta(\hat{L}, \hat{R})$ that

$$\eta(\hat{L}, \hat{R}) \leq \|\mathcal{H}^+ r\|_2. \tag{21}$$

On the other side,

$$\|z\|_2^2 = \frac{\|(\Delta A, \Delta D)\|_F^2}{\alpha^2} + \frac{\|(\Delta B, \Delta E)\|_F^2}{\beta^2} + \frac{\|(\Delta C, \Delta F)\|_F^2}{\gamma^2} \leq 3\epsilon^2.$$

In summary, we have

$$\frac{1}{\sqrt{3}}\|\mathcal{H}^+ r\|_2 \leq \eta(\hat{L}, \hat{R}) \leq \|\mathcal{H}^+ r\|_2 \leq \|\mathcal{H}^+\|_2 \|r\|_2, \tag{22}$$

or in words, the maximum size of the backward error relative to the residual $\|(R_1, R_2)\|_F$ is dependent on $\|\mathcal{H}^+\|_2 = \sigma_{\min}(\hat{H})^{-1}$, where $\hat{H}$ is defined by (19).

Now, let $\hat{R} = U_R \Sigma_R V_R^H$ and $\hat{L} = U_L \Sigma_L V_L^H$ be the singular value decompositions of $\hat{R}$ and $\hat{L}$, respectively, giving

$$\hat{H} = \begin{bmatrix} \alpha(\overline{V}_R \Sigma_R^T U_R^T \otimes I_m) & -\beta(I_n \otimes U_L \Sigma_L V_L^H) & -\gamma I_{mn} \end{bmatrix}. \tag{23}$$

As in [13] we can find unitary transformations

$$Q = V_R^T \otimes U_L^H, \qquad P = \operatorname{diag}\{\overline{U}_R \otimes U_L, \overline{V}_R \otimes V_L, \overline{V}_R \otimes U_L\}$$

such that

$$\tilde{H} = Q\hat{H}P = \begin{bmatrix} \alpha(\Sigma_R^T \otimes I_m) & -\beta(I_n \otimes \Sigma_L) & -\gamma I_{mn} \end{bmatrix}. \tag{24}$$

Since $\hat{H}$ and $\tilde{H}$ are unitarily equivalent they have the same singular values, namely, the square roots of the eigenvalues of the diagonal matrix:

$$\tilde{H}\tilde{H}^H = \alpha^2(\Sigma_R^T \Sigma_R \otimes I_m) + \beta^2(I_n \otimes \Sigma_L \Sigma_L^T) + \gamma^2 I_{mn}. \tag{25}$$

Let $\sigma_i(\hat{R})$ and $\sigma_i(\hat{L})$ for $i = 1, \ldots, \min(m,n)$ denote the singular values of $\hat{R}$ and $\hat{L}$ in decreasing order ($\sigma_i() \geq \sigma_{i+1}()$), respectively, and define $\sigma_i(\hat{R}) = \sigma_i(\hat{L}) = 0$ for $i = \min(m,n) + 1, \ldots, \max(m,n)$. Then, assuming $\hat{H}$ has full (row) rank, we have

$$\|\mathcal{H}^+\|_2 = \|\hat{H}^+\|_2 = (\alpha^2 \sigma_n(\hat{R})^2 + \beta^2 \sigma_m(\hat{L})^2 + \gamma^2)^{-\frac{1}{2}}. \tag{26}$$

Substituting (26) in (22) gives us

$$\eta(\hat{L}, \hat{R}) \leq \mu(\hat{L}, \hat{R}) \frac{\|(R_1, R_2)\|_F}{(\alpha + \beta)\|(\hat{L}, \hat{R})\|_F + \gamma}, \tag{27}$$

where

$$\mu(\hat{L}, \hat{R}) = \frac{(\alpha + \beta)\|(\hat{L}, \hat{R})\|_F + \gamma}{(\alpha^2 \sigma_n(\hat{R})^2 + \beta^2 \sigma_m(\hat{L})^2 + \gamma^2)^{\frac{1}{2}}}. \tag{28}$$

By squaring both sides of (28), we see that $\mu(\hat{L}, \hat{R}) \geq 1$ and as for the standard Sylvester equation [13], $\mu(\hat{L}, \hat{R})$ is a *growth factor* that measures by how much the backward error $\eta(\hat{L}, \hat{R})$, at most, can be greater than the relative residual as defined in (27). It is now interesting to examine the size of $\mu(\hat{L}, \hat{R})$ more closely.

If $D = I_m$, $E = I_n$, and $F = 0$, then (1) reduces to the standard Sylvester equation $AR - RB = C$, and by choosing $\alpha = \|A\|_F, \beta = \|B\|_F, \gamma = \|C\|_F$, $\mu(\hat{L}, \hat{R})$ is at most a factor $\sqrt{2}$ times the growth factor for the standard Sylvester equation [13].

The detailed discussion of the growth factor in [13] can be extended to our case. For clarity, we accomplish some of it here. If $m = n$ then

$$\mu(\hat{L}, \hat{R}) = \frac{(\|(A,D)\|_F + \|(B,E)\|_F)\|(\hat{L}, \hat{R})\|_F + \|(C,F)\|_F}{(\|(A,D)\|_F{}^2 \sigma_{\min}(\hat{R})^2 + \|(B,E)\|_F{}^2 \sigma_{\min}(\hat{L})^2 + \|(C,F)\|_F{}^2)^{\frac{1}{2}}}. \tag{29}$$

The growth factor is large only when

$$\|(\hat{L}, \hat{R})\|_F \gg \sigma_{\min}(\hat{R}), \sigma_{\min}(\hat{L}) \tag{30}$$

and

$$\|(\hat{L}, \hat{R})\|_F \gg \frac{\|(C,F)\|_F}{\|(A,D)\|_F + \|(B,E)\|_F}, \tag{31}$$

i.e., $(\hat{L}, \hat{R})$ is an ill-conditioned, large-normed solution to the generalized Sylvester equation.

If $m \neq n$, one of $\sigma_n(\hat{R})$ and $\sigma_m(\hat{L})$ must be zero and we will have a large growth factor if the data is badly scaled: (A, D) (if $m < n$) or (B, E) (if $m > n$) greatly exceeds the rest of the data in norm. The effect on $\mu(\hat{L}, \hat{R})$ from badly scaled data can be overcome by regarding (A, D) and (B, E) as one set of data and choosing $\alpha = \beta = \|(A, D)\|_F + \|(B, E)\|_F$.

In practice, when solving the generalized Sylvester equation, we typically have (A, D) and (B, E) in generalized Schur form, i.e., A and B are quasi triangular and D and E are triangular. If we wish to restrict the perturbations to have the same structure, this can be done by removing elements in $\mathrm{col}(X), X = \Delta A, \Delta B, \Delta D, \Delta E$ in (18) that correspond to the "zero triangles" of A, B, D, E and deleting the corresponding columns of $\mathcal{H}$. However, removing columns of $\mathcal{H}$ will result in a possibly smaller $\sigma_{\min}$ and potentially a larger backward error.

4. Normwise perturbation and error bounds. Consider the perturbed generalized Sylvester equation

$$\begin{aligned} (A + \delta A)(R + \delta R) - (L + \delta L)(B + \delta B) &= C + \delta C, \\ (D + \delta D)(R + \delta R) - (L + \delta L)(E + \delta E) &= F + \delta F. \end{aligned} \tag{32}$$

By dropping second order terms in (32), we obtain

$$\begin{aligned} A\delta R - \delta L B &= \delta C - \delta A R + L \delta B, \\ D\delta R - \delta L E &= \delta F - \delta D R + L \delta E. \end{aligned} \tag{33}$$

The system (33) can be written

(34)

$$\begin{bmatrix} I_n \otimes A & -B^T \otimes I_m \\ I_n \otimes D & -E^T \otimes I_m \end{bmatrix} \begin{bmatrix} \mathrm{col}(\delta R) \\ \mathrm{col}(\delta L) \end{bmatrix} = - \begin{bmatrix} R^T \otimes I_m & -I_m \otimes L & -I_{mn} & 0 & 0 & 0 \\ 0 & 0 & 0 & R^T \otimes I_m & -I_m \otimes L & -I_{mn} \end{bmatrix} \begin{bmatrix} \mathrm{col}(\delta A) \\ \mathrm{col}(\delta B) \\ \mathrm{col}(\delta C) \\ \mathrm{col}(\delta D) \\ \mathrm{col}(\delta E) \\ \mathrm{col}(\delta F) \end{bmatrix}.$$

If we solve for $(\delta L, \delta R)$ in (34) and measure the perturbations normwise by

$$\epsilon = \max\left(\frac{\|(\delta A, \delta D)\|_F}{\alpha}, \frac{\|(\delta B, \delta E)\|_F}{\beta}, \frac{\|(\delta C, \delta F)\|_F}{\gamma} \right), \tag{35}$$

where $\alpha = \|(A, D)\|_F, \beta = \|(B, E)\|_F, \gamma = \|(C, F)\|_F$, we can derive the following relative perturbation bound.

THEOREM 4.1. *Assume that the unperturbed and perturbed matrix equations* (1) *and* (32), *respectively, are given, and that the perturbations in* (32) *fulfill* (35). *Then*

$$\frac{\|(\delta L, \delta R)\|_F}{\|(L, R)\|_F} \leq \frac{\|Z^{-1}\mathcal{X}\|_2}{\|(L, R)\|_F} \epsilon \sqrt{3}, \tag{36}$$

where Z is the coefficient matrix in (2), (34), *and*

$$\mathcal{X} = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}, \tag{37}$$

with

$$H = \begin{bmatrix} \alpha(R^T \otimes I_m) & -\beta(I_n \otimes L) & -\gamma I_{mn} \end{bmatrix}. \tag{38}$$

Proof. From (34) we obtain

$$\|(\delta L, \delta R)\|_F \leq \|Z^{-1}\mathcal{X}\|_2 \|\mathrm{diag}\{\alpha^{-1}, \beta^{-1}, \gamma^{-1}, \alpha^{-1}, \beta^{-1}, \gamma^{-1}\} \begin{bmatrix} \mathrm{col}(\delta A) \\ \mathrm{col}(\delta B) \\ \mathrm{col}(\delta C) \\ \mathrm{col}(\delta D) \\ \mathrm{col}(\delta E) \\ \mathrm{col}(\delta F) \end{bmatrix} \|_2.$$

The 2-norm of the second term above is equivalent to

$$(\alpha^{-2}\|(\delta A, \delta D)\|_F^2 + \beta^{-2}\|(\delta B, \delta E)\|_F^2 + \gamma^{-2}\|(\delta C, \delta F)\|_F^2)^{\frac{1}{2}},$$

which from (35) is less than or equal to $\epsilon\sqrt{3}$. □

The perturbation bound (36) is sharp to first order in ϵ and

$$\Psi = \frac{\|Z^{-1}\mathcal{X}\|_2}{\|(L, R)\|_F} \tag{39}$$

is the corresponding condition number for the generalized Sylvester equation. From (36) it is possible to derive a (weaker) normwise perturbation bound that relates to $\|Z^{-1}\|_2$.

COROLLARY 4.2. Assumptions from Theorem 4.1. *Then*

$$\frac{\|(\delta L, \delta R)\|_F}{\|(L, R)\|_F} \leq \Phi\epsilon\sqrt{3}, \tag{40}$$

where

$$\Phi = \|Z^{-1}\|_2 \frac{(\alpha + \beta)\|(L, R)\|_F + \gamma}{\|(L, R)\|_F}. \tag{41}$$

Proof. We have $\|Z^{-1}\mathcal{X}\|_2 \leq \|Z^{-1}\|_2\|\mathcal{X}\|_2$ and

$$\begin{aligned} \|\mathcal{X}\|_2 = \|H\|_2 &\leq \alpha\|R^T \otimes I_m\|_2 + \beta\|I_n \otimes L\|_2 + \gamma \\ &= \alpha\|R\|_2 + \beta\|L\|_2 + \gamma \leq (\alpha + \beta)\|(L, R)\|_F + \gamma, \end{aligned}$$

which in (36) gives (40). □

In [18] a similar bound was derived by applying standard perturbation theory for linear systems to (2). If $\|Z^{-1}\|_2(\alpha+\beta)\epsilon < r \leq 1$ then that theory (see, e.g., [9]) gives us the following strict bound for the relative error

$$\frac{\|(\delta L, \delta R)\|_F}{\|(L, R)\|_F} \leq \frac{2}{1 - r}\Phi\epsilon\sqrt{3}. \tag{42}$$

It is of course interesting to know how much the bounds (36) and (42) differ. From definition we know that $\Psi \leq \Phi$. In general the bounds will be of the same magnitude. But there are examples where (36) can be arbitrarily better than (42) (see §6). The reason is that the bound (42) does not take any account to the special structure of the problem. This is in contrast to the bound (36).

We can also show the following (a posteriori) forward error bound for a computed solution.

COROLLARY 4.3. *Let* $(\hat{L}, \hat{R})$ *be a computed solution to* (1) *with residuals*

$$R_1 = A\hat{R} - \hat{L}B - C,$$
$$R_2 = D\hat{R} - \hat{L}E - F.$$

Then

$$\frac{\|(L,R) - (\hat{L},\hat{R})\|_F}{\|(L,R)\|_F} \leq \|Z^{-1}\|_2 \frac{\|(R_1,R_2)\|_F}{\|(L,R)\|_F}. \tag{43}$$

Proof. Let $(\hat{L}, \hat{R}) = (L + \delta L, R + \delta R)$ and set $(\delta A, \delta D) = (0,0), (\delta D, \delta E) = (0,0)$ and $(\delta C, \delta F) = (R_1, R_2)$ in (34) and apply Theorem 4.1. □

The normwise error bound (43) holds in general but can be weaker than one based on componentwise errors that are described in the next section.

5. Componentwise error bounds. We will now derive a LAPACK-style error bound [1] for an approximate solution of the generalized Sylvester equation. In [13] it was shown that such a bound could be derived for the standard Sylvester equation and estimated by the technique used in the LAPACK library for linear systems. For clarity, we outline the technique used for the $Ax = b$ case. Let $\hat{x}$ be an approximate solution and $r = b - A\hat{x}$. Then the following error bound holds:

$$\frac{\|x - \hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq \frac{\|A^{-1}r\|_\infty}{\|\hat{x}\|_\infty} \leq \frac{\| \, |A^{-1}||r| \, \|_\infty}{\|\hat{x}\|_\infty}. \tag{44}$$

This bound can be interpreted as a componentwise bound since it measures the largest error in the components of the quantities involved. To obtain a strict error bound, we must add a term r_u for rounding errors in forming r (let $d = |r| + |r_u|$):

$$\frac{\|x - \hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq \frac{\| \, |A^{-1}|d\|_\infty}{\|\hat{x}\|_\infty}. \tag{45}$$

The trick is to write $D = \mathrm{diag}(d)$ and let $e = (1, \ldots, 1)^T$. Then the numerator of the error bound (44) can be written as [2]

$$\| \, |A^{-1}|De\|_\infty = \| \, |A^{-1}D|e\|_\infty = \|A^{-1}D\|_\infty.$$

Let $B = (A^{-1}D)^T$ and $\|B^T\|_\infty$ can now be estimated by the 1-norm estimator described in [11] and [12] (also implemented in LAPACK). It estimates $\|B\|_1$ at the cost of computing a few matrix-vector products involving B and B^T, that is, solving a few linear systems involving A^T and A, respectively.

Now we apply the same technique to the generalized Sylvester equation. The rounding errors in the computed residuals can be expressed as

$$\hat{R}_1 = \mathrm{fl}(C - (A\hat{R} - \hat{L}B)) = R_1 + \Delta R_1,$$
$$\hat{R}_2 = \mathrm{fl}(F - (D\hat{R} - \hat{L}E)) = R_2 + \Delta R_2,$$

where

$$|\Delta R_1| \le u\left(3|C| + (m+3)|A||\hat{R}| + (n+3)|\hat{L}||B|\right) \equiv {R_1}^u,$$
$$|\Delta R_2| \le u\left(3|F| + (m+3)|D||\hat{R}| + (n+3)|\hat{L}||E|\right) \equiv {R_2}^u.$$

Introducing

$$g = |\mathrm{col}(\begin{bmatrix} \hat{R}_1 & \hat{R}_2 \end{bmatrix})| + \mathrm{col}(\begin{bmatrix} {R_1}^u & {R_2}^u \end{bmatrix}),$$

and $G = \mathrm{diag}(g)$, we get the bound

$$\frac{\|(L,R) - (\hat{L},\hat{R})\|_M}{\|(\hat{L},\hat{R})\|_M} \le \frac{\| \, |Z^{-1}|g\|_M}{\|(\hat{L},\hat{R})\|_M} = \frac{\|Z^{-1}G\|_M}{\|(\hat{L},\hat{R})\|_M}. \tag{46}$$

From (46) we see that large elements in the kth column of $|Z^{-1}|$ can be offset by a small kth element of g. This situation can never be reflected in the bound (43) which in these cases is a weaker bound. Accordingly, (46) has better scaling properties than (43) but none of the bounds are invariant under diagonal scalings of the (generalized) Sylvester equation [13]. $Z^T y = c$ is not a transposed generalized Sylvester equation (1) (as in the standard case). However, y can be recovered at the same cost by utilizing that the matrix pairs (A, D) and (B, E) are already in generalized real Schur form [15]. The cost for solving (9) using the GS algorithm is $O(m^2n + mn^2)$ flops (see §2 and [18]).

We would like to compare the strict bound (46) with the following componentwise approximate error bound:

$$\frac{\|(L,R) - (\hat{L},\hat{R})\|_M}{\|(\hat{L},\hat{R})\|_M} \lesssim \frac{\|Z^{-1}r\|_M}{\|(\hat{L},\hat{R})\|_M}, \tag{47}$$

where

$$r = \mathrm{col}(\begin{bmatrix} \hat{R}_1 & \hat{R}_2 \end{bmatrix}) + \mathrm{col}(\begin{bmatrix} {R_1}^u & {R_2}^u \end{bmatrix}).$$

The bound (47) is easy to compute and corresponds to the defect in one step of iterative refinement of (3). To compute $Z^{-1}r$ is equivalent to solving a generalized Sylvester equation (1) with the right-hand sides $(C, F) = (\hat{R}_1 + {R_1}^u, \hat{R}_2 + {R_2}^u)$. The bound (47) has the same appealing properties as (46) and only requires one generalized Sylvester solve. Estimating (46) is an iterative process that typically requires four–six generalized Sylvester solves [15]. In our extensive testing we have only seen once that (47) underestimates the exact error. In this case the error bound was still the same order as the exact error.

6. Some computational experiments. In the following we illustrate the perturbation analysis of the generalized Sylvester equation on some test problems ranging from well-conditioned to extremely ill-conditioned problems.

Examples. The *first* set of test problems illustrates *well-conditioned to (moderately) ill-conditioned* generalized Sylvester equations [18]. In the first example we let $A = J_m(1,-1), D = I_m$ and $B = J_n(1-\alpha, 1), E = I_n$, where $J_k(d,s)$ denotes a Jordan block of size k with d and s as diagonal and superdiagonal elements, respectively, and $\alpha > 0$ is a real parameter. When $\alpha \to 0$, the separation between (A, D) and (B, E) is of order $O(\alpha^{m+n+1})$ [22]. Furthermore, we apply unitary random equivalence transformations to (A, D) and (B, E), respectively, and choose (L, R) randomly,

TABLE 1
Condition numbers for well-conditioned to moderately ill-conditioned problems.

m	n	α	$\|(\hat{L},\hat{R})\|_F$	$\sigma_{\min}(Z)$	Φ	Ψ	Φ/Ψ
4	4	1/2	3.21e+00	1.94e−04	3.76e+04	1.14e+04	3.30e+00
10	10	1/2	8.46e+00	1.95e−11	5.73e+11	1.19e+11	4.82e+00
6	6	1/16	4.85e+00	1.05e−16	9.03e+16	1.72e+16	5.27e+00
2	10		3.91e+00	1.93e−03	5.88e+03	2.16e+03	2.73e+00
8	4		4.43e+00	1.94e−04	5.84e+04	1.32e+04	4.43e+00
10	6		6.11e+00	1.27e−05	1.12e+06	2.34e+05	4.77e+00
10	10		8.02e+00	6.27e−07	2.65e+07	4.58e+06	5.78e+00
2	10		3.77e+00	3.56e−04	2.51e+04	9.48e+03	2.65e+00
8	4		4.61e+00	1.02e−05	8.85e+05	1.69e+05	5.24e+00
10	6		6.47e+00	1.02e−09	1.16e+10	2.70e+09	4.30e+00
10	10		8.46e+00	4.53e−08	3.23e+08	5.37e+07	6.01e+00

which similarly specifies the right-hand sides (C, F). The second example comprises upper triangular random problems. First, two upper triangular $(m+n) \times (m+n)$ matrices M and N are generated with entries chosen randomly (uniform distribution on $[-1, 1]$). Then A, B, C, D, E, F are given from the partitioning (4). Finally, the third example comprises quasi upper triangular random problems, i.e., M is chosen with 2×2 diagonal blocks and N upper triangular as before.

The *second* test problem illustrates *badly scaled* data. The examples chosen are upper triangular random problems (as above) and scaled in the following way: $A = \alpha A, D = \alpha D$ (if $m < n$), and $B = \alpha B, E = \alpha E$ (if $m > n$), where $\alpha = 10^{13}, 10^{16}$, and 10^{19}.

The *third* set of test problems illustrates *ill-conditioned* (generalized) Sylvester equations [13]. In the first example, $A = J_3(0,1), D = I_3, B = J_3(\alpha, 1), E = I_3$, and the entries of C, F are chosen to 1 and 0, respectively ($C_{ij} = 1, F_{ij} = 0$). In the second example, we have

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

First we choose $\mathrm{col}(C, F)$ as the right singular vector corresponding to the smallest singular value of Z (3). In the second case, $\mathrm{col}(C)$ is chosen as the left singular vector of the smallest singular value of $I_n \otimes A - B^T \otimes I_m$, the coefficient matrix of the corresponding $Zx = b$ representation of the standard Sylvester equation [13] and $F = 0$.

Test results and discussion. In Tables 1, 4, and 7 quantities that reflect the conditioning of our three sets of test problems are displayed. Besides $\|(\hat{L}, \hat{R})\|_F$ and $\mathrm{Dif}[(A, D), (B, E)] = \sigma_{\min}(Z)$, the Dif-based condition number Φ (41), the new structure preserving condition number Ψ (39) and their ratio Φ/Ψ are shown. In Tables 2, 5, and 8 associated computed residuals and backward error bounds are displayed. More precisely, $\|(R_1, R_2)\|_F$, the relative residual $\|(R_1, R_2)\|_F / ((\alpha + \beta)\|(\hat{L}, \hat{R})\|_F + \gamma)$ the exact backward error $\|\mathcal{H}^+ r\|_2$ (22), an upper bound on the backward error $\eta(\hat{L}, \hat{R})$ (27), and the growth factor $\mu(\hat{L}, \hat{R})$ (28) that measures by how much the backward error, at most, can be greater than the relative residual. Finally, exact (relative) errors (when the exact solution is known) and the forward error bounds (43), (46), and (47) are shown in Tables 3, 6, and 9.

All results presented are computed in the MATLAB environment [19] with unit roundoff $\approx 2.2 \times 10^{-16}$, and $(\hat{L}, \hat{R})$ is obtained by solving the system $Zx = b$ (3)

TABLE 2
Backward error bounds for well-conditioned to moderately ill-conditioned problems.

m	n	α	$\|(R_1, R_2)\|_F$	$\frac{\|(R_1,R_2)\|_F}{(\alpha+\beta)\|(\hat{L},\hat{R})\|_F+\gamma}$	$\|\mathcal{H}^+ r\|_2$	u.b. $\eta(\hat{L}, \hat{R})$	$\mu(\hat{L}, \hat{R})$
4	4	1/2	1.02e−15	4.32e−17	4.11e−16	2.69e−16	6.24e+00
10	10	1/2	8.80e−15	9.29e−17	1.36e−15	8.91e−16	9.59e+00
6	6	1/16	2.30e−15	5.01e−17	3.77e−16	3.66e−16	7.30e+00
2	10		2.41e−15	5.44e−17	4.13e−16	3.42e−16	6.28e+00
8	4		3.38e−15	6.73e−17	3.99e−16	3.99e−16	5.93e+00
10	6		6.25e−15	7.23e−17	4.42e−16	4.42e−16	6.11e+00
10	10		1.35e−14	1.01e−16	9.54e−16	7.09e−16	7.00e+00
2	10		1.95e−15	5.80e−17	3.88e−16	2.97e−16	5.13e+00
8	4		2.09e−15	5.05e−17	3.58e−16	3.15e−16	6.23e+00
10	6		4.48e−15	5.84e−17	3.89e−16	3.89e−16	6.66e+00
10	10		1.72e−14	1.39e−16	9.36e−16	9.35e−16	6.73e+00

by Gaussian elimination with partial pivoting. In some cases we would (probably) obtain better results (for example, smaller residuals) if we use the generalized Schur methods [18]. Notice that some of the test problems are standard Sylvester equations that we solve as generalized Sylvester equations. As a consequence we should expect less favourable results compared to the standard case.

The *first* set of test problems have all small-normed solutions and the ratios Φ/Ψ are of size $O(1)$, i.e., the Dif-based and the structure-preserving condition numbers are similar. We also see that the relative residuals and the relative backward errors are both of the size $O(\epsilon)$, where ϵ denotes the relative machine precision, and the growth factors $\mu(\hat{L}, \hat{R})$ are of size $O(1)$. Finally, the normwise and componentwise forward error bounds are similar and quite accurate.

The *badly scaled data* illustrate that the ratio Φ/Ψ can be arbitrarily large and show that these problems are not really ill conditioned (Ψ of size $O(10^2)$). Notice that the relative backward errors are at the machine precision level but are much larger than the relative residuals. The componentwise forward error bounds are in most cases better than the normwise forward error bounds, but this is not always the case $(m = 2, n = 4, \alpha = 10^{16})$.

TABLE 3
Forward error bounds for well-conditioned to moderately ill-conditioned problems.

m	n	α	$\frac{\|(L-\hat{L},R-\hat{R})\|_F}{\|(\hat{L},\hat{R})\|_F}$	$\frac{\|Z^{-1}\|_2\|(R_1,R_2)\|_F}{\|(\hat{L},\hat{R})\|_F}$	$\frac{\|(L-\hat{L},R-\hat{R})\|_M}{\|(\hat{L},\hat{R})\|_M}$	$\frac{\|Z^{-1}r\|_M}{\|(\hat{L},\hat{R})\|_M}$	$\frac{\|\|Z^{-1}G\|_M}{\|(\hat{L},\hat{R})\|_M}$
4	4	1/2	4.15e−14	2.31e−11	9.41e−14	3.08e−12	2.00e−11
10	10	1/2	2.44e−07	4.29e−04	1.37e−06	3.90e−05	2.03e−04
6	6	1/16	9.78e−03	6.13e+01	3.38e−02	2.23e−01	1.14e+01
2	10		1.70e−14	5.99e−12	5.94e−14	4.01e−13	3.06e−12
8	4		5.37e−13	5.34e−11	1.62e−12	2.57e−12	1.25e−11
10	6		1.56e−13	1.09e−09	6.51e−13	9.35e−12	1.35e−10
10	10		6.85e−11	3.10e−08	3.37e−10	3.69e−10	7.76e−09
2	10		1.17e−13	2.84e−11	2.73e−13	4.44e−12	1.07e−11
8	4		7.88e−12	8.25e−10	2.28e−11	1.09e−10	4.53e−10
10	6		5.00e−10	1.26e−05	1.37e−09	4.96e−07	6.45e−07
10	10		6.24e−10	3.98e−07	2.55e−09	1.01e−08	1.65e−07

The *third* set of (ill-conditioned) test problems illustrates the situation when the relative backward error is much larger than the relative residual. The problems

TABLE 4
Condition numbers for badly scaled data.

m	n	α	$\|(\hat{L},\hat{R})\|_F$	$\sigma_{\min}(Z)$	Φ	Ψ	Φ/Ψ
4	2	10^{13}	5.28e+02	1.71e−03	9.16e+15	1.90e+03	4.81e+12
2	4	10^{13}	3.98e+02	1.20e−03	9.80e+15	2.53e+02	3.87e+13
4	2	10^{16}	9.89e+00	2.16e−02	6.89e+17	1.49e+02	4.62e+15
2	4	10^{16}	4.28e+02	4.92e−02	2.37e+17	4.94e+03	4.79e+13
4	2	10^{19}	2.29e+00	0	∞	2.71e+01	∞
2	4	10^{19}	5.37e+02	0	∞	8.70e+02	∞

TABLE 5
Backward error bounds for badly scaled data.

m	n	α	$\|(R_1,R_2)\|_F$	$\frac{\|(R_1,R_2)\|_F}{(\alpha+\beta)\|(\hat{L},\hat{R})\|_F+\gamma}$	$\|\mathcal{H}^+ r\|_2$	u.b. $\eta(\hat{L},\hat{R})$	$\mu(\hat{L},\hat{R})$
4	2	10^{13}	4.73e−14	5.73e−30	4.08e−17	1.86e−14	3.24e+15
2	4	10^{13}	3.48e−14	7.41e−30	4.41e−17	1.66e−14	2.24e+15
4	2	10^{16}	1.30e−15	8.84e−33	1.02e−16	5.60e−16	6.34e+16
2	4	10^{16}	7.05e−14	1.41e−32	8.16e−17	3.45e−14	2.44e+18
4	2	10^{19}	3.40e−16	1.22e−35	1.15e−16	1.66e−16	1.36e+19
2	4	10^{19}	3.77e−14	6.12e−36	7.56e−16	1.57e−14	2.56e+21

TABLE 6
Forward error bounds for badly scaled data.

m	n	α	$\frac{\|Z^{-1}\|_2\|(R_1,R_2)\|_F}{\|(\hat{L},\hat{R})\|_F}$	$\frac{\|Z^{-1}r\|_M}{\|(\hat{L},\hat{R})\|_M}$	$\frac{\|\|Z^{-1}G\|_M}{\|(\hat{L},\hat{R})\|_M}$
4	2	10^{13}	2.33e−12	5.16e−15	9.61e−15
2	4	10^{13}	2.89e−12	3.46e−15	3.54e−14
4	2	10^{16}	2.13e−13	3.01e−14	7.12e−14
2	4	10^{16}	8.19e−14	2.66e−13	2.86e−13
4	2	10^{19}	∞	8.77e−15	6.30e−14
2	4	10^{19}	∞	6.11e−14	7.15e−14

TABLE 7
Condition numbers for ill-conditioned problems.

m	n	α	$\|(\hat{L},\hat{R})\|_F$	$\sigma_{\min}(Z)$	Φ	Ψ	Φ/Ψ
3	3	10^{-2}	8.49e+10	1.18e−11	3.80e+11	1.11e+07	3.43e+04
3	3	10^{-3}	8.49e+15	9.61e−17	4.65e+16	1.11e+10	4.21e+06
2	2	10^{-6}	6.30e+11	2.21e−16	2.22e+16	3.46e+12	6.41e+03
2	2	10^{-6}	2.83e+18	2.21e−16	2.22e+16	4.90e+12	4.53e+03

are characterized by a large-normed solution $(\hat{L},\hat{R})$, a small $\mathrm{Dif}[(A,D),(B,E)]$ (Z is almost singular) and a quite large Ψ. The condition numbers $\kappa(\hat{L}),\kappa(\hat{R})$ of the first example are moderate and, therefore, the computed solutions are nevertheless quite accurate (relative backward errors and componentwise forward error bounds of size $O(10^2\epsilon)$). Note that the componentwise forward error bounds are much smaller than the normwise error bound for this example. However, in the second example $\kappa(\hat{L}),\kappa(\hat{R})$ are large ($(\hat{L},\hat{R})$ are large normed and ill conditioned [13]) and the computed solutions have almost no accuracy at all. It is clear that the last example reflects an extremely ill-conditioned (generalized) Sylvester equation, but it is still an open problem (in general) to identify the exact conditions for an ill-conditioned

TABLE 8
Backward error bounds for ill-conditioned problems.

m	n	α	$\|(R_1,R_2)\|_F$	$\frac{\|(R_1,R_2)\|_F}{(\alpha+\beta)\|(\hat{L},\hat{R})\|_F+\gamma}$	$\|\mathcal{H}^+r\|_2$	u.b. $\eta(\hat{L},\hat{R})$	$\mu(\hat{L},\hat{R})$
3	3	10^{-2}	4.46e−12	1.17e−23	1.69e−15	4.23e−14	3.60e+09
3	3	10^{-3}	1.19e−07	3.14e−24	1.91e−15	1.13e−10	3.60e+13
2	2	10^{-6}	4.10e−05	1.33e−17	3.05e−05	4.00e−05	3.02e+12
2	2	10^{-6}	1.25e+01	1.13e−17	5.18e−05	9.01e−05	8.00e+12

TABLE 9
Forward error bounds for ill-conditioned problems.

m	n	α	$\frac{\|Z^{-1}\|_2\|(R_1,R_2)\|_F}{\|(\hat{L},\hat{R})\|_F}$	$\frac{\|Z^{-1}r\|_M}{\|(\hat{L},\hat{R})\|_M}$	$\frac{\|\|Z^{-1}G\|_M}{\|(\hat{L},\hat{R})\|_M}$
3	3	10^{-2}	1.60e−04	2.66e−15	2.24e−14
3	3	10^{-3}	1.96e+01	1.99e−15	2.27e−14
2	2	10^{-6}	1.61e+01	2.77e−01	8.87e+03
2	2	10^{-6}	1.60e+01	1.01e+00	8.88e+03

(generalized) Sylvester equation.

In all of our test examples we see that the componentwise error bounds (46) and (47) overall give very similar results. In most cases, the approximate error bound (47) gives the sharpest bounds.

Acknowledgments. I am grateful to Nick Higham and Ji Guang Sun for reading and commenting on the manuscript.

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, AND D. SORENSEN, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[2] M. ARIOLI, J. DEMMEL, AND I. S. DUFF, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 165–190.

[3] R. H. BARTELS AND G. W. STEWART, *Solution of the equation $AX + XB = C$*, Comm. Assoc. Comput. Mach., 15 (1972), pp. 820–826.

[4] J. BUNCH, *The weak and strong stability of algorithms in numerical linear algebra*, Linear Algebra Appl., 88/89 (1987), pp. 49–66.

[5] K-W. E. CHU, *The solution of the matrix equations $AXB - CXD = E$ and $(YA - DZ, YC - BZ) = (E, F)$*, Linear Algebra Appl., 93 (1987), pp. 93–105.

[6] J. DEMMEL AND B. KÅGSTRÖM, *Computing stable eigendecompositions of matrix pencils*, Linear Algebra Appl., 88/89 (1987), pp. 139–186.

[7] J. D. GARDINER, A. L. LAUB, J. A. AMATO, AND C. B. MOLER, *Solution of the Sylvester matrix equation $AXB^T + CXD^T = E$*, ACM Trans. Math. Software, 18 (1992), pp. 223–231.

[8] G. GOLUB, S. NASH, AND C. VAN LOAN, *A Hessenberg-Schur method for the matrix problem $AX + XB = C$*, IEEE Trans. Automat. Control, AC-24 (1979), pp. 909–913.

[9] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, Second Ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[10] N. GOULD, *On growth in Gaussian elimination with pivoting*, SIAM J. Matrix Anal. Appl., 12 (1990), pp. 354–361.

[11] W. W. HAGER, *Condition estimators*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 311–316.

[12] N. J. HIGHAM, ALGORITHM 674: *Fortran codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation*, ACM Trans. Math. Software, 15 (1989), p. 168.

[13] ———, *Perturbation theory and backward error for $AX - XB = C$*, Numerical Analysis Report No. 211, Department of Mathematics, University of Manchester, Manchester M13 9PL, England, 1992; BIT, to appear.

[14] B. KÅGSTRÖM, *A Direct Method for Reordering Eigenvalues in the Generalized Real Schur Form of a Regular Matrix Pair* (A, B), in Linear Algebra for Large Scale and Real–Time Applications, M. S. Moonen, G. H. Golub, and B. L. R. De Moor, eds., Kluwer Academic Publishers, Amsterdam, 1993, pp. 195–218.

[15] B. KÅGSTRÖM AND P. POROMAA, LAPACK-*style Algorithms and Software for Solving the Generalized Sylvester Equation and Estimating the Separation between Regular Matrix Pairs*, Report UMINF-93.23, Institute of Information Processing, University of Umeå, S-901 87 Umeå, Sweden, 1993.

[16] B. KÅGSTRÖM AND P. VAN DOOREN, *Additive decomposition of a transfer function with respect to a specified region*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Proceedings of the International Symposium MTNS-89, Vol. 3, Birkhauser Boston Inc, 1990, pp. 469–477.

[17] ———, *A generalized state-space approach for the additive decomposition of a transfer matrix*, Internat. J. Numer. Linear Algebra Appl., 1 (1992), pp. 165–181.

[18] B. KÅGSTRÖM AND L. WESTIN, *Generalized Schur methods with condition estimators for solving the generalized Sylvester equation*, IEEE Trans. Autom. Contr., 34 (1989), pp. 745–751.

[19] C. MOLER, J. LITTLE, AND S. BANGERT, PRO-MATLAB *Users' Guide*, The Math Works Inc., Sherborn, MA, 1987.

[20] G. W. STEWART, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.

[21] G. W. STEWART AND J. G. SUN, *Matrix Perturbation Theory*, Academic Press, New York, 1990.

[22] J. VARAH, *On the separation of two matrices*, SIAM J. Numer. Anal., 16 (1979), pp. 216–222.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1061–1074, October 1994

002

UNIFORM STABILITY OF MARKOV CHAINS*

ILSE C. F. IPSEN† AND CARL D. MEYER‡

Abstract. By deriving a new set of tight perturbation bounds, it is shown that all stationary probabilities of a finite irreducible Markov chain react essentially in the same way to perturbations in the transition probabilities. In particular, if at least one stationary probability is insensitive in a relative sense, then all stationary probabilities must be insensitive in an absolute sense. New measures of sensitivity are related to more traditional ones, and it is shown that all relevant condition numbers for the Markov chain problem are small multiples of each other. Finally, the implications of these findings to the computation of stationary probabilities by direct methods are discussed, and the results are applied to stability issues in nearly transient chains.

Key words. Markov chains, stationary distribution, stochastic matrix, sensitivity analysis, perturbation theory, stability of a Markov chain, condition numbers

AMS subject classifications. 65U05, 65F35, 60J10, 60J20, 15A51, 15A12, 15A18

1. Introduction.

The purpose of this paper is to analyse the sensitivity of individual stationary probabilities to perturbations in the transition probabilities of finite irreducible Markov chains. In addition to providing perturbation bounds that are much sharper than the traditional bounds, our analysis demonstrates that all stationary probabilities in an irreducible chain react in a somewhat uniform manner to perturbations in the transition probabilities. This property of uniform sensitivity markedly distinguishes Markov problems from general linear systems. Examples are presented in §3 to illustrate why a Markov problem should not be treated as just another linear system.

Previous perturbation theory for irreducible chains focused on the derivation of norm-based bounds of the following kind. Let P and $\tilde{P} = P + E$ be transition probability matrices with respective stationary probability vectors π^T and $\tilde{\pi}^T$ satisfying

$$\pi^T P = \pi^T, \quad \tilde{\pi}^T \tilde{P} = \tilde{\pi}^T, \quad \sum_i \pi_i = 1 = \sum_i \tilde{\pi}_i.$$

For suitable vector and matrix norms, it is known that

$$\|\pi^T - \tilde{\pi}^T\| \leq \kappa \|E\|$$

where values for the condition number κ can be derived in various ways. Schweitzer (1968) derives a value for κ from the fundamental matrix of Kemeny and Snell (1960) whereas the group inverse $A^{\#}$ of $A = I - P$ is used by Meyer (1980), Golub and Meyer (1986), Funderlic and Meyer (1986), Meyer (1994), Meyer and Stewart (1988), Barlow (1993), and Stewart (1991). Seneta (1991) suggests using a coefficient of ergodicity for κ.

---

*Received by the editors September 24, 1992; accepted for publication (in revised form) July 26, 1993.

†Computer Science, Yale University, New Haven, Connecticut 06520 (ipsen@cs.yale.edu). The work of this author was supported in part by National Science Foundation grant CCR-9102853.

‡Mathematics Department, North Carolina State University, Raleigh, North Carolina 27695–8205 (meyer@math.ncsu.edu). The work of this author was supported in part by National Science Foundation grants DMS-9020915 and DDM-8906248.

These norm-based bounds are not satisfying for two reasons. First, there exist irreducible chains for which the bounds are not tight, so the condition number κ may seriously overestimate the sensitivity to perturbations. Secondly, the bounds generally provide little information about the relative error $|\pi_j - \tilde{\pi}_j|/\pi_j$ in individual stationary probabilities. We remedy this situation in §4 by deriving tight perturbation bounds for individual stationary probabilities. On the basis of these bounds, we prove a uniform sensitivity theorem saying that if at least one stationary probability has low relative sensitivity, or if at least one large stationary probability has low absolute sensitivity, then all probabilities have low absolute sensitivity.

In §5 we relate our measure of sensitivity to the traditional condition numbers for the Markov problem, and we prove that all relevant condition numbers for the problem $\pi^T A = 0$ are small multiples of each other. After discussing the ramifications of the perturbation results on direct methods for computing the stationary probabilities, we consider the case of nearly transient chains in §§6 and 7. We show that under special perturbations even small stationary probabilities may have low relative sensitivity. In addition, we give conditions under which a nearly transient chain is absolutely stable under general perturbations.

2. Norms and notation. Throughout the article the infinity-norm is exclusively used for matrices and column vectors, and the one-norm is used for row vectors. Since it will always be clear from the context whether a quantity is a matrix, column, or row, the subscripts on $\| \star \|_\infty$ and $\| \star \|_1$ are suppressed. Row vectors will always be transposed (e.g., π^T), and column vectors will be untransposed. The jth column of the identity matrix I is denoted by e_j and the column of all ones is denoted by e. The matrix P denotes the transition probability matrix of an n-state irreducible Markov chain with stationary distribution π^T whose entries satisfy $\pi_i > 0$ and $\sum_i \pi_i = 1$. We define $A = I - P$ and $A^\#$ denotes the group inverse of A, properties of which can be found in Campbell and Meyer (1991), Meyer (1975), and Meyer (1982). The matrix $\tilde{P} = P + E$ is a perturbation of P that represents the transition matrix of another irreducible chain with stationary distribution $\tilde{\pi}^T$. The perturbation matrix E is not necessarily constrained to be small. We use $E^{(j)}$ to denote the matrix obtained by deleting the jth column of E, and A_j denotes the principal submatrix obtained by deleting the jth row and column from $A = I - P$. Finally, we let N denote the matrix obtained by replacing the last column of A by a column of ones.

3. Absolutely stable chains. The solution of a general ill-conditioned linear system $Ax = b$ need not be uniformly sensitive to small perturbations. Some components of x can be sensitive while others are not. Furthermore, as shown in Chandrasekaran and Ipsen (1992), the sensitivity of the x_i's need not be a result of their size. Our purpose is to demonstrate that this cannot happen for Markov chains, but first it is important to distinguish between absolute sensitivity and relative sensitivity in the Markov chain setting.

Example 3.1. For the three-state chain whose transition matrix is

$$P(\epsilon) = \begin{pmatrix} 0 & 1-\epsilon & \epsilon \\ 1-\epsilon & 0 & \epsilon \\ 1 & 0 & 0 \end{pmatrix},$$

the associated stationary distribution is

$$\pi^T(\epsilon) = \left(\frac{1}{(2-\epsilon)(1+\epsilon)}, \frac{1-\epsilon}{(2-\epsilon)(1+\epsilon)}, \frac{\epsilon}{1+\epsilon} \right).$$

If $P = P(10^{-8})$ is perturbed to become $\tilde{P} = P(10^{-4})$, then the magnitude of the perturbation $E = \tilde{P} - P$ is

$$\|E\| = 2(10^{-4} - 10^{-8}).$$

Consider the change in the respective stationary distributions

$$\pi^T = \pi^T(10^{-8}) \quad \text{and} \quad \tilde{\pi}^T = \pi^T(10^{-4}).$$

The *absolute change* (the change relative to 1) in π_3 is

$$|\pi_3 - \tilde{\pi}_3| = \left| \frac{10^{-8}}{1+10^{-8}} - \frac{10^{-4}}{1+10^{-4}} \right| = \frac{10^{-4} - 10^{-8}}{(1+10^{-4})(1+10^{-8})} \approx 10^{-4} - 10^{-8} = \frac{\|E\|}{2},$$

but the *relative change* (the change relative to the original value) is

$$\left| \frac{\pi_3 - \tilde{\pi}_3}{\pi_3} \right| = \left| 1 - \frac{10^{-4}(1+10^{-8})}{10^{-8}(1+10^{-4})} \right| \approx 10^4.$$

If the change in probabilities is assessed in an absolute sense by comparing it to 1, then π_3 is not at all sensitive to the perturbation because the change of magnitude $\|E\|$ in the transition probabilities produces a change in π_3 of only $\|E\|/2$. We say that π_3 is *absolutely insensitive*. But if the change in probabilities is assessed in a relative sense then the change in π_3 is large, so π_3 is *relatively sensitive*. As for the sensitivity of the other two probabilities π_1 and π_2, if $a_{ij}^{\#}$ is element (i,j) in the group inverse $A^{\#}$ of $A = I - P$, then, as shown by Funderlic and Meyer (1986), the absolute error in the jth stationary probability is bounded by

$$|\pi_j - \tilde{\pi}_j| \leq \kappa_j \|E\|, \qquad \kappa_j = \max_i |a_{ij}^{\#}|.$$

In this example, $\max_{i,j} |a_{ij}^{\#}| < 1$, so all three stationary probabilities are insensitive in the absolute sense. Because π_1 and π_2 are both very close to .5, they are insensitive in the relative sense as well. This example motivates the following definition.

DEFINITION 3.1. *An irreducible chain is said to be* absolutely stable *whenever each π_j is insensitive to perturbations in P in the absolute sense; i.e., whenever there is a small constant κ such that for all perturbations E,*

$$|\pi_j - \tilde{\pi}_j| \leq \kappa \|E\| \quad \textit{for each } 1 \leq j \leq n,$$

where the term "small" is to be interpreted in the context of the underlying application.

Sufficient conditions for absolute stability are well-known. The results in Barlow (1993), Funderlic and Meyer (1986), Golub and Meyer (1986), Meyer (1980), Meyer (1994), Meyer and Stewart (1988), Stewart (1991), for instance, use the fact that a chain is absolutely stable if the group inverse $A^{\#}$ of $A = I - P$ has no large entries (relative to 1). The results of §5 will establish that the converse of this statement is also true.

4. Componentwise analysis. In this section we derive tight upper bounds on the relative change in individual stationary probabilities, and we prove that all stationary probabilities show essentially the same sensitivity to perturbations in the transition probabilities.

We make use of the following properties of M–matrices, details of which can be found in the text by Berman and Plemmons (1979). If P is an irreducible stochastic matrix of order n, then $A = I - P$ is a singular M–matrix of rank $n-1$. Moreover, if A_j is the principal submatrix of A obtained by deleting the jth row and column from A, then A_j is a nonsingular M–matrix. Hence $A_j^{-1} > 0$, and if e is the column vector of all ones, then $\|A_j^{-1}e\| = \|A_j^{-1}\|$. The following theorem demonstrates that the entries in A_j^{-1} determine the relative sensitivity of the jth stationary probability to perturbations in the transition probabilities.

THEOREM 4.1. *If $E^{(j)}$ denotes the matrix obtained by deleting the jth column of E, then*

$$\frac{\pi_j - \tilde{\pi}_j}{\pi_j} = \tilde{\pi}^T E^{(j)} A_j^{-1} e.$$

Furthermore,

$$\left| \frac{\pi_j - \tilde{\pi}_j}{\pi_j} \right| \leq \|E^{(j)}\| \, \|A_j^{-1}\|,$$

and there always exists a perturbation E (dependent on j) for which equality is attained.

Proof. By applying a symmetric permutation to P, the states may be reordered so that a particular stationary probability occurs in the last position of π^T. Thus it suffices to prove the theorem for $j = n$. With the partitioning

$$\pi^T = (\overline{\pi}^T \quad \pi_n), \qquad A = \begin{pmatrix} A_n & b \\ c^T & \delta \end{pmatrix},$$

$\pi^T A = 0^T$ implies $\overline{\pi}^T = -\pi_n c^T A_n^{-1}$. Replacing the last column of A by the vector of all ones produces a nonsingular matrix

$$N = \begin{pmatrix} A_n & e \\ c^T & 1 \end{pmatrix} \quad \text{with inverse} \quad N^{-1} = \begin{pmatrix} A_n^{-1}(I - e\overline{\pi}^T) & -\pi_n A_n^{-1} e \\ \overline{\pi}^T & \pi_n \end{pmatrix}.$$

The stationary distribution of the original chain is the solution of the system

$$\pi^T N = e_n^T \quad \text{where} \quad e_n^T = (0 \quad \cdots \quad 0 \quad 1),$$

and the stationary distribution for the perturbed chain is the solution of

$$\tilde{\pi}^T (N - F) = e_n^T \quad \text{where} \quad F = (E^{(n)} \quad 0).$$

Consequently,

$$\pi^T - \tilde{\pi}^T = -\tilde{\pi}^T F N^{-1}, \tag{4.1}$$

so

$$\pi_n - \tilde{\pi}_n = -\tilde{\pi}^T \begin{pmatrix} E^{(n)} & 0 \end{pmatrix} \begin{pmatrix} -\pi_n A_n^{-1} e \\ \pi_n \end{pmatrix} = \pi_n \left(\tilde{\pi}^T E^{(n)} A_n^{-1} e \right),$$

and therefore

$$\frac{\pi_n - \tilde{\pi}_n}{\pi_n} = \tilde{\pi}^T E^{(n)} A_n^{-1} e.$$

Applying Hölder's inequality and $\|A_n^{-1}e\| = \|A_n^{-1}\|$ yields

$$\left|\frac{\pi_n - \tilde{\pi}_n}{\pi_n}\right| \leq \|\tilde{\pi}\| \, \|E^{(n)} A_n^{-1} e\| \leq \|E^{(n)}\| \, \|A_n^{-1}\|.$$

To see that equality is always attainable, let k be the position where the largest component of $A_n^{-1}e$ occurs so that

$$e_k^T A_n^{-1} e = \|A_n^{-1}e\| = \|A_n^{-1}\|,$$

and let $E = \epsilon e(e_k - e_n)^T$. Then $\tilde{\pi}^T E^{(n)} = \epsilon e_k^T$ and $\|E^{(n)}\| = \epsilon$, so that

$$\frac{\pi_n - \tilde{\pi}_n}{\pi_n} = \tilde{\pi}^T E^{(n)} A_n^{-1} e = \epsilon e_k^T A_n^{-1} e = \epsilon \|A_n^{-1}\| = \|E^{(n)}\| \, \|A_n^{-1}\|. \qquad \square$$

COROLLARY 4.1. *An irreducible chain is absolutely stable if and only if* $\pi_j \|A_j^{-1}\|$ *is small for every* $1 \leq j \leq n$.

The results of Theorem 4.1 and its corollary suggest the following definitions.

DEFINITION 4.1. *Let* A_j *be the principal submatrix obtained by deleting the* j*th row and column from* A, *and let* π_j *denote the* j*th stationary probability. The* relative condition number *for* π_j *is defined to be*

$$\rho_j = \|A_j^{-1}\| \quad \textit{and we set} \quad \rho = \min_j \{\rho_j\}.$$

The absolute condition number *for* π_j *is defined to be*

$$\alpha_j = \pi_j \|A_j^{-1}\| \quad \textit{and we set} \quad \alpha = \max_j \{\alpha_j\}.$$

In terms of this notation, Theorem 4.1 states

$$\left|\frac{\pi_j - \tilde{\pi}_j}{\pi_j}\right| \leq \rho_j \, \|E^{(j)}\|, \quad |\pi_j - \tilde{\pi}_j| \leq \alpha_j \, \|E\|, \quad \text{and} \quad \|\pi - \tilde{\pi}\| \leq \alpha \|E\|,$$

so α is the *absolute condition number for the entire chain.*

Notice that if a stationary probability is relatively well-conditioned, then it is absolutely well-conditioned but not conversely, cf., Example 3.1. It may be of interest to note that the existence of a small ρ_j means that the $(n-1)$st singular value of A is large (Barlow (1993)).

We now arrive at one of our principal conclusions which states that the sensitivity of the stationary distribution is uniform in the sense that all π_j's are absolutely well-conditioned if and only if at least one π_j is relatively well-conditioned.

THEOREM 4.2. *For every* $1 \leq j \leq n$,

$$|\pi_j - \tilde{\pi}_j| \leq \rho \, \|E\|.$$

Consequently, an irreducible chain is absolutely stable if and only if at least one π_j *is relatively well-conditioned.*

Proof. As in the proof of Theorem 4.1, assume that the states have been permuted so the best relatively conditioned stationary probability is in the last position; i.e., $\rho_n = \rho$. If

$$N = \begin{pmatrix} A_n & e \\ c^T & 1 \end{pmatrix}$$

is the matrix obtained by replacing the last column of A by ones then, as in (4.1),

$$\pi^T - \tilde{\pi}^T = -\tilde{\pi}^T F N^{-1}, \quad \text{where} \quad F = \begin{pmatrix} E^{(n)} & 0 \end{pmatrix}.$$

From

$$N^{-1} = \begin{pmatrix} A_n^{-1}(I - e\overline{\pi}^T) & -\pi_n A_n^{-1} e \\ \overline{\pi}^T & \pi_n \end{pmatrix} = \begin{pmatrix} A_n^{-1} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I - e\overline{\pi}^T & -\pi_n e \\ \overline{\pi}^T & \pi_n \end{pmatrix}, \tag{4.2}$$

it follows that

$$\pi_j - \tilde{\pi}_j = -\tilde{\pi}^T F N^{-1} e_j = \begin{cases} -\tilde{\pi}^T E^{(n)} A_n^{-1}(e_j - \pi_j e) & \text{if} \quad j < n. \\ -\tilde{\pi}^T E^{(n)} A_n^{-1}(-\pi_j e) & \text{if} \quad j = n. \end{cases}$$

Since $\|e_j - \pi_j e\| = \max\{\pi_j, 1 - \pi_j\} < 1$ and $\|A_n^{-1} e\| = \|A_n^{-1}\| = \rho_n$, we have that

$$|\pi_j - \tilde{\pi}_j| \leq \rho_n \|E^{(n)}\| \leq \rho_n \|E\|, \qquad 1 \leq j \leq n.$$

Therefore, if at least one stationary probability is relatively well-conditioned, then all stationary probabilities are absolutely well-conditioned. The converse follows from Corollary 4.1 because at least one π_j must be greater than or equal to $1/n$. □

The following two statements are direct consequences of Theorem 4.2, but they are important to state because they drive home the extent to which there exists uniform stability in Markov chains.

COROLLARY 4.2. *If any stationary probability is relatively well-conditioned, then all large stationary probabilities are relatively well-conditioned.*

COROLLARY 4.3. *If any large stationary probability is absolutely well-conditioned, then the chain is absolutely stable.*

A natural question arises at this point. We know that the existence of one relatively well-conditioned π_j implies the chain is absolutely stable, but does the existence of one absolutely well-conditioned π_j insure absolute stability? Unfortunately, the answer is "no," and this can be seen by considering

$$P = \begin{pmatrix} 1-\epsilon & \epsilon/2 & \epsilon/2 \\ \epsilon/2 & 1-\epsilon & \epsilon/2 \\ 1/2 & 1/2 & 0 \end{pmatrix}, \qquad \pi^T = \frac{1}{2+\epsilon}\begin{pmatrix} 1 & 1 & \epsilon \end{pmatrix}$$

for small $0 < \epsilon < 1$. The absolute and relative condition numbers are

$$\alpha_1 = \alpha_2 = \frac{1}{2+\epsilon}\left(\frac{2}{3} + \frac{4}{3\epsilon}\right), \quad \alpha_3 = \frac{2}{2+\epsilon}, \quad \rho_1 = \rho_2 = \frac{2}{3} + \frac{4}{3\epsilon}, \quad \text{and} \quad \rho_3 = \frac{2}{\epsilon},$$

so, for small ϵ, π_3 is absolutely well-conditioned, but π_1 and π_2 are not. The chain is not absolutely stable, and no π_j is relatively well-conditioned.

Small stationary probabilities are the ones that appear least likely to be relatively well-conditioned. Therefore it makes sense to try to determine features that may be responsible for the small size. The following theorem shows that those π_j whose associated submatrix A_j is well-conditioned cannot be small. It also shows that a nearly reducible matrix A that is far from being uncoupled produces small π_j.

THEOREM 4.3. *If Q is a permutation matrix such that*

$$Q^T A Q = \begin{pmatrix} A_j & b_j \\ c_j^T & \delta_j \end{pmatrix},$$

then

$$\frac{1}{1+\rho_j} \leq \pi_j \leq \frac{\|b_j\|}{\|c_j^T\| + \|b_j\|}.$$

Proof. Let $\pi^T Q = \psi^T = \begin{pmatrix} \overline{\psi}^T & \pi_j \end{pmatrix}$. Since $\psi^T A = 0$ implies $\overline{\psi}^T = -\pi_j c_j^T A_j^{-1}$, Hölder's inequality gives the lower bound

$$1 - \pi_j = \overline{\psi}^T e = \pi_j |c_j^T A_j^{-1} e| \leq \pi_j \rho_j.$$

To obtain the upper bound, use $\|c_j^T\| = -c_j^T e = \delta_j$ and $\delta_j \pi_j = -\overline{\psi}^T b_j$, and again apply Hölder's inequality,

$$\pi_j \|c_j^T\| = \pi_j \delta_j = -\overline{\psi}^T b_j \leq \|\overline{\psi}^T\| \, \|b_j\| = (1 - \pi_j)\|b_j\|. \qquad \square$$

5. Condition numbers and linear systems. It was demonstrated in the previous section that the sensitivity of the stationary distribution is governed by ρ. We now compare this measure of sensitivity to other condition numbers, and we relate these results to numerical techniques for computing stationary probabilities by solving certain linear systems.

The nonsingular matrix

$$N = \begin{pmatrix} A_n & e \\ c^T & 1 \end{pmatrix} \quad \text{and the associated system} \quad \pi^T N = e_n^T$$

are focal points of the development. The expression (4.2) together with the fact that $\rho_n = \|A_n^{-1}\| \geq 1$ (because $e = -A_n^{-1} b$) produces

$$1 \leq \|N^{-1}\| \leq 2\rho_n. \tag{5.1}$$

This means that if π_n is relatively well-conditioned, then $\pi^T N = e_n^T$ is a well-conditioned nonsingular system and therefore any stable algorithm can accurately solve it. But it is not clear that the solution of $\pi^T N = e_n^T$ should be attempted when ρ_n is large, even if the chain is absolutely stable. Theorem 4.1 insures that some ρ_j must be small, but, as Example 3.1 demonstrates, it need not be ρ_n. Of course, safety can be guaranteed if one is willing to determine a value of k such that $\rho_k = \rho$ because the same logic that produced (5.1) insures that the system $\pi^T \hat{N} = e_k^T$ is well-conditioned where $\hat{N}$ is the nonsingular matrix obtained by replacing the kth column of A by e. But determining ρ (or its position) is prohibitively expensive, and this may be why this approach is dismissed as "naive" by Paige, Styan, and Wachter (1975) and not included in their comparisons.

Surprisingly, it does not matter which column of A is replaced by e. This is a consequence of the next theorem that relates N^{-1} to the group inverse $A^{\#}$.

THEOREM 5.1. *For the numbers* α *and* ρ *given in Definition* 4.1,

$$\frac{\|A^{\#}\|}{2} \leq \|N^{-1}\| \leq 2\,\|A^{\#}\| + 1$$

and

$$\frac{\alpha}{2} \leq \|A^{\#}\| \leq 4\rho.$$

Proof. We derive the upper bounds first. It is easily verified (Meyer (1975)) that

$$\begin{aligned} A^{\#} &= (I - e\pi^T)\begin{pmatrix} A_n^{-1} & 0 \\ 0 & 0 \end{pmatrix}(I - e\pi^T) \\ &= \begin{pmatrix} (I - e\overline{\pi}^T)A_n^{-1}(I - e\overline{\pi}^T) & -\pi_n(I - e\overline{\pi}^T)A_n^{-1}e \\ -\overline{\pi}^T A_n^{-1}(I - e\overline{\pi}^T) & \pi_n \overline{\pi}^T A_n^{-1} e \end{pmatrix}. \end{aligned} \tag{5.2}$$

A symmetric permutation can bring any principal submatrix A_j of A to the upper left-hand corner of $Q^T A Q$. Then $(Q^T A Q)^{\#} = Q^T A^{\#} Q$, and

$$\psi^T = \pi^T Q = \begin{pmatrix} \overline{\psi}^T & \pi_j \end{pmatrix}$$

imply

$$\begin{aligned} Q^T A^{\#} Q &= (I - e\psi^T)\begin{pmatrix} A_j^{-1} & 0 \\ 0 & 0 \end{pmatrix}(I - e\psi^T) \\ &= \begin{pmatrix} (I - e\overline{\psi}^T)A_j^{-1}(I - e\overline{\psi}^T) & -\pi_j(I - e\overline{\psi}^T)A_j^{-1}e \\ -\overline{\psi}^T A_j^{-1}(I - e\overline{\psi}^T) & \pi_j \overline{\psi}^T A_j^{-1} e \end{pmatrix}. \end{aligned}$$

The second upper bound is now immediate because

$$\|A^{\#}\| = \|Q^T A^{\#} Q\| \leq 4\rho_j \quad \text{for all } j.$$

The first upper bound follows from

$$N^{-1} = \begin{pmatrix} I & -e \\ -c^T & -\delta \end{pmatrix} A^{\#} + e_n e_n^T,$$

which can be verified by using (5.2), so that $\|N^{-1}\| \leq 2\|A^{\#}\| + 1$. To establish the lower bounds, use the expressions for $A^{\#}$ and $Q^T A^{\#} Q$ to write

$$A^{\#} = \begin{pmatrix} I - e\overline{\pi}^T & 0 \\ -\overline{\pi}^T & 0 \end{pmatrix} N^{-1} \quad \text{and} \quad \pi_j A_j^{-1} = \begin{pmatrix} I & -e \end{pmatrix} Q^T A^{\#} Q \begin{pmatrix} \pi_j I \\ -\overline{\psi}^T \end{pmatrix}.$$

Hence $\|A^{\#}\| \leq 2\|N^{-1}\|$ and, for every j,

$$\alpha_j = \pi_j \|A_j^{-1}\| \leq 2\|A^{\#}\|. \qquad \square$$

The group inverse is relevant because

$$\pi - \tilde{\pi} = \tilde{\pi} E A^{\#} \quad \text{and} \quad |\pi_j - \tilde{\pi}_j| \leq \|E\|\,\|A^{\#}\|, \tag{5.3}$$

(Meyer (1980)), so if $\|A^{\#}\|$ is small, then the chain is absolutely stable. While conjectured, the converse of this statement has never been proven. However, on the basis of Theorems 4.2 and 5.1, the converse is now evident.

The logic used in proving Theorem 5.1 dictates that replacing *any* column of A by e results in a well-conditioned matrix when the chain is absolutely stable, and when the chain is not absolutely stable, all such matrices are ill-conditioned. Consequently, it does not matter which column of A is replaced by 1's, so the problem addressed by Harrod and Plemmons (1984) and Barlow (1986, 1993) of having to locate a well-conditioned principal submatrix A_j in order to build a well-conditioned system is obviated. Furthermore, since N is nonsingular, standard numerical techniques[1] can be applied to solve $\pi^T N = e_n^T$.

So far we have viewed the stationary distribution π^T as a solution to two different linear systems; the singular system $\pi^T A = 0$ and the nonsingular system $\pi^T N = e_n^T$. There is a yet a third linear system of which π^T is a solution, namely

$$\pi^T M = e_{n+1}^T \quad \text{where} \quad M = (A \quad e). \tag{5.4}$$

The augmented matrix M is of order $n \times (n+1)$ and has full row rank. The perturbed system is $\tilde{\pi}^T(M+E) = e_{n+1}^T$, so

$$\pi^T - \tilde{\pi}^T = \tilde{\pi}^T E M^{\dagger},$$

where $M^{\dagger}$ is the Moore-Penrose pseudo-inverse (Campbell and Meyer (1991)), and

$$|\pi_j - \tilde{\pi}_j| \leq \|\tilde{\pi}\| \, \|E\| \, \|M^{\dagger}\| \quad \text{for each } j.$$

Hence $\|M^{\dagger}\|$ is a condition number for measuring absolute stability. Another such number is $\|Z\|$ where $Z = (A + e\pi^T)^{-1}$ is the Kemeny and Snell (1960) fundamental matrix because $\pi^T - \tilde{\pi}^T = \tilde{\pi}^T E Z$ and $|\pi_j - \tilde{\pi}_j| \leq \|\tilde{\pi}\| \, \|E\| \, \|Z\|$ (Schweitzer (1968)). The following lemma shows that $\|M^{\dagger}\|$ and $\|Z\|$ are small multiples of each other and that they are not significantly different from $\|A^{\#}\|$.

LEMMA 5.1. *For the matrices M and Z defined above,*

$$\frac{\|Z\|}{3} \leq \|M^{\dagger}\| \leq 2\|Z\| \quad \text{and} \quad \|A^{\#}\| - 1 \leq \|Z\| \leq \|A^{\#}\| + 1.$$

Proof. The first set of inequalities follows from the identities

$$Z = \Big((I - e\pi^T) \;\; e\Big) M^{\dagger} \quad \text{and} \quad M^{\dagger} = \begin{pmatrix} I - ee^T/n \\ \pi^T \end{pmatrix} Z,$$

each of which is straightforward to verify. The second set of inequalities is a consequence of the fact that $Z = A^{\#} + e\pi^T$ (Meyer (1975)). □

Combining the results of Lemma 5.1 with those of Theorems 4.2 and 5.1 produces the following complete statement concerning the stability of irreducible Markov chains.

---

[1] Gaussian elimination with exact arithmetic generates positive pivots, but floating-point arithmetic may produce a zero or negative pivot (Funderlic and Mankin (1981)). This can be avoided with diagonal adjustment schemes as discussed by Grassmann, Taksar, and Heyman (1985), Stewart and Zhang (1991), and Barlow (1993).

THEOREM 5.2. *For an n-state irreducible Markov chain, the following statements are equivalent.*

- *At least one stationary probability is relatively well-conditioned.*
- *The chain is absolutely stable.*
- *All entries of the group inverse* $A^{\#}$ *are small.*
- *The matrix* N *and the system* $\pi^T N = e_n^T$ *are well-conditioned.*
- *The matrix* M *and the system* $\pi^T M = e_{n+1}^T$ *are well-conditioned.*
- *All entries in the Kemeny and Snell fundamental matrix* Z *are small.*

6. Sensitivity of nearly transient chains. In this section we examine the sensitivity of stationary probabilities of irreducible chains with nearly transient states; i.e., irreducible chains in which the states can be ordered so that the transition matrix is almost block triangular in the sense that

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \quad \text{with} \quad \|P_{21}\| << 1. \tag{6.1}$$

We prove two results, one for structured perturbations and one for more general perturbations.

The first theorem establishes a result similar to the one by Stewart (1992b). It says that *small* stationary probabilities of an absolutely stable chain are *relatively* well-conditioned if only the states corresponding to these probabilities are perturbed and all other states remain unaffected.

THEOREM 6.1. *If* E *can be symmetrically permuted so that*

$$E = \begin{pmatrix} 0 \\ E_2 \end{pmatrix} \quad \textit{with} \quad \|E\| = \epsilon,$$

and if $\pi^T = (\,\pi_1^T \quad \pi_2^T\,)$ *is partitioned conformably, then*

$$\frac{|\pi_j - \tilde{\pi}_j|}{\|\tilde{\pi}_2\|} \leq 4\epsilon\,\rho, \qquad 1 \leq j \leq n.$$

Proof. Combine (5.3) with the fact $\|A^{\#}\| \leq 4\rho$ from Theorem 5.1. □

The second theorem concerns nearly transient chains, but no restriction is placed on the structure of the perturbation matrix.

THEOREM 6.2. *Suppose* P_{11} *in* (6.1) *is* $s \times s$, *and let*

$$A = \begin{pmatrix} A_n & b \\ c^T & \delta \end{pmatrix}, \quad A_n = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad \textit{and} \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

where B_{11} *is* $s \times s$, b_1 *is* $s \times 1$, *and* $b_i \neq 0$ *for each* i. *The relative condition of* π_n *is bounded by*

$$\rho_n < \frac{2\max\{\|B_{11}^{-1}\|, \|B_{22}^{-1}\|\}}{1 - \|B_{22}^{-1}B_{21}\|},$$

so the chain is absolutely stable whenever B_{11} *and* B_{22} *have small inverses.*

Proof.

$$A_n = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ B_{21} & 0 \end{pmatrix} = T + K = T(I + T^{-1}K).$$

If $\|T^{-1}K\| < 1$, then, from results in §2.3.4 in Golub and Van Loan (1989),

$$\rho_n = \|A_n^{-1}\| \le \|T^{-1}\| \, \|(I + T^{-1}K)^{-1}\| \le \frac{\|T^{-1}\|}{1 - \|T^{-1}K\|}.$$

Since A is an M–matrix, $B_{ii}^{-1} > 0$, $B_{ij} \le 0$, and $b \le 0$. Consequently, $Ae = 0$ implies

$$0 \le -B_{11}^{-1}B_{12}e = e + B_{11}^{-1}b_1 \le e.$$

By assumption, $b_1 \ne 0$, so $B_{11}^{-1}b_1 < 0$, and thus

$$\|B_{11}^{-1}B_{12}\| = \|B_{11}^{-1}B_{12}e\| < 1.$$

A similar argument shows that $\|B_{22}^{-1}B_{21}\| < 1$. Since

$$T^{-1} = \begin{pmatrix} B_{11}^{-1} & -B_{11}^{-1}B_{12}B_{22}^{-1} \\ 0 & B_{22}^{-1} \end{pmatrix} = \begin{pmatrix} I & -B_{11}^{-1}B_{12} \\ 0 & I \end{pmatrix} \begin{pmatrix} B_{11}^{-1} & 0 \\ 0 & B_{22}^{-1} \end{pmatrix},$$

we have

$$\|T^{-1}\| \le (1 + \|B_{11}^{-1}B_{12}\|) \max\{\|B_{11}^{-1}\|, \|B_{22}^{-1}\|\} < 2\max\{\|B_{11}^{-1}\|, \|B_{22}^{-1}\|\}.$$

Similarly,

$$T^{-1}K = \begin{pmatrix} -B_{11}^{-1}B_{12}B_{22}^{-1}B_{21} & 0 \\ B_{22}^{-1}B_{21} & 0 \end{pmatrix}$$

implies

$$\|T^{-1}K\| \le \max\{\|B_{11}^{-1}B_{12}B_{22}^{-1}B_{21}\|, \|B_{22}^{-1}B_{21}\|\} < \|B_{22}^{-1}B_{21}\|,$$

so

$$\rho_n \le \frac{\|T^{-1}\|}{1 - \|T^{-1}K\|} < \frac{2\max\{\|B_{11}^{-1}\|, \|B_{22}^{-1}\|\}}{1 - \|B_{22}^{-1}B_{21}\|}. \qquad \square$$

7. Small probabilities in nearly transient chains. Let $\pi^T = (\bar{\pi}_1^T \quad \bar{\pi}_2^T)$ be the stationary distribution of the nearly transient matrix P in (6.1), and set

$$A = I - P = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad \text{where} \quad \|A_{21}\| = \|P_{21}\| = \epsilon.$$

Since $\bar{\pi}_1^T = -\bar{\pi}_2^T A_{21}A_{11}^{-1}$ implies $\|\bar{\pi}_1^T\| \le \epsilon\|A_{11}^{-1}\|$, we see that the trailing stationary probabilities dominate the leading ones provided $\|A_{11}^{-1}\|$ is not too large. For nearly transient chains with a finer block structure, say

$$(7.1) \qquad A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1k} \\ F_{21} & A_{22} & A_{23} & \cdots & A_{2k} \\ F_{31} & F_{32} & A_{33} & \cdots & A_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F_{k1} & F_{k2} & F_{k3} & \cdots & A_{kk} \end{pmatrix}, \quad \pi = \begin{pmatrix} \bar{\pi}_1 \\ \bar{\pi}_2 \\ \bar{\pi}_3 \\ \vdots \\ \bar{\pi}_k \end{pmatrix}, \quad \left\| \begin{pmatrix} F_{j+1,j} \\ F_{j+2,j} \\ F_{j+3,j} \\ \vdots \\ F_{k,j} \end{pmatrix} \right\| = \epsilon_j,$$

$1 \leq j \leq k-1$, the same should be true; i.e., the trailing stationary probabilities tend to be larger than the leading ones. We will quantify this statement by providing bounds in terms of ϵ_j on the probabilities $\bar{\pi}_j$ associated with each block.

The strategy is to proceed inductively by applying the above 2×2 case to successive diagonal blocks. This is accomplished by applying the following lemma that provides a perturbation of size ϵ that essentially uncouples A_{11} from the remaining blocks. In particular, the lemma shows that the remaining probabilities are the exact probabilities of a perturbed problem of the same form (the only difference being that the sum of the probabilities is less than one).

LEMMA 7.1. *Let*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

with $\|A_{21}\| \leq \epsilon$. *If*

$$\Delta = \frac{A_{21} e \bar{\pi}_1^T A_{12}}{\bar{\pi}_2^T A_{21} e},$$

then $A_{22} + \Delta$ *is a singular M–matrix such that*

$$\bar{\pi}_2^T (A_{22} + \Delta) = 0, \quad (A_{22} + \Delta)e = 0, \quad \textit{and} \quad \|\Delta\| \leq \epsilon.$$

Proof. We first verify that Δ satisfies the required equations. From $\pi^T A = 0$ and $Ae = 0$ we get $r_1^T = \bar{\pi}_2^T A_{22} = -\bar{\pi}_1^T A_{12}$ and $r_2 = A_{22}e = -A_{21}e$, so one can write

$$\Delta = -\frac{r_2 r_1^T}{\bar{\pi}_2^T r_2}.$$

Since $\bar{\pi}_2^T \Delta = -r_1^T$, it follows that $\bar{\pi}_2^T (A_{22} + \Delta) = 0$, and thus Δ satisfies the first equation. To prove that Δ satisfies the second equation, observe that $\pi^T A = 0$ and $Ae = 0$ imply

$$\bar{\pi}_2^T r_2 = -\bar{\pi}_2^T A_{21} e = \bar{\pi}_1^T A_{11} e = -\bar{\pi}_2^T A_{12} e = r_1^T e.$$

Thus,

$$\Delta = -\frac{r_2 r_1^T}{r_1^T e},$$

so $\Delta e = -r_2$ and $(A_{22} + \Delta)e = 0$. As for the bound on the norm of Δ, notice that r_1 and r_2 both consist entirely of nonnegative elements since A is an M–matrix so Δ consists entirely of nonpositive elements. This means

$$\|\Delta\| = \left\| \frac{r_2 r_1^T}{r_1^T e} e \right\| = \|r_2\| \leq \epsilon.$$

Moreover, since all elements of Δ are nonpositive, the off-diagonal elements in $A_{22}+\Delta$ are more negative than those of A_{22}. This implies with $(A_{22} + \Delta)e = 0$ that the diagonal elements must be nonnegative. From $\pi > 0$ it follows that $A_{22}+\Delta$ must be irreducible, for otherwise a component of $\bar{\pi}_2$ would be zero. According to Corollary 1 in §3.5 of Varga (1962), the signs of the matrix elements and the irreducibility imply that every principal submatrix of $A_{22} + \Delta$ is an M–matrix. Therefore, $A_{22} + \Delta$ is a singular M–matrix. □

Now we can prove the following theorem that says that in a nearly transient chain, the size of the π_i in the jth block is controlled by the smallness of the preceding off-diagonal columns $1, \ldots, j-1$, and by the condition of a perturbed jth diagonal block. The size of this perturbation is again determined by the smallness of the off-diagonal columns $1, \ldots, j-1$. This implies that the trailing solution components tend to be larger than the leading ones.

THEOREM 7.1. *If A is partitioned as indicated in* (7.1), *then* $\|\bar{\pi}_1^T\| \leq \epsilon_1\kappa_1$ *with* $\kappa_1 = \|A_{11}^{-1}\|$. *Furthermore, there exist matrices* $X_{j+1,j+1}$ *such that*

$$\|A_{j+1,j+1} - X_{j+1,j+1}\| \leq \epsilon_1 + \cdots + \epsilon_j, \quad 1 \leq j \leq k-2,$$

and

$$\|\bar{\pi}_{j+1}^T\| \leq (\epsilon_1 + \cdots + \epsilon_{j+1})\kappa_{j+1}, \quad \textit{where} \quad \kappa_{j+1} = \|X_{j+1}^{-1}\|.$$

Proof. The statements for $\bar{\pi}_1$ follow from the 2×2 block partitioning. Now apply the same argument recursively to the matrix

$$\bar{A}_{22} = \begin{pmatrix} A_{22} & * & \cdots & * & * \\ F_{32} & A_{33} & \cdots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & A_{k-1,k-1} & * \\ F_{k2} & F_{k3} & \cdots & F_{k,k-1} & A_{kk} \end{pmatrix} + \Delta,$$

where Δ is given by Lemma 7.1. For instance, X_2 is the leading diagonal block of $\bar{A}_{22}$ with $\|X_2^{-1}\| = \kappa_2$. Lemma 7.1 insures $\|\Delta\| \leq \epsilon_1$ and $\|A_{22} - X_2\| \leq \Delta \leq \epsilon_1$. Since the norm of the first off-diagonal column is bounded above by $\epsilon_1 + \epsilon_2$, Lemma 7.1 gives $\|\bar{\pi}_2^T\| \leq (\epsilon_1 + \epsilon_2)\kappa_2$. ∎

8. Concluding remarks. Our goal was to better understand how individual stationary probabilities are affected by unstructured perturbations to the transition probabilities. Consequently, we measured all perturbations relative to 1 rather than relative to $A = I - P$ or relative to the structure of P. In other words, we measured the magnitude of a perturbation by $\|E\|/\|P\| = \|E\|$ instead of $\|E\|/\|A\|$ or $\max_{ij} |e_{ij}|/p_{ij}$. The latter two measures result in significantly different interpretations of sensitivity. For example, perturbations that are small relative to 1 can greatly affect the stationary probabilities of

$$P = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}, \qquad \epsilon << 1,$$

but measured relative to $\|A\| = 2\epsilon$, or measured by $\max_{ij} |e_{ij}|/p_{ij}$, small perturbations cannot have a drastic effect (Meyer (1980), O'Cinneide (1993), and Zhang (1993)).

REFERENCES

J. L. BARLOW (1986), *On the smallest positive singular value of a singular M–matrix with applications to ergodic Markov chains*, SIAM J. Algebraic Discrete Meth., 7, pp. 414–424.

——— (1993), *Error bounds for the computation of null vectors with applications to Markov chains*, SIAM J. Matrix Anal. Appl., 14, pp. 598–618.

A. BERMAN AND R. J. PLEMMONS (1979), *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York.

S. L. CAMPBELL AND C. D. MEYER (1991), *Generalized Inverses of Linear Transformations*, Dover Publications (1979 edition by Pitman Pub. Ltd., London), New York.

S. CHANDRASEKARAN AND I. IPSEN (1992), *On the sensitivity of solution components in linear systems of equations*, Computer Science Technical Report, Yale University, New Haven, CT; SIAM J. Matrix Anal. Appl., to appear.

R. E. FUNDERLIC AND J. MANKIN (1981), *Solution of homogeneous systems of linear equations arising from compartmental models*, SIAM J. Sci. Statist. Comput., 2, pp. 375–383.

R. E. FUNDERLIC AND C. D. MEYER (1986), *Sensitivity of the stationary distribution vector for an ergodic Markov chain*, Linear Algebra Appl., 76, pp. 1–17.

G. H. GOLUB AND C. D. MEYER (1986), *Using the QR factorization and group inversion to compute, differentiate, and estimate the sensitivity of stationary probabilities for Markov chains*, SIAM J. Algebraic Discrete Meth., 7, pp. 273–281.

G. H. GOLUB AND C. F. VAN LOAN (1989), *Matrix Computations, Second Ed.*, The Johns Hopkins Press, Baltimore.

W. K. GRASSMANN, M. I. TAKSAR, AND D. P. HEYMAN (1985), *Regenerative analysis and steady state distributions for Markov chains*, Oper. Res., 33, pp. 1107–1116.

W. J. HARROD AND R. J. PLEMMONS (1984), *Comparison of some direct methods for computing stationary distributions of Markov chains*, SIAM J. Sci. Statist. Comput., 5, pp. 453–469.

J. G. KEMENY AND J. L. SNELL (1960), *Finite Markov Chains*, D. Van Nostrand, New York.

C. D. MEYER (1975), *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Rev., 17, pp. 443–464.

——— (1982), *Analysis of finite Markov chains by group inversion techniques*, in Recent Applications of Generalized Inverses, S. L. Campbell, ed., Research Notes in Mathematics, Vol. 66, Pitman, Boston, pp. 50–81.

——— (1980), *The condition of a finite Markov chain and perturbation bounds for the limiting probabilities*, SIAM J. Algebraic Discrete Meth., 1, pp. 273–283.

——— (1994), *Sensitivity of Markov chains*, SIAM J. Matrix Anal. Appl., 15, pp. 715–728.

C. D. MEYER AND G. W. STEWART (1988), *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25, pp. 679–691.

C. A. O'CINNEIDE (1993), *Entrywise perturbation theory and error analysis for Markov chains*, Numer. Math., 65, pp. 109–120.

C. PAIGE, G. STYAN, AND P. WACHTER (1975), *Computation of the stationary distribution of a Markov chain*, J. Statist. Comput. Simul., 4, pp. 173–186.

P. J. SCHWEITZER (1968), *Perturbation theory and finite Markov chains*, J. Appl. Probab., 5, pp. 401–413.

E. SENETA (1991), *Sensitivity analysis, ergodicity coefficients, and rank-one updates for finite Markov chains*, in Numerical Solution of Markov Chains, W. J. Stewart, ed., Probability: Pure and Applied, No. 8, Marcel Dekker, New York, pp. 121–129.

G. W. STEWART (1991), *On the sensitivity of nearly uncoupled Markov chains*, in Numerical Solution of Markov Chains, W. J. Stewart, ed., Probability: Pure and Applied, No. 8, Marcel Dekker, New York, pp. 105–119.

——— (1992b), *On the perturbation of Markov chains with nearly transient states*, Computer Science Technical Report, University of Maryland, College Park, MD.

G. W. STEWART AND G. ZHANG (1991), *On a direct method for the solution of nearly uncoupled Markov chains*, Numer. Math., 59, pp. 1–11.

R. VARGA (1962), *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ.

G. ZHANG (1993), *On the sensitivity of the solution of nearly uncoupled Markov chains*, SIAM J. Matrix Anal. Appl., 14, pp. 1112–1123.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1075–1091, October 1994

003

AN EFFICIENT ALGORITHM TO COMPUTE ROW AND COLUMN COUNTS FOR SPARSE CHOLESKY FACTORIZATION*

JOHN R. GILBERT†, ESMOND G. NG‡, AND BARRY W. PEYTON‡

Abstract. Let an undirected graph G be given, along with a specified depth-first spanning tree T. Almost-linear-time algorithms are given to solve the following two problems. First, for every vertex v, compute the number of descendants w of v for which some descendant of w is adjacent (in G) to v. Second, for every vertex v, compute the number of ancestors of v that are adjacent (in G) to at least one descendant of v.

These problems arise in Cholesky and QR factorizations of sparse matrices. The authors' algorithms can be used to determine the number of nonzero entries in each row and column of the triangular factor of a matrix from the zero/nonzero structure of the matrix. Such a prediction makes storage allocation for sparse matrix factorizations more efficient. The authors' algorithms run in time linear in the size of the input times a slowly growing inverse of Ackermann's function. The best previously known algorithms for these problems ran in time linear in the sum of the nonzero counts, which is usually much larger. Experimental results are given demonstrating the practical efficiency of the new algorithms.

Key words. sparse Cholesky factorization, sparse QR factorization, symbolic factorization, graph algorithms, chordal graph completion, disjoint set union, column counts, row counts

AMS subject classifications. 65F50, 68Q20

1. Introduction.

Direct solution of a sparse symmetric positive definite linear system requires four steps [7], [15]: reordering, symbolic factorization, sparse Cholesky factorization, and sparse triangular solutions. Let A be the $n \times n$ coefficient matrix of the linear system after it has been reordered to reduce fill, and let L be the lower triangular Cholesky factor of A. This paper presents improved algorithms for computing the number of nonzero entries in each row and column of L *prior to the symbolic factorization step*. We refer to these parameters as the *row counts* and *column counts* of L.

In least squares computations, A is $m \times n$, with $m \geq n$. It is often necessary to compute the orthogonal factorization $A = QR$. Our algorithms can be used also to predict upper bounds on the row counts and column counts of the upper triangular factor R, since the structure of R is always contained in the structure of the Cholesky factor of A^TA [12].

Throughout the paper we assume familiarity with graphs, trees, and such basic techniques as depth-first search [24]. We also assume a basic knowledge of the four steps in solving sparse systems by Cholesky factorization, and with the use of graphs in these algorithms [15]. More specifically, we assume familiarity with elimination trees [19], skeleton graphs [18], postorderings, supernodes [1], [2], [16], [20], [21], and the subscript compression scheme for L [15], [25].

* Received by the editors September 14, 1992; accepted for publication (in revised form) May 14, 1993. This work was supported in part by the Applied Mathematical Sciences Research Program, Office of Energy Research, United States Department of Energy contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Incorporated, and in part by the Institute for Mathematics and Its Applications with funds provided by the National Science Foundation.

† Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304-1314 (gilbert@parc.xerox.com).

‡ Mathematical Sciences Section, Oak Ridge National Laboratory, P. O. Box 2008, Oak Ridge, Tennessee 37831-6367 (esmond@msr.epm.ornl.gov, peyton@msr.epm.ornl.gov).

1.1. Applications. Here we survey some of the sparse matrix settings in which it is useful to precompute the row counts, the column counts, or the total number of nonzeros in the Cholesky factor of a sparse matrix.

Either the row or column counts can be used to compute $|L|$, the total number of nonzeros in the factor. (We write $|X|$ for the number of nonzeros in a matrix X or the number of elements in a set X.) Knowing $|L|$ before the numeric factorization step makes it possible to allocate storage all at once instead of dynamically. In sparse Cholesky factorization, the time required to compute $|L|$ by existing methods is dominated by the time required for numerical factorization; but there are at least two settings in which it is valuable to be able to compute $|L|$ as fast as possible.

First, some methods for large-scale numerical optimization use Cholesky factorization on a Hessian matrix [5], [6]. If the Hessian is indefinite, Cholesky factorization will abort, but the partial factorization contains enough information to help determine a good descent direction containing negative curvature information. In this case, the symbolic factorization time may dominate the time spent on the numeric factorization before it aborts. Thus it may be more efficient to skip the symbolic phase and to build the data structure for L during the numeric factorization. However, for this to be efficient, we still need to find $|L|$ (and perhaps the column counts) before starting the factorization.

Second, much research remains to be done on the issue of how best to reorder the initial matrix to reduce fill, i.e., to reduce $|L|$. It is sometimes useful to compute $|L|$ for many different orderings of the same matrix, both in experiments with reordering algorithms and when trying to optimize an ordering for a specific matrix. Our new algorithms make this much faster.

Besides fill, there are several other measures of the quality of a reordering. Some of them can be computed from the column counts; for example, the total number of arithmetic operations is the sum of the squares of the column counts, and the maximum front size is equal to the largest column count. The smallest maximum front size, over all reorderings of a graph, is one more than the graph's *treewidth* [3]. Thus the fast column count algorithm may also be useful in experimental studies of treewidth.

Two applications related to the supernodal structure of L also require the column counts. Supernodes are clusters of columns with related nonzero patterns, which can be exploited to use fast dense matrix computation kernels in sparse factorization; §3 describes them in more detail. First, there is a simple, flexible $O(n)$ scheme for computing supernode partitions [2], [17] that takes the column counts and the elimination tree as input. This algorithm is more versatile and faster than the $O(|A|)$ algorithm of Liu, Ng, and Peyton [20], which takes the original matrix and its elimination tree as input. The latter algorithm computes the so-called fundamental supernode partition. Given a fast algorithm to compute column counts, the more flexible scheme could be used efficiently to compute coarser supernode partitions [2], which trade extra fill for a simpler sparsity structure that can be used to improve efficiency on vector supercomputers or to reduce synchronization overhead on shared-memory multiprocessors.

The second supernodal application of the column counts is to compute the storage required for indexing information for L in the usual compressed format generated by the symbolic factorization step [25]. Current software packages [4], [9] do not precompute the space needed for this compressed symbolic factorization, because it is too expensive using the currently known algorithms. The storage required for the other three steps in the solution process is usually computed in advance; we believe

that the new algorithms introduced here are efficient enough to be used by a software package to precompute the storage requirement of the symbolic factorization step as well.

Finally, we know of only one application that specifically requires the row counts rather than the column counts. The row counts are the numbers of column modifications (sparse SAXPYs) required to complete each column in sparse Cholesky factorization algorithms. Some parallel implementations [13], [14] need the row counts to determine when all the modifications have arrived for each column.

1.2. Previous work. Like many combinatorial algorithms in sparse matrix factorization, all the efficient algorithms for row and column counts begin by computing the elimination tree of the matrix (defined in the next section). The fastest known elimination tree algorithm is due to Liu [19]. The time complexity for this algorithm is dominated by disjoint set union operations, which take time $O(m\,\alpha(m,n))$, where A is $n \times n$ and has $2m$ off-diagonal nonzeros. Here $\alpha(m,n)$ is a slowly growing inverse of Ackermann's function defined by Tarjan [27]; for all values of m and n less than the number of elementary particles in the observable universe, $\alpha(m,n) \leq 4$. Thus a function that is $O(m\,\alpha(m,n))$ is often called "almost linear."

The fastest previously known algorithm for computing row and column counts is also due to Liu [19]. It first computes the elimination tree of A and then traverses each "row subtree" of the elimination tree (defined in the next section). The total size of the row subtrees is the number of nonzeros in the factor, so the running time of this step is $O(|L|)$. Unless the factor is extremely sparse, the subtree traversals dominate the time to find the elimination tree. To put this in perspective, suppose A is the matrix of an n-node finite difference mesh ordered by nested dissection. Then m is $O(n)$, and $|L|$ is $O(n \log n)$ in two dimensions or $O(n^{4/3})$ in three dimensions.

The algorithm in this paper also takes A and the elimination tree as input but runs in almost-linear time $O(m\,\alpha(m,n))$; the time complexity for the new algorithm is dominated by disjoint set union operations. Thus it computes the row and column counts in the same asymptotic time needed to find the elimination tree. As we will see in §4, this asymptotic efficiency is also reflected in practice.

1.3. Outline. Section 2 presents the row and column count algorithm from a graph-theoretic point of view. Here it is convenient to think of the input not as the graph $G(A)$ of a matrix, but as the graph $G(A) \cup T(A)$ that has edges both for the matrix nonzeros and for the elimination tree. (The elimination tree $T(A)$ usually has edges not contained in $G(A)$.) The elimination tree is a depth-first spanning tree of the graph $G(A) \cup T(A)$; thus for the purpose of the high-level view in §2, the input is just an undirected graph with a specified depth-first spanning tree. In this setting, we suspect that our results may be useful in efficient algorithms involving chordal graphs, chordal completion, and treewidth.

In §3 we return to the matrix-computation point of view, and discuss details of the implementation in the sparse matrix setting. Two points of practical importance arise here: we modify the algorithm slightly to make only one pass over its input, and we take advantage of supernodal structure to compute only with a subgraph called the *skeleton graph*. We show how to organize the entire computation, including the skeleton graph reduction, within the framework of the fundamental supernode algorithm of Liu, Ng, and Peyton [20].

Section 4 contains experimental results. We experiment with both the nodal and supernodal versions of the algorithm, as well as with several implementations of the disjoint set union operations (UNION and FIND) that dominate the asymptotic

running time. The best version is the supernodal algorithm with path-halving and no union by rank (definitions are in §3.3); it performs well enough that we argue it should be a standard part of high-performance sparse factorization codes. Finally, §5 contains concluding remarks.

2. The algorithm.

2.1. Definitions and problem statement. Let $G = (V, E)$ be a connected undirected graph with n vertices and m edges, and let T be a specific depth-first spanning tree for G (e.g., $G = G(A) \cup T(A)$ and $T = T(A)$). We call vertices v and w *adjacent* if they are joined by an edge in G; that is, if $(v, w) \in E$. We say that vertex v is an *ancestor* of vertex w if v is on the path in T from w to the root of T. Vertex v is a *descendant* of w if w is an ancestor of v. Note that a vertex is its own ancestor and its own descendant; a *proper* ancestor or descendant is one that is different from the vertex itself. We write $T[v]$ for the set of descendants of v and also for the subtree of T (rooted at v) that those vertices induce.

Since T is a depth-first spanning tree, every edge of G (whether or not it is an edge of T) joins an ancestor in T to a descendant in T [24].

To simplify notation, we assume that the vertices of G are the integers 1 through n. We also assume that the vertex numbers are a *postorder* on T; that is, that for every vertex v, the vertices of $T[v]$ are numbered consecutively, with v numbered last. Thus vertex n is the root of T.

The *level* of vertex v, which we write *level*(v), is its distance in T from the root. The *least common ancestor* of vertices v and w, which we write $lca(v, w)$, is the ancestor of v and w with the smallest postorder number (or the largest level). Both a postorder numbering and the vertex levels for an arbitrary tree can be computed in linear time by depth-first search [26]. Given a set of k pairs $\{v, w\}$ of vertices, the k least common ancestors $lca(v, w)$ can be computed in $O(k\,\alpha(k, n))$ time, where α is the very slowly growing inverse of Ackermann's function mentioned above [28]. We describe these algorithms in more detail in §3.

We consider the following two problems.

Problem 1 (row counts). For every node $u \in V$, let *row*$[u]$ be the set of descendants v of u for which either $v = u$ or there exists an edge (u, w) with $w \in T[v]$. The problem is to compute $rc(u) = |row[u]|$ for every u.

Problem 2 (column counts). For every node $v \in V$, let *col*$[v]$ be the set of ancestors u of v for which either $u = v$ or there exists an edge (u, w) with $w \in T[v]$. The problem is to compute $cc(v) = |col[v]|$ for every v.

Note that $v \in row[u]$ if and only if $u \in col[v]$, and that u is an element of both *row*$[u]$ and *col*$[u]$. For each u, the subgraph of T induced by *row*$[u]$, denoted by $T_r[u]$ and referred to as the *row subtree* of u, is connected; it is a "pruned subtree" rooted at u. The subgraph of T induced by *col*$[v]$ may not be connected.

We conclude by briefly describing the relationship between these problems and sparse Cholesky factorization. It may seem a bit confusing that we include the elimination tree edges in the graph G in the graph problem but not in the matrix problem; however, the answer is the same in either case.

Let an $n \times n$ symmetric, positive definite matrix A be given, and let $G(A)$ be its undirected graph (whose vertices are the integers 1 through n). Let $G^+(A)$ be the *filled graph* of $G(A)$ [22] obtained by adding to $G(A)$ edge (v, w) whenever there is a path in $G(A)$ from v to w whose intermediate vertices are all smaller than both v and w. The graph $G^+(A)$ is chordal, and (ignoring numerical cancellation) is the graph of $L + L^T$, where L is the Cholesky factor of A [23].

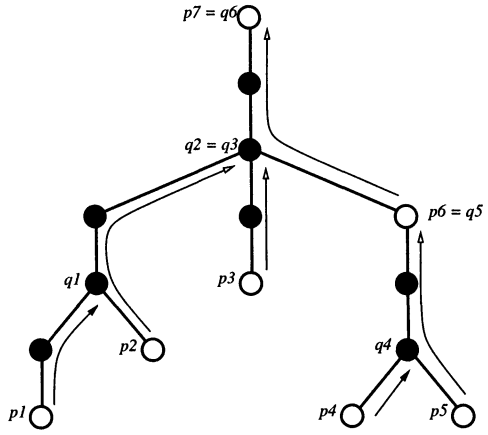


FIG. 1. *Example of path decomposition.*

The *elimination tree* of A, denoted $T(A)$, has vertices 1 through n, and the parent of vertex v is the smallest $w > v$ such that (v, w) is an edge of $G^+(A)$. Liu [19] surveys the uses and properties of this structure. It is a forest with one tree for each connected component of $G(A)$; if A is irreducible then $T(A)$ is a tree. The elimination tree may not be a subgraph of $G(A)$, but it is a subgraph of $G^+(A)$, and in fact it is a depth-first spanning tree of that graph. If A' is a matrix whose graph is $G(A') = G(A) \cup T(A)$, it is straightforward that $G^+(A') = G^+(A)$ and $T(A') = T(A)$.

Now consider problems (1) and (2) above for $G = G(A')$ and $T = T(A')$. It is easy to show [19] that the edges of $G^+(A) = G^+(A')$ are exactly those (u, v) for which $v \neq u$ and $v \in row[u]$ (or $u \in col[v]$). Thus $rc(u)$ is the number of nonzeros in row u of the Cholesky factor L of A, and $cc(v)$ is the number of nonzeros in column v of L.

2.2. Row counts. We count the vertices in $row[u]$ by counting the edges in the pruned subtree $T_r[u]$ of T that $row[u]$ induces. The following lemma lets us partition those edges into paths.

LEMMA 2.1. *Let $p_1 < p_2 < \cdots < p_k$ be some of the vertices of a rooted tree R (where $<$ is postorder), and suppose all the leaves and the root of R are among the p_i's. Let q_i be the least common ancestor of p_i and p_{i+1} for $1 \leq i < k$. Then each edge (s, t) of the tree is on the tree path from p_j to q_j for exactly one j.*

Proof. Suppose t is the parent of s in R. The descendants of s include at least one leaf, so they include at least one p_i. Let p_j be the largest p_i among the descendants of s. Then $p_j \leq s < p_{j+1}$. (There must be a p_{j+1}—that is, we cannot have $j = k$—because p_k is the root, which is a proper ancestor of s.) Since s is an ancestor of p_j but not of p_{j+1}, the least common ancestor q_j of p_j and p_{j+1} is a proper ancestor of s, and hence an ancestor of t. Therefore (s, t) is on the path from p_j to q_j.

Now consider an $i \neq j$. If s is not an ancestor of p_i, then (s, t) is not on the path from p_i to its ancestor q_i. If s is an ancestor of p_i, then $p_i \leq s$, and $i \neq j$ implies $p_i \leq p_{i+1} \leq s$. Since postorder assigns consecutive numbers to the vertices in a subtree, this means that s is also an ancestor of p_{i+1}, and hence of the least common ancestor q_i. Thus (s, t) is not on the path from p_i to q_i. □

Figure 1 shows an example of the path decomposition.

Recall that T is a depth-first spanning tree of G and hence every edge of G joins an ancestor in T to a descendant in T. Now consider a vertex u of G. If the lower-numbered neighbors of u in G are $p_1 < p_2 < \cdots < p_{k-1}$, and if $p_k = u$, then the pruned subtree $R = T_r[u]$ induced by $row[u]$ satisfies the hypotheses of Lemma 2.1. Thus the number of edges in $T_r[u]$ is the sum of the lengths of the paths in the lemma. The length of the path from p_i to its ancestor q_i is the difference of their levels. The number of vertices in $row[u]$ is one more than the number of edges, so

$$rc(u) = 1 + \sum_{1 \le i < k} \Big(level(p_i) - level(lca(p_i, p_{i+1})) \Big).$$

(Here *lca* and *level* are taken in T rather than $T_r[u]$, but it is clear that for any two vertices in $row[u]$ the least common ancestor and the difference in levels are the same in either tree.)

Let $ladj[u]$ be the lower numbered neighbors of u in G. The algorithm to compute $rc(u)$ for all u first sorts each set $ladj[u] \cup \{u\}$ by postorder, then computes all the necessary least common ancestors, and finally computes the sum above for each u. Computing level numbers (and the postorder itself if necessary) takes linear time, and sorting the sets $ladj[u] \cup \{u\}$ into postorder takes linear time by a lexicographic bucket sort. There is one least-common-ancestor computation for each edge of G, so the dominant term in the algorithm's time complexity is $O(m\,\alpha(m, n))$.

2.3. Column counts. Because $u \in col[v]$ if and only if $v \in row[u]$, the column count $cc(v)$ is equal to the number of row subtrees $T_r[u]$ that contain v. We could compute $cc(v)$ by traversing each row subtree in turn, and counting the number of times each vertex was traversed [19]. This, however, would take time proportional to $\sum_v cc(v)$.

To get a faster algorithm, we define weights $wt(v)$ on the vertices of G in such a way that the column count for vertex v turns out to be the sum of the weights of the descendants of v. The key observation is that we can compute these weights as a sum of contributions from each row subtree, and that the row subtree contributions can be computed efficiently using the same least common ancestors as in the row count algorithm.

Here are the details. For each vertex u, define χ_u to be the characteristic function of $row[u]$, so that $\chi_u(v) = 1$ if $v \in row[u]$ and $\chi_u(v) = 0$ otherwise. Define wt_u by

$$wt_u(v) = \chi_u(v) - \sum_{\text{children } y \text{ of } v} \chi_u(y). \tag{1}$$

These weights may be positive, negative, or zero. This definition implies that

$$\chi_u(v) = \sum_{x \in T[v]} wt_u(x). \tag{2}$$

In a sense, wt_u is a "first difference" down the tree of the characteristic function of $row[u]$. Finally, define

$$wt(v) = \sum_{u \in V} wt_u(v). \tag{3}$$

Now we prove three lemmas relating the column counts to the weights, the weights to the sets $row[u]$, and finally the $row[u]$, once more, to the least common ancestors.

LEMMA 2.2. *For every vertex v,*

$$cc(v) = \sum_{x \in T[v]} wt(x).$$

Proof. Because $v \in row[u]$ if and only if $u \in col[v]$, we have

$$cc(v) = |col[v]| = \sum_{u \in V} \chi_u(v).$$

Equation (2) states that this is equal to

$$\sum_{u \in V} \sum_{x \in T[v]} wt_u(x).$$

The result follows by reversing the order of summation and using (3). □

Lemma 2.2 implies that we can compute the column counts easily and efficiently from the weights by traversing the tree in postorder and summing the weights of the subtrees. It remains to describe how to compute the weights.

LEMMA 2.3. *Let u and v be vertices. Suppose that d of the children of v are vertices of $row[u]$. Then*

$$wt_u(v) = \begin{cases} 1 - d & \text{if } v \in row[u], \\ -1 & \text{if } v \text{ is the parent of } u, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. This is immediate from (1) and the definition of χ_u. □

Lemma 2.3 implies that the only vertices v for which $wt_u(v)$ is nonzero are the leaves of the pruned row subtree $T_r[u]$, the internal vertices of $T_r[u]$ that have more than one child in $T_r[u]$, and the parent of u. The following lemma allows us to compute $wt_u(v)$ for each v from the same p_i's and q_i's we used in the row count algorithm.

LEMMA 2.4. *Let $p_1 < p_2 < \cdots < p_k$ be some of the vertices of a rooted tree R (where $<$ is postorder), and suppose all the leaves and the root of R are among the p_i's. Let q_i be the least common ancestor of p_i and p_{i+1}, for $1 \le i < k$. Then for each vertex v of R, the number of children of v in R is*

$$|\{i : q_i = v\}| - |\{i : p_i = v\}| + 1.$$

Proof. Let $Q = |\{i : q_i = v\}|$, let $P = |\{i : p_i = v\}|$, and let d be the number of children of v in R. Consider the set of directed paths from p_i to q_i in R, for $1 \le i < k$. For any collection of directed paths, each path that includes vertex v either *begins* at v or *enters* v along edges from other vertices. Similarly, each path that includes vertex v either *ends* at v or *leaves* v along edges to other vertices. Consequently:

> The number of paths that either begin at v or enter v along edges from other vertices must be equal to the number of paths that either end at v or leave v along edges to other vertices.

(This is essentially Kirchoff's law for a flow of unit size from p_i to q_i for each i.) Lemma 2.1 says that every edge of R is on exactly one of these paths. Therefore one path enters v from each of the d children of v; exactly one path leaves v, to its parent, unless v is the root; one path begins at v for each i such that $p_i = v$ (except for $i = k$

```
Sort the vertices and their lists of neighbors by a postorder of T;
Compute level(u) as the distance from u to n (the root), for all u;
Compute lca(p, p') for every p and its successor p' in ladj[u] ∪ {u}, for all u;
rc(u) ← 1, for all u;
wt(u) ← 1, for all u;
for u ← 1 to n do
    if u ≠ n then
        wt(parent(u)) ← wt(parent(u)) − 1;
    end if
    for p ∈ ladj[u] (in order) do
        wt(p) ← wt(p) + 1;
        p' ← the successor of p in ladj[u] ∪ {u};
        q ← lca(p, p');
        rc(u) ← rc(u) + level(p) − level(q);
        wt(q) ← wt(q) − 1;
    end for
end for
cc(v) ← wt(v), for all v;
for v ← 1 to n − 1 do
    cc(parent(v)) ← cc(parent(v)) + cc(v);
end for
```

FIG. 2. *Algorithm to compute row and column counts.*

if v is the root); and one path ends at v for each i such that $q_i = v$. A trivial path with $p_i = q_i = v$ both starts and ends at v, but does not enter or leave v. Thus the relation above is

$$P + d = Q + 1$$

if v is not the root of R, or

$$(P - 1) + d = Q + 0$$

if v is the root. In either case, we have $d = Q - P + 1$ as desired. □

Now consider a vertex u of G. If the vertices of $ladj[u]$ are $p_1 < p_2 < \cdots < p_{k-1}$, and if $p_k = u$, then the pruned subtree $R = T_r[u]$ induced by $row[u]$ satisfies the hypotheses of Lemma 2.4. Therefore, using Lemma 2.3, if v is a vertex of $row[u]$ then $wt_u(v) = |\{i : p_i = v\}| - |\{i : q_i = v\}|$. Thus we could compute $wt_u(v)$ for all v by initializing each weight to zero, setting the weight of the parent of u to -1, and then adding one to the weight of each p_i and subtracting one from the weight of each q_i.

In fact we do not need to compute $wt_u(v)$ separately for each u; we can compute $wt(v) = \sum_u wt_u(v)$ all at once. The algorithm begins, like the row count algorithm, by sorting each set $ladj[u] \cup \{u\}$ in postorder and computing all the necessary least common ancestors. It initializes $wt(u)$ to one for each u. Then, for each u, it subtracts one from the weight of the parent of u, adds one to $wt(p)$ for each $p \in ladj[u]$, and subtracts one from $wt(q)$ for the least common ancestor q of each pair p and p' of consecutive members (in postorder) of $ladj[u] \cup \{u\}$. Finally, the algorithm computes $cc(v)$ for all v by summing the weights of each subtree in postorder. Figure 2 sketches the algorithm to compute both row and column counts. The only step

that takes more than linear time is the least common ancestor computation, and the dominant term in the algorithm's time complexity is $O(m\,\alpha(m,n))$.

3. Implementation. The discussion in the previous section was in a general graph-theoretic setting. However, to obtain the most efficient implementation of the new algorithm for our applications, we need to switch back to a sparse matrix setting.

Consider a symmetric matrix A and its graph $G(A)$. Assume that the elimination tree $T(A)$, the postordering, and the values *level*(u) (with respect to $T(A)$) have been computed, as required in Fig. 2. Two other requirements must be met to obtain a practical and efficient implementation of the new algorithm.

First, we must reorganize the computation to avoid sorting the adjacency lists by postorder and precomputing all the least common ancestors. Indeed, direct implementation of the algorithm in Fig. 2 would require that $G(A)$ be processed three times, and we doubt that any multiple-pass implementation will come close to realizing the practical efficiency of the single-pass implementation presented in this section.

Second, we must discard some edges of $G(A)$ that do not affect the result. Recall from Liu [18] that the *skeleton graph* $G^-(A)$ is obtained from $G(A)$ by removing every edge (u,v) for which $v<u$ and the vertex v is not a leaf of $T_r[u]$. The skeleton graph is the smallest subgraph of $G(A)$ whose filled graph is identical with that of $G(A)$. Consequently, the new algorithm produces the same results when applied to $G^-(A)$ as when applied to $G(A)$. Indeed, if $G=G^-(A)\cup T(A)$ rather than $G=G(A)\cup T(A)$ in Lemmas 2.1 and 2.4, then every vertex $p_1,p_2,\ldots,p_{k-1}$ is a leaf in the tree R. This reduces the number of edges searched and least common ancestors computed by the new algorithm to the minimum possible. Since $G^-(A)$ often has far fewer edges than $G(A)$ in practice, an implementation that processes $G^-(A)$ rather than $G(A)$ promises to be substantially faster; we see in §4 that this is indeed the case.

The skeleton graph $G^-(A)$ is closely related to fundamental supernodes of A, and can be computed efficiently in linear time by a simple modification of the algorithm of Liu, Ng, and Peyton [20] to find fundamental supernodes. Indeed, that algorithm is a good framework for implementing our new algorithm, whether the skeleton graph is exploited or not. We can combine the two algorithms to obtain an efficient single-pass implementation. As this implementation processes the edges of $G(A)$, it discards edges not in the skeleton graph, and uses only the skeleton edges to compute the data for the row and column counts. If m^- is the number of edges in $G^-(A)$, then this scheme runs in $O(m+m^-\,\alpha(m^-,n))$ time.

Section 3.1 below reviews the material we need from Liu, Ng, and Peyton [20]. Section 3.2 presents a detailed version of the new combined implementation. Section 3.3 briefly describes our implementation of the disjoint set union algorithm for computing the least common ancestors, upon which the time complexity of our algorithm depends.

3.1. A fast algorithm for finding supernodes. Liu, Ng, and Peyton [20] introduced an $O(|A|)$ algorithm to compute a fundamental supernode partition. Their algorithm assumes that the elimination tree $T(A)$ has been computed and that the vertices are numbered by a postordering of $T(A)$. Let the *higher adjacency set* of v, denoted by *hadj*$[v]$, be the set of neighbors of v in $G(A)$ that are numbered higher than v, and let *hadj*$^+[v]$ be the higher adjacency set of v in $G^+(A)$. Ashcraft and Grimes [2] defined a *fundamental supernode* as a maximal contiguous set of vertices $\{v,v+1,\ldots,v+s\}$ such that $v+i$ is the *only* child of $v+i+1$ in the elimination

tree (for $i = 0, 1, \ldots, s-1$) and

$$hadj^+[v] = hadj^+[v+s] \cup \{v+1, v+2, \ldots, v+s\}.$$

The fundamental supernodes partition the vertices of $G(A)$.

In matrix terms, a supernode is any group of consecutive columns in L with a full diagonal block and with identical column patterns below the diagonal block. A fundamental supernode is maximal subject to the following condition: every column of the supernode except the last is an only child in the elimination tree. Liu, Ng, and Peyton [20] give several reasons why fundamental supernodes are the most appropriate choice of supernodes for most applications, one of which is that they are independent of the choice of postordering for $T(A)$.

Finding the set of fundamental supernodes is equivalent to finding the first vertex of each supernode. These "first vertices" are characterized by the following result.

THEOREM 3.1 (Liu, Ng, and Peyton [20]). *Vertex v is the first vertex in a fundamental supernode if and only if vertex v has two or more children in the elimination tree, or v is a leaf of some row subtree of $T(A)$.*

The key observation is that the vertices required by the row/column count algorithm (the p_i's and q_i's) are in fact first vertices of fundamental supernodes. It follows from the discussion immediately after Lemma 2.3 in §2.3 that the vertex pairs p_i, p_{i+1} whose least common ancestors must be found can be restricted to vertices that are leaves of some row subtree of $T(A)$. This is equivalent to restricting the algorithm in Fig. 2 to the skeleton graph $G^-(A)$. Furthermore, when the p_i's are restricted in this manner, it is clear that every least common ancestor $q_i = lca(p_i, p_{i+1})$ has two or more children. Consequently, the Liu, Ng, and Peyton algorithm is an excellent vehicle for an efficient implementation of our new algorithm.

3.2. Detailed implementation of the new algorithm. The details of our single-pass, column-oriented implementation are given in Fig. 3. Note that it traverses the higher adjacency sets *hadj*[p] rather than the lower adjacency sets used by the algorithm in Fig. 2. Again, the vertices are numbered by a postorder of the tree $T(A)$, but here no assumption is made concerning the order of the vertices in *hadj*[p], nor are the least common ancestors computed in advance. Consequently, this implementation makes only a single pass through $G(A)$.

The vector of markers *prev_p*(u) stores the most recently visited vertex p' that is a leaf in $T_r[u]$. The pairs p, p' produced by the algorithm are precisely the multiset consisting of every consecutive pair of leaves in every row subtree $T_r[u]$. The reason for this is that one of the **if** tests in the algorithm screens out all edges in $G(A)$ except those in the skeleton graph $G^-(A)$. The lines marked with asterisks have been added to the algorithm solely for this purpose. Of these, the key line is the test for whether or not the first (i.e., lowest numbered) descendant of p (*fst_desc*(p)) is greater than the most recently visited vertex in *ladj*[u], namely the vertex stored in the marker variable *prev_nbr*(u). It is not difficult to verify that when the condition holds true, no descendant of p is adjacent to u in $G(A)$; hence p is indeed a leaf in $T_r[u]$. For full details of this test, see Liu, Ng, and Peyton [20].

The implementation is correct with or without the starred lines. We have implemented both versions: we call the one with the starred lines the *supernodal* version, and the one without these lines the *nodal* version.[1] We experiment with both versions of the algorithm in our tests in §4.

[1] In the nodal version, *prev_p*(u) functions precisely as *prev_nbr*(u) does in the supernodal version.

```
    Sort the vertices by a postorder of T(A);
    Compute level(u) as the distance from u to n (the root), for all u;
*   Compute fst_desc(u) as the first (least) descendant of u in T(A), for all u;
    prev_p(u) ← 0, for all u;
*   prev_nbr(u) ← 0, for all u;
    rc(u) ← 1, for all u;
    wt(u) ← 0, for all nonleaves u in T(A);
    wt(u) ← 1, for all leaves u in T(A);
    for p ← 1 to n do
        if p ≠ n then
            wt(parent(p)) ← wt(parent(p)) − 1;
        end if
        for u ∈ hadj[p] do
*           if fst_desc(p) > prev_nbr(u) then
                wt(p) ← wt(p) + 1;
                p' ← prev_p(u);
                if p' = 0 then
                    rc(u) ← rc(u) + level(p) − level(u);
                else
                    q ← FIND(p');
                    rc(u) ← rc(u) + level(p) − level(q);
                    wt(q) ← wt(q) − 1;
                end if
                prev_p(u) ← p;
*           end if
*           prev_nbr(u) ← p;
        end for
        UNION(p, parent(p));
    end for
    cc(v) ← wt(v), for all v;
    for v ← 1 to n − 1 do
        cc(parent(v)) ← cc(parent(v)) + cc(v);
    end for
```

FIG. 3. *Implementation of algorithm to compute row and column counts.*

3.3. Disjoint set union. To compute least common ancestors, the algorithm in Fig. 3 must manipulate disjoint sets of vertices, each of which induces a subtree of the elimination tree. The highest numbered vertex in each set (the root of the subtree) is used to "name" the set, and is called the *representative vertex* of the set. Initially each vertex p from 1 to n is a singleton set. As the algorithm proceeds, it executes a sequence of FIND and UNION operations which are defined as follows.

> FIND(p): return the representative vertex of the unique set that contains p.
> UNION(u, v): combine the two distinct sets represented by u and v into a single set, which will be represented by the larger of u and v.

It is not hard to show that the call to FIND(p') in our algorithm returns $lca(p', p)$; see Tarjan [28] for details.

Each disjoint set is implemented as a tree stored using a parent vector (not to be confused with the *parent* vector in the elimination tree). The operation UNION(u, v) joins the two distinct trees represented by u and v together by making one of the roots a child of the other root. Consequently, UNION is a constant-time operation. This is not the case for FIND. The operation FIND(p) traces the *find path* from p to the root of p's tree. This root either is the representative vertex or contains a pointer to the representative vertex, depending on the implementation of UNION.

Tarjan [29] describes several techniques to shorten the find paths and thus reduce the amount of work spent on the FIND operations. *Union by rank* makes the shorter tree's root a child of the taller tree's root in UNION, which tends to keep the trees short and bushy. With no other enhancements, union by rank ensures that find paths are no longer than $O(\log_2(n))$. This is usually combined with one of two techniques for shortening the find path during a FIND operation. The first of these is *path compression*, which, after finding the root, makes the parent for each vertex on the find path point to the root during a second pass along the path. Alternatively, *path halving* resets the parent pointer for every other vertex on the find path to point to its grandparent. Path compression shortens the find path more, but requires two passes over the find path; path halving needs only one pass.

Tarjan [27], [29] showed that when union by rank is combined with either path compression or path halving, any sequence of n UNION's and m FIND's takes only $O(m\,\alpha(m, n))$ time. Tarjan [28] pointed out how to use the disjoint set union algorithm to find the least common ancestors of an arbitrary set of pairs of vertices from the same tree; our implementation of the row and column count algorithm uses the same method. Consequently, we can implement the nodal version of our algorithm to run in $O(m\,\alpha(m, n))$ time, and similarly we can implement the supernodal version to run in $O(m + m^-\,\alpha(m^-, n))$ time.

Gabow and Tarjan [10] showed that if the order of the UNION operations is known in advance (as is the case in our problem), then disjoint set union can be implemented so that a sequence of n UNION's and m ($\geq n$) FIND's takes only $O(m)$ time. Their sophisticated hybrid algorithm partitions the vertices into *microsets* and performs all the operations in a hierarchical fashion, using table look-up to answer queries within the microsets, and using the standard disjoint set union algorithm on the microsets themselves. We did not implement this algorithm; we believe its increased overhead would wipe out the difference between $O(m\,\alpha(m, n))$ and $O(m)$ in our application.

We implemented and tested the following six combinations.

1. No union by rank, no path compression or halving.
2. No union by rank, path compression.
3. No union by rank, path halving.
4. Union by rank, no path compression or halving.
5. Union by rank, path compression.
6. Union by rank, path halving.

We found surprisingly little difference in performance among the various options. Far more important is whether or not the row/column count processing is limited to the skeleton graph, as we see in the next section. We found that any gains due to union by rank were more than offset by the additional overhead required for its implementation. The third option—no union by rank, path halving—performed slightly better on most machines we tried. Path halving was clearly superior to path compression when the skeleton adjacency structure was not exploited. Consequently, we recommend path halving to those implementing the method, and in the next section all our timings

were obtained using path halving and no union by rank.

4. Experimental results. We ran the new algorithms on several problems from the Harwell–Boeing sparse matrix collection [8]. Table 1 lists our test problems, and Table 2 contains the problem statistics that have a bearing on the observed performance of our algorithms. Throughout this section `supcnt` refers to the "supernodal" version of the algorithm (Fig. 3 with the starred lines), which identifies the edges of the skeleton graph $G^-(A)$ and uses only those edges in its row and column count calculations, and `nodcnt` refers to the "nodal" version of the algorithm (Fig. 3 without the starred lines), which uses all the edges of $G(A)$.

4.1. Performance of the disjoint set union options. The primary purpose of Table 3 is to explain two things we observed in our tests: (i) why exploiting the skeleton graph is so beneficial and (ii) why the various disjoint set union (DSU) implementation options have so little influence on the performance of our code. The number of FIND operations required by `nodcnt` and `supcnt` is bounded above by m and m^-, respectively, and bounded below by $m - n$ and $m^- - n$. Thus, the huge difference between the number of FIND's required by `nodcnt` and the number of FIND's required by `supcnt` (see Table 3) simply reflects the fact that the skeleton graph of A is typically much sparser than the graph of A (see Table 2).

Each FIND(p) operation traverses the find path in p's tree beginning at p and ending at the root of the tree. The average number of vertices on these find paths is reported for each DSU implementation. We tested only two options for `nodcnt`: path compression and path halving, both without union by rank. Note that the average number of vertices on a find path ranges from 2 to 2.7, with path compression faring slightly better than path halving. The performance of path compression suffers, however, because the find path must be traversed twice, compared with once for path halving. Our tests indicate that path halving does indeed substantially outperform path compression, and in `nodcnt`, where the number of FIND's is large, the gain in efficiency is substantial.

We tried all six options mentioned in §3.3 in our implementations of `supcnt` and, as noted earlier, we saw little difference in performance from one option to the next. The primary explanation for this phenomenon is the small proportion of `supcnt`'s total work devoted to DSU operations. The number of FIND operations is small relative to m, and the average number of vertices on a find path is small (from 1.4 to 2.6) for five of the six options tested. For the sixth option (no DSU enhancements), the average number of vertices on a find path is still quite modest (from 3.6 to 5.8), with less work required for each vertex visited. Consequently, even this option is competitive in our tests.

When path compression or path halving is used, union by rank obtains only modest reductions in the average number of nodes visited. The overhead costs associated with union by rank more than offset any advantages conferred by the technique. Comparing path compression and path halving with no union by rank, the same observations made previously for `nodcnt` hold for `supcnt` also. The primary difference is that the total work associated with DSU operations in `supcnt` is so small that the performance edge of path halving over path compression is quite small. Nonetheless, path halving with no union by rank has proven most effective overall and has the added advantage of simplicity. Finally, note that for our chosen option the total number of vertices visited by FIND operations is much less than m for most of the test problems.

TABLE 1
List of test problems.

Problem	Brief description
NASA1824	Structure from NASA Langley, 1824 degrees of freedom
NASA2910	Structure from NASA Langley, 2910 degrees of freedom
NASA4704	Structure from NASA Langley, 4704 degrees of freedom
BCSSTK13	Stiffness matrix—fluid flow generalized eigenvalues
BCSSTK14	Stiffness matrix—roof of Omni Coliseum, Atlanta
BCSSTK15	Stiffness matrix—module of an offshore platform
BCSSTK16	Stiffness matrix—Corps of Engineers dam
BCSSTK17	Stiffness matrix—elevated pressure vessel
BCSSTK18	Stiffness matrix—R. E. Ginna nuclear power station
BCSSTK23	Stiffness matrix—portion of a 3D globally triangular building
BCSSTK24	Stiffness matrix—winter sports arena

TABLE 2
Problem statistics.

Problem	Dimension n	Edges in $G(A)$ m	Edges in $G^-(A)$ m^-	Edges in $G^+(A)$ m^+
NASA1824	1824	18692	3565	71875
NASA2910	2910	85693	8113	201493
NASA4704	4704	50026	9672	276768
BCSSTK13	2003	40940	5598	269668
BCSSTK14	1806	30824	4352	110461
BCSSTK15	3948	56934	13186	647274
BCSSTK16	4884	142747	11665	736294
BCSSTK17	10974	208838	24569	994885
BCSSTK18	11948	68571	23510	650777
BCSSTK23	3134	21022	8500	417177
BCSSTK24	3562	78174	6977	275360

TABLE 3
Average number of vertices on a find path for DSU *implementation options*: PC *is path compression,* PH *is path halving,* R *is union by rank, and* NR *is no union by rank.*

Problem	nodcnt			supcnt						
	vertices on path		FIND's	vertices on path						FIND's
	PC	PH		none		PC		PH		
	NR			NR	R	NR	R	NR	R	
NASA1824	2.1	2.3	17050	4.1	1.9	2.3	1.6	2.5	1.6	1923
NASA2920	2.0	2.1	83071	3.6	1.6	2.1	1.4	2.2	1.4	5491
NASA4704	2.2	2.3	45809	4.1	1.9	2.2	1.6	2.4	1.6	5455
BCSSTK13	2.1	2.2	39125	5.5	2.2	2.2	1.8	2.4	1.8	3783
BCSSTK14	2.1	2.2	29200	4.2	1.7	2.2	1.5	2.3	1.5	2728
BCSSTK15	2.2	2.3	53468	4.7	2.0	2.2	1.6	2.3	1.7	9720
BCSSTK16	2.1	2.1	138121	4.4	2.0	2.1	1.7	2.2	1.7	7039
BCSSTK17	2.1	2.1	199092	4.1	1.9	2.2	1.6	2.2	1.6	14823
BCSSTK18	2.3	2.5	59624	5.5	2.2	2.5	1.8	2.8	1.9	14563
BCSSTK23	2.4	2.7	18419	5.8	2.4	2.4	1.9	2.6	1.9	5897
BCSSTK24	2.0	2.1	74762	3.8	1.7	2.1	1.6	2.2	1.6	3565

TABLE 4
Run times in seconds on an IBM RS/6000 (*model* 320).

Problem	E-tree	Post-ordering	Row/column counts			Super-nodes
			Liu's	New		
			`lnzcnt`	`nodcnt`	`supcnt`	
NASA1824	.035	.006	.076	.047	.038	.031
NASA2920	.156	.009	.256	.198	.144	.128
NASA4704	.096	.016	.261	.128	.104	.085
BCSSTK13	.078	.006	.238	.098	.074	.064
BCSSTK14	.057	.005	.118	.074	.056	.048
BCSSTK15	.108	.013	.513	.142	.113	.091
BCSSTK16	.262	.016	.691	.331	.239	.216
BCSSTK17	.391	.037	.965	.500	.408	.329
BCSSTK18	.144	.040	.549	.197	.181	.141
BCSSTK23	.044	.010	.310	.059	.054	.039
BCSSTK24	.143	.012	.295	.184	.134	.120

4.2. Performance of the row and column count algorithm. We coded `nodcnt` and `supcnt` in Fortran 77 and ran our tests on an IBM RS/6000 (model 320). We used the standard Fortran compiler and compiler optimization flag (`xlf -O`). We used a high-resolution timer (`readrtc`) to obtain our timings on this machine, repeating each run ten times in succession and returning the average elapsed time. The results are shown in Table 4. We used path halving and no union by rank in the implementation of the disjoint set union algorithm for both `nodcnt` and `supcnt`. The time required to compute the elimination tree and postordering are of interest for two reasons. First, they must be computed before the row/column counts can be computed. Second, the algorithm for computing the elimination tree is, like `nodcnt` and `supcnt`, a single-pass $O(m\,\alpha(m,n))$ algorithm that relies on efficient implementation of the disjoint set union operations for efficiency. Thus it is interesting to compare its performance with that of the new algorithms.

Both `nodcnt` and `supcnt` are much more efficient than `lnzcnt`, the $O(|L|)$ algorithm from Liu [19]. Algorithm `nodcnt` is 1.29 to 5.25 times faster than `lnzcnt`, while `supcnt` is, in turn, 1.08 to 1.39 times faster than `nodcnt`. For every problem but one, `supcnt` is at least twice as fast as `lnzcnt`. (For NASA2920, `supcnt` is 1.77 times faster than `lnzcnt`.) For four of the problems, `supcnt` is more than three times faster than `lnzcnt`. For BCSSTK15 `supcnt` is 4.54 times faster, and for BCSSTK23 `supcnt` is 5.74 times faster.

Finally, it is interesting to compare the timings for the elimination tree algorithm [19] and the supernode algorithm [20] with those for `supcnt`. First, `supcnt` can be viewed as an extension of the supernode algorithm, and consequently the time for `supcnt` should be bounded below by the time for the supernode algorithm. Though there are some differences in the amount and kind of $O(n)$ work performed by the two algorithms before and after the main loop, the difference in the two timings can nevertheless be viewed as a crude measure of the cost of adding the instructions necessary to compute row and column counts to the supernode algorithm. Clearly, this cost is quite small, especially considering the simplicity and demonstrated practical efficiency of the supernode algorithm. Note also that the timings for `supcnt` and the elimination tree algorithm closely track each other. From these observations, we conclude that it is probably not possible to improve the performance of `supcnt` much beyond what we are currently observing.

5. Conclusion. We have considered in this paper the problem of predicting the row counts and column counts in the Cholesky factor L of a sparse symmetric positive definite matrix A, given the zero/nonzero structure of A and the elimination tree $T(A)$. We have presented new algorithms for determining the counts, the complexities of which are linear in $|A|$ times a slowly growing inverse of Ackermann's function; the previously known algorithms ran in $O(|L|)$ time. The key to the new algorithms is the computation of least common ancestors in a tree using the disjoint set union algorithm. We have investigated different ways of implementing the disjoint set union operations in our algorithms. Based on our experimental results, we conclude that path halving with no union by rank is the best technique for an efficient implementation of the disjoint set union algorithm.

We have further improved our new algorithms by exploiting the skeleton graph of A. We have demonstrated that the supernodal version is faster than the nodal version in all of the problems we tested. Moreover, both the nodal and supernodal versions are much more efficient than the previously known $O(|L|)$-time algorithms. We expect the algorithms described in this paper to be of practical use in a wide range of sparse matrix computations.

REFERENCES

[1] C. C. ASHCRAFT, *A Vector Implementation of the Multifrontal Method for Large Sparse, Symmetric Positive Definite Linear Systems*, Tech. Report ETA-TR-51, Engineering Technology Applications Division, Boeing Computer Services, Seattle, WA, 1987.

[2] C. 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.

[3] H. BODLAENDER, J. R. GILBERT, H. HAFSTEINSSON, AND T. KLOKS, *Approximating treewidth, pathwidth, frontsize, and minimum elimination tree height*, J. Algorithms, to appear.

[4] E. C. H. CHU, A. GEORGE, J. W-H. LIU, AND E. G-Y. NG, *User's Guide for SPARSPAK-A: Waterloo Sparse Linear Equations Package*, Tech. Report CS-84-36, University of Waterloo, Waterloo, Ontario, 1984.

[5] T. COLEMAN AND Y. LI, *On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds*, Tech. Report 92–1314, Cornell University Computer Science Department, Ithaca, NY, 1992.

[6] ———, *A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on the Variables*, Tech. Report 92–1315, Cornell University Computer Science Department, Ithaca, NY, 1992.

[7] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, Oxford, England, 1987.

[8] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.

[9] S. C. EISENSTAT, M. C. GURSKY, M. H. SCHULTZ, AND A. H. SHERMAN, *The Yale sparse matrix package* I: *The symmetric codes*, Internat. J. Numer. Meth. Engrg., 18 (1982), pp. 1145–1151.

[10] H. N. GABOW AND R. E. TARJAN, *A linear time algorithm for a special case of disjoint set union*, J. Comput. Syst. Sci., 30 (1985), pp. 209–221.

[11] F. GAVRIL, *The intersection graphs of subtrees in trees are exactly the chordal graphs*, J. Combinatorial Theory B, 16 (1974), pp. 47–56.

[12] A. GEORGE AND M. T. HEATH, *Solution of sparse linear least squares problems using Givens rotations*, Linear Algebra Appl., 34 (1980), pp. 69–83.

[13] A. GEORGE, M. T. HEATH, J. W-H. LIU, AND E. G-Y. NG, *Solution of sparse positive definite systems on a shared memory multiprocessor*, Internat. J. Parallel Programming, 15 (1986), pp. 309–325.

[14] ———, *Sparse Cholesky factorization on a local-memory multiprocessor*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 327–340.

[15] A. GEORGE AND J. W-H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[16] M. T. HEATH, E. NG, AND B. W. PEYTON, *Parallel algorithms for sparse linear systems*, SIAM Rev., 33 (1991), pp. 420–460.
[17] J. G. LEWIS, B. W. PEYTON, AND A. POTHEN, *A fast algorithm for reordering sparse matrices for parallel factorization*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1156–1173.
[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] ———, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.
[20] J. W-H. LIU, E. NG, AND B. W. PEYTON, *On finding supernodes for sparse matrix computations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 242–252.
[21] A. POTHEN, *Simplicial Cliques, Shortest Elimination Trees, and Supernodes in Sparse Cholesky Factorization*, Tech. Report CS-88-13, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1988.
[22] D. J. ROSE, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Read, ed., Academic Press, 1972, pp. 183–217.
[23] D. J. ROSE, R. E. TARJAN, AND G. S. LUEKER, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
[24] R. SEDGEWICK, *Algorithms*, Addison-Wesley, Reading, MA, 1983.
[25] A. H. SHERMAN, *On the efficient solution of sparse systems of linear and nonlinear equations*, Ph.D. thesis, Yale University, New Haven, CT, 1975.
[26] R. E. TARJAN, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.
[27] ———, *Efficiency of a good but not linear set union algorithm*, J. ACM, 22 (1975), pp. 215–225.
[28] ———, *Applications of path compression on balanced trees*, J. ACM, 26 (1979), pp. 690–715.
[29] ———, *Data Structures and Network Algorithms*, CBMS-NSF Regional Conference Series in Applied Math, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

SIAM J. MATRIX. ANAL. APPL.
Vol. 15, No. 4, pp. 1092–1107, October 1994

004

CONVEXITY AND CONCAVITY OF THE PERRON ROOT AND VECTOR OF LESLIE MATRICES WITH APPLICATIONS TO A POPULATION MODEL *

STEPHEN J. KIRKLAND† AND MICHAEL NEUMANN‡

Abstract. This paper considers the Leslie model of population growth and analyzes its asymptotic growth rate and its asymptotically stable age distribution as functions of the fecundity and survival rates of each age group in the population. This analysis is performed by computing first- and second-order partial derivatives of the Perron root and vector of a Leslie matrix with respect to each relevant entry in the matrix, with emphasis on the second partial derivatives. The signs of these derivatives as well as the qualitative implications that the results have for the Leslie model are discussed. Where possible, quantitative interpretations of the results are also given. Throughout, the techniques employ ideas from the theory of group generalized inverses.

Key words. Leslie matrix, group inverse, Perron root, Perron vector

AMS subject classifications. 15A09, 15A18, 92D25

1. Introduction.

This paper investigates first- and second-order effects of changes in fecundity and survival rates of various age groups on the asymptotic rate of growth and the asymptotically stable age distribution vector of the Leslie population model. This population model can be represented by a nonnegative matrix whose Perron root and an appropriately normalized Perron eigenvector that furnish the rate of growth and the stable age distribution of the model, respectively. The first- and second-order effects are obtained by computing the first- and second-order partial derivatives of the Perron root and Perron vector with respect to those matrix entries that represent the fecundity and survival rates of each age group. The existence of these derivatives is assured because, in the problem's setting, the Perron root is simple. It should be mentioned that first-order effects of changes in the fecundity and survival rates upon the growth rates of the model have already been obtained by authors such as Demetrius [8], Goodman [14], and Lal and Anderson [17].

In the Leslie model it is assumed that the population consists of n age groups. Let $x_i(t)$ denote the number of individuals in the ith age group at time t. Let $F_i, i = 1, \ldots, n$, denote the fecundity of each individual in the ith age group and let $P_i, i = 1, \ldots, n-1$, denote the probability of survival of an individual from age i to age $i+1$. Assume that both the fecundity and survival rates are independent of the time t. Then, as can be readily ascertained, the age distribution at time $t+1, t \geq 0$, can be

*Received by the editors January 11, 1993; accepted for publication (in revised form) May 10, 1993.

†Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, Minnesota 55455 (kirkland@max.cc.uregina.ca).

‡Department of Mathematics, University of Connecticut, Storrs, Connecticut 06269-3009 (neumann@uconnvm.uconn.edu). This author's research was supported by National Science Foundation grants DMS-8901860 and DMS-9007030.

described by the matrix–vector relation

$$x(t+1) = \begin{pmatrix} F_1 & F_2 & \cdots & \cdots & F_{n-1} & F_n \\ P_1 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & 0 & 0 \\ \vdots & \cdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & P_{n-1} & 0 \end{pmatrix} x(t) =: \tilde{A}x(t), \tag{1.1}$$

where $x(\tau) = (x_1(\tau) \cdots x_n(\tau))^T$ for all $\tau \geq 0$. For more background material on the Leslie population model, see Pollard [20] and Caswell [4].

Papers in demography [8] say that the population has reached a *stationary age distribution* if there exists a time t_0 and a constant $\lambda > 0$ such that

$$x(t+1) = \lambda x(t) \quad \forall t > t_0. \tag{1.2}$$

In this case λ is called the *growth rate of the population.* The term stationary age distribution comes from the fact that if (1.2) holds, then from time $t_0 + 1$ onward the ratio between the various age groups in the population is maintained. We note further that if (1.2) holds, then

$$\lambda x(t) = x(t+1) = \tilde{A}x(t) = \cdots = \tilde{A}^{t+1}x(0) \quad \forall t > t_0. \tag{1.3}$$

Because a stationary age distribution is reached in finite time only when the initial age distribution vector $x(0)$ is a linear combination of a Perron vector and a generalized null vector of $\tilde{A}$, we prefer to think of Demetrius' notions of growth rate of the population and stationary age distribution as the *asymptotic growth rate* and the *asymptotically stable age distribution vector*, respectively.

If $\tilde{A}$ is an irreducible matrix, then by the Perron–Frobenius theory (see §2 for preliminaries and Berman and Plemmons [2] for a comprehensive background), λ must be the Perron root of $\tilde{A}$ and $x(t)$ is a corresponding right Perron (eigen)vector and the theory guarantees the existence and differentiability of both, the latter subject to an appropriate normalization. Furthermore, if $\tilde{A}$ is *primitive*, namely, it is irreducible with a single eigenvalue of maximum modulus, then, for example, the power method for computing dominant eigenvalues and corresponding normalized eigenvectors [21, p. 340] shows that the righthand side of (1.3) will always converge to a Perron vector of $\tilde{A}$ so that asymptotically (1.2) holds. We comment now that just as the entries of the nonnegative matrix $\tilde{A}$ have a physical interpretation, we see that the Perron root and vector and their derivatives also have a physical meaning for the model.

One can relax the assumption of irreducibility. Even then the special structure of the Leslie matrix $\tilde{A}$ implies that its Perron root remains simple, thus ensuring that the Perron root and vector are still differentiable. While the reducible case may lead to some interesting questions, we nevertheless restrict ourselves to the case of irreducible Leslie matrices, i.e., we always assume that $F_n > 0$ since this captures the largest part of the mathematical content and difficulty.

It is natural to ask how the growth rate and the stable age distribution vectors are affected as we change the fecundity and survival probabilities at each age group. Demetrius [8] has investigated one of these questions by looking at the derivatives of the Perron root with respect to the fecundities and the survival rates that he obtained

via the characteristic equation for a Leslie matrix. His results can also be obtained from a more general theorem giving expressions of the partial derivatives of a simple eigenvalue of a matrix with respect to the matrix entries as follows: Let $B = (b_{i,j})$ be an $n \times n$ real or complex matrix and let μ be a simple eigenvalue of B. Let ξ and η be right and left eigenvectors of B corresponding to μ normalized, such that $\eta^T \xi = 1$. Then it is known (see, for example, Stewart [21, Exer. 1, p. 305]) that

$$\frac{\partial \mu(B)}{\partial_{i,j}} = \xi_j \eta_i \quad \forall 1 \leq i, j \leq n, \tag{1.4}$$

where $\partial\mu/\partial_{i,j}$ is the derivative of μ with respect to the (i,j)th entry at B.[1]

Consider the matrix $C = \mu I - B$. Zero now is a simple eigenvalue of C and therefore the group generalized inverse of $C, C^{\#}$, exists and, as is known, $\xi\eta^T = I - CC^{\#}$. Thus, we see that the group inverse of C can be used to express first-order partial derivatives of μ with respect to the matrix entries at B. Furthermore, as is shown in Deutsch and Neumann [9], the second-order partial derivatives of μ with respect to the (i,j)th entry can also be written in terms of $C^{\#}$. Specifically, they showed that

$$\frac{\partial^2 \mu(B)}{\partial_{i,j}^2} = 2\left(I - CC^{\#}\right)_{j,i} C^{\#}_{j,i}, \quad \forall 1 \leq i, j \leq n. \tag{1.5}$$

Assume that B is an $n \times n$ nonnegative and irreducible matrix. Since any right and left Perron vectors of B can be chosen positive, (1.4) readily confirms the well-known fact that the Perron value is a strictly increasing function in any of the matrix entries. In a series of papers, Cohen [5]–[7] established the fact that the Perron root is a convex function in the main diagonal of the matrix. His principal approach to proving this fact was probabilistic relying on evolution equations due to Kac. In [9] a matrix theoretic proof of this fact is presented. Moreover, from the formula (1.5), which was found in [9], we see that *for any pair (i,j), the convexity or concavity of the Perron root with respect to (i,j)th entry is determined by the sign of the (j,i)th entry of $C^{\#} = (\lambda I - B)^{\#}$*. Perturbation and convexity theory of the Perron root have been investigated by a number of authors. To list a few we mention Elsner [11], Friedland [13], Golub and Meyer [15], Haviv, Ritov, and Rothblum [16], and Meyer and Stewart [19].

Let us return to our Leslie matrix $\tilde{A}$ and assume that it is irreducible. In this paper we explicitly compute expressions for the first- and second-order derivatives of the Perron root and an appropriately normalized Perron vector of a Leslie matrix with respect to its entries in the first row and on its subdiagonal. We then interpret what meaning our results have for the population model that the matrix represents. *Most of our results show that younger ages exert more influence on the behavior of the growth rate and the stable age distribution vector than older age groups do.* Our experience is that it is much more difficult to analyze the effects of changes in survival rates on the population than it is to analyze changes in the fecundity rates.

In §2 we present further notation and preliminaries necessary for the work here. We obtain an explicit formula for the group inverse of $\lambda I - \tilde{A}$. In §3 we derive formulas

[1] See §2 for a more precise discussion of the partial derivatives of eigenvalues of matrix with respect to matrix entries.

for the second-order derivatives of the Perron root of a Leslie matrix with respect to its top row and subdiagonal. In doing so, some results of Deutsch and Neumann in [10] on the first and second derivatives of the Perron vector are extended. In §4 we also extend for the partial derivatives of the Perron vector of $\tilde{A}$. In §§3 and 4 we explain the implications of our results on the population model both qualitatively and where possible quantitatively.

2. Notation and preliminaries. In this paper we use the following notation.

$\mathbb{R}^k$ denotes the k-dimensional real space.

$\mathbb{R}^{k,k}$ denotes the space of all $k \times k$ real matrices.

$e \in \mathbb{R}^k$ denotes the k-dimensional vector whose entries are all 1's.

$e_i \in R^k$ denotes the k-dimensional unit coordinate vector, $i = 1, \ldots, n$.

$E_{i,j} \in \mathbb{R}^{k,k}$ is the matrix whose (i,j)th entry is 1 and whose remaining entries are 0.

The symbol $\sim$ indicates algebraic expressions with the same sign. Let $u = (u_1, \ldots, u_k) \in \mathbb{R}^k$ and $v = (v_1, \ldots, v_k) \in \mathbb{R}^k$. We write that $u \geq v$, if $u_i \geq v_i, i = 1, \ldots, k$. We say that u *majorizes* v, in notation $u \succeq v$, if

$$\sum_{j=1}^{i} u_j \geq \sum_{j=1}^{i} v_j, \quad i = 1, \ldots, k-1 \quad \text{and} \quad \sum_{j=1}^{k} u_j = \sum_{j=1}^{k} v_j.$$

Let $C \in \mathbb{R}^{n,n}$ and consider the matrix equations

$$CXC = C, \quad XCX = X, \quad \text{and} \quad CX = XC.$$

A matrix $X \in \mathbb{R}^{n,n}$ which satisfies all three equations, if it exists, is called the *group inverse of* C and is denoted by $C^{\#}$. Moreover, if $C^{\#}$ exists and D is any nonsingular matrix, then

$$(D^{-1}CD)^{\#} = D^{-1}C^{\#}D. \tag{2.1}$$

In particular, if D is a diagonal matrix whose diagonal entries are all positive, then all the entries of the matrices $C^{\#}$ and $D^{-1}C^{\#}D$ have identical signs in the same locations. It is known [1], [3] that a necessary and sufficient condition for $C^{\#}$ to exist, the elementary divisors, if any, of C corresponding to the eigenvalue zero are all linear. Thus, if B is an $n \times n$ nonnegative and irreducible matrix so that its Perron root is simple, then 0 is a simple eigenvalue of $C = \lambda I - B$, showing that the group inverse of C exists.

Suppose that $B \in \mathbb{R}^{n,n}$ has a simple eigenvalue called $\mu(B)$. It readily follows from considerations involving the minimal polynomial that there is an open ball in $\mathbb{R}^{n,n}$ about B such that every matrix in the ball has a simple eigenvalue. Thus, for any $E \in \mathbb{R}^{n,n}$ and for sufficiently small $t \in \mathbb{R}, B + tE$ has a simple eigenvalue $\mu(B + tE)$, such that $\mu(B + tE) \to \mu(B)$ as $t \to 0$. Wilkinson [22, pp. 66–67] shows that for sufficiently small $t, \mu(B + tE)$ can be expanded in a convergent power series about B. Hence, the derivatives of all orders of μ with respect to t exist at B. In particular, the partial derivative of μ with respect to the (i,j)th entry at B is given by the limit

$$\frac{\partial \mu}{\partial_{i,j}} := \frac{\partial \mu(B)}{\partial_{i,j}} = \lim_{t \to 0} \frac{\mu(B + tE_{i,j}) - \mu(b)}{t} \overset{\text{by}(1.4)}{=} \xi_j \eta_i, \tag{2.2}$$

where $\xi = \xi(B) = (\xi_1 \cdots \xi_n)^T$ and $\eta = \eta(B) = (\eta_1 \cdots \eta_n)^T$ are right and left eigenvectors of B corresponding to $\mu(B)$ normalized so that their inner product is 1. In a

similar manner we define the higher-order partial derivatives of μ at B with respect to the matrix entries. Wilkinson goes on to show that if one of many standard normalizations (but not the infinity norm) is applied to the eigenvector corresponding to $\mu(B+tE)$ throughout the ball, then the entries of the corresponding eigenvector can be expanded in a convergent power series. Therefore they too are differentiable with respect to t at B. When it is absolutely clear from the context, we shall suppress the letter representing the matrix from the expressions for the partial derivatives, viz., we write $\partial\mu/\partial_{i,j}$ for $\partial\mu(B)/\partial_{i,j}$ and so on.

Our approach is to first develop our results for stochastic Leslie matrices and then transform them to general Leslie matrices from which we shall be able to draw our conclusion for the population model under consideration. To do so it is helpful to have formulas connecting the partial derivatives with respect to the Perron root of a general nonnegative and irreducible matrix $\tilde{B}$ and the stochastic and irreducible matrix B to which $\tilde{B}$ is transformed using the diagonal similarity

$$B = \frac{1}{\lambda} D^{-1} \tilde{B} D, \tag{2.3}$$

where

$$D = \operatorname{diag}\ (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \tag{2.4}$$

and where $\tilde{x} = (\tilde{x}_1 \cdots \tilde{x}_n)^T$ is a right eigenvector of $\tilde{B}$ corresponding to its Perron root λ. Throughout we shall normalize all right Perron vectors so that their first entry equals 1. Thus, if $\tilde{y}$ is the left Perron vector of $\tilde{B}$ normalized so that $\tilde{x}^T\tilde{y} = 1$, then it follows from (1.4), (1.5), (2.1), (2.3), and (2.4) that

$$\frac{\partial^2 \lambda}{\partial_{i,j}^2} = \frac{2\tilde{y}_i}{\lambda} \frac{(\tilde{x}_j)^2}{\tilde{x}_i} Q_{j,i}^{\#}, \tag{2.5}$$

where $Q = I - B$. Similarly, relations for the eigenvectors will also be useful. In particular, the following formulas can be obtained:

$$\frac{\partial \tilde{x}}{\partial_{i,j}} = \frac{\tilde{x}_j}{\lambda \tilde{x}_i} D \frac{\partial x}{\partial_{i,j}}, \qquad 1 \le i, j \le n \tag{2.6}$$

and

$$\frac{\partial^2 \tilde{x}}{\partial_{1,i}^2} = \frac{\tilde{y}_1 \tilde{x}_i}{\lambda^2} D \frac{\partial^2 x}{\partial_{1,i}^2}, \qquad 1 \le i \le n. \tag{2.7}$$

For the specific case of the Leslie matrix $\tilde{A}$ as given in (1.1), the right Perron vector is given by

$$\tilde{x} = \left(1 \ \frac{P_1}{\lambda} \ \frac{P_1 P_2}{\lambda^2} \cdots \frac{P_1 \cdots P_{n-1}}{\lambda^{n-1}}\right)^T. \tag{2.8}$$

With (2.3) and (2.4) in mind, we find that $\tilde{A}$ can be transformed into the *stochastic and irreducible* Leslie matrix

$$A = \begin{pmatrix} a_1 & a_2 & \cdots & \cdots & a_{n-1} & a_n \\ 1 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & 0 & 0 \\ \vdots & \cdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 & 0 \end{pmatrix}, \tag{2.9}$$

where

$$a_1 = \frac{F_1}{\lambda} \geq 0 \quad \text{and} \quad a_i = \frac{P_1, \ldots, P_{i-1}F_i}{\lambda^i} \geq 0, \qquad i = 2, \ldots, n, \tag{2.10}$$

with $a_n > 0$.

LEMMA 2.1. *Let A be an irreducible stochastic Leslie matrix whose top row is given by $(a_1 \cdots a_n)$. Set*

$$Q = I - A. \tag{2.11}$$

Then

$$Q^{\#} = \begin{bmatrix} (Q^{\#})_{1,1} & (Q^{\#})_{1,2} \\ (Q^{\#})_{2,1} & (Q^{\#})_{2,2} \end{bmatrix}, \tag{2.12}$$

where

$$\begin{aligned} (Q^{\#})_{1,1} = {} & M^{-1} + \chi M^{-1} e r^T M^{-1} + \chi e r^T M^{-2} \\ & + \chi^2 (r^T M^{-2} e)\, e r^T M^{-1}, \end{aligned} \tag{2.13}$$

$$(Q^{\#})_{1,2} = -\chi M^{-1} e - \chi^2 (r^T M^{-2} e)\, e, \tag{2.14}$$

$$(Q^{\#})_{2,1} = \chi r^T M^{-2} + \chi^2 (r^T M^{-2} e)\, r^T M^{-1}, \tag{2.15}$$

and

$$(Q^{\#})_{2,2} = -\chi^2 (r^T M^{-2} e), \tag{2.16}$$

and where $e \in \mathbb{R}^{n-1}$,

$$r = \begin{pmatrix} 0 \\ \vdots \\ -1 \end{pmatrix} \in \mathbb{R}^{n-1} \quad \textit{and} \quad \chi = \frac{1}{1 - r^T M^{-1} e} = \frac{1}{\frac{1}{a_n} \sum_{i=0}^{n-1} (1 - s_i)} \tag{2.17}$$

and

$$M^{-1} = \frac{1}{a_n} \begin{pmatrix} 1 - s_0 & s_{n-1} - s_1 & s_{n-1} - s_2 & \cdots & s_{n-1} - s_{n-3} & s_{n-1} - s_{n-2} \\ 1 - s_0 & 1 - s_1 & s_{n-1} - s_2 & \cdots & s_{n-1} - s_{n-3} & s_{n-1} - s_{n-2} \\ 1 - s_0 & 1 - s_1 & 1 - s_2 & \cdots & s_{n-1} - s_{n-3} & s_{n-1} - s_{n-2} \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 1 - s_0 & 1 - s_1 & 1 - s_2 & \cdots & 1 - s_{n-3} & 1 - s_{n-2} \end{pmatrix}. \tag{2.18}$$

Here

$$s_0 = 0 \quad \textit{and} \quad s_i = \sum_{j=1}^{i} a_j \quad \forall 1 \leq i \leq n-1. \tag{2.19}$$

Proof. Let M be the $(n-1)\times(n-1)$ leading principal submatrix of Q. It is not difficult to check using the fact that the a_i's sum to 1 that Q admits the full rank factorization

$$Q = \begin{pmatrix} M \\ r^T \end{pmatrix} (I \quad -e) =: BC. \tag{2.20}$$

According to Ben-Israel and Greville [1], $Q^{\#} = B(CB)^{-2}C$. Now it can be verified that $(CB)^{-1} = M^{-1} + \chi M^{-1}er^TM^{-1}$, where χ and M^{-1} are as given in (2.17) and (2.18), respectively. That $Q^{\#}$ is given by (2.12) and as specified in (2.13)–(2.19) can now be ascertained from the above partitioning of C and B and the aforementioned formula for $(CB)^{-1}$. □

We comment that the above lemma is a specialization to the case of a singular M–matrix obtained from an $n\times n$ irreducible Leslie matrix of a formula for the group inverse of a singular M–matrix obtained from a general $n\times n$ nonnegative and irreducible matrix found by Meyer [18, Thm. 5.2].

3. The Perron root of a Leslie matrix as a function of top row and subdiagonal. We begin by determining the second-order behavior of the Perron root as a function of the top row.

THEOREM 3.1. *Let A be an $n\times n$ irreducible stochastic Leslie matrix whose top row is given by $(a_1,\ldots,a_n)$. Then the entries in the first column of the group inverse of $Q = I - A$ are given by*

$$Q^{\#}_{i,1} = \frac{\chi}{a_n}\left[n-i-\frac{\chi}{a_n}\sum_{j=0}^{n-2}(1-s_j)(n-1-j)\right], \qquad i=1,\ldots,n, \tag{3.1}$$

where χ and the s_j's are as in (2.17) *and* (2.19). *In particular, the entries of the first column of $Q^{\#}$ are strictly decreasing from first to last. Moreover, there exists an integer $k_0 < (n+1)/2$ such that $Q^{\#}_{1,1},\ldots,Q^{\#}_{k_0,1}$ are all nonnegative, whereas $Q^{\#}_{k_0+1,1},\ldots,Q^{\#}_{n,1}$ are all nonpositive.*

Proof. For each $1\le i\le n-1$, we find from (2.13) that

$$Q^{\#}_{i,1} = e_i^T\left[M^{-1} + \chi M^{-1}er^TM^{-1} + \chi er^TM^{-2} + \chi^2(r^TM^{-2}e)er^TM^{-1}\right]e_1.$$

Also from (2.18), we have that

$$r^TM^{-2}e = (r^TM^{-1})(M^{-1}e) = \sum_{j=0}^{n-2}\frac{1-s_j}{a_n}(n-2-j) - \left(\sum_{j=0}^{n-2}\frac{1-s_j}{a_n}\right)^2. \tag{3.2}$$

It now follows after several algebraic reductions that

$$Q^{\#}_{i,1} = \frac{\chi}{a_n}(1-s_0)\left(1-\chi r^TM^{-2}e - e_i^TM^{-1}e\right).$$

Further algebraic manipulations now yield (3.1). A similar calculation shows that (3.1) holds also for the case $i=n$.

From (3.1) it readily follows that the entries in the first column of $Q^{\#}$ are strictly decreasing from first to last and that the first entry is always positive (which is a

general result of Meyer [18] for all diagonal entries of the group inverse of a singular and irreducible M–matrix, and it is also an outcome of Cohen's results described in the introduction), whereas the last entry is always negative. To complete the proof we need only show that if $i \geq (n+1)/2$, then $Q^{\#}_{i,1} \leq 0$. From (3.1) we find that for $2 \leq i \leq n-1, Q^{\#}_{i,1}$ is nonpositive if and only if

$$(n-i)\sum_{j=0}^{n-1}(1-s_j) \leq \sum_{j=0}^{n-1}(1-s_j)(n-1-j). \tag{3.3}$$

But (3.3) holds if and only if

$$\sum_{j=0}^{n-1-i}(1-s_{j+i})(j+1) \leq \sum_{j=0}^{i-2}(1-s_{j+2-i})(j+1). \tag{3.4}$$

Since $1-s_{j+i} \leq 1-s_{j+2-i}$ for each $0 \leq j \leq n-i-1$, it now follows that if $i \geq (n+1)/2$, then (3.4) always holds. Consequently, $Q^{\#}_{i,1} \leq 0$ whenever $i \geq (n+1)/2$. □

Several comments are in order. (i) Theorem 3.1 shows that the sign change in the first column of $Q^{\#}$ always occurs somewhere before the $\lfloor (n+1)/2 \rfloor$th position. The following example shows that the sign change can occur at any position after the first and before the $(n+1)/2$th if n is odd or before the $\lfloor (n+1)/2 \rfloor$th if n is even. Fix $2 \leq k < (n+1)/2$. Let A be given by (2.9) with

$$a_j = \begin{cases} \alpha & \text{if } j = k-1, \\ 1-\alpha & \text{if } j = n, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\frac{n(n-1-2k)}{(n-k)(n-k-1)-2} < \alpha < \frac{n(n+1-2k)}{(n-k)(n-k+1)} < 1$$

with $0 \leq \alpha \leq 1$. Then a straightforward exercise shows that $Q^{\#}_{k,1} > 0 > Q^{\#}_{k+1,1}$ and that such an α causes the sign change to occur at the desired admissible position. (ii) Our next comment, which is given in a form of lemma, shows that under certain conditions on the stochastic Leslie matrices A and $\hat{A}$, there is a relationship between $Q^{\#}_{i,1}$ and $(\hat{Q})^{\#}_{i,1}, 1 \leq i \leq n$. The proof of the lemma follows directly from (3.1).

COROLLARY 3.2. *Let $a = (a_1 \cdots a_n)$ and $\hat{a} = (\hat{a}_1 \cdots \hat{a}_n)$ be the top rows of the irreducible stochastic Leslie matrices A and $\hat{A}$, respectively. Suppose $a \succeq \hat{a}$. Then*

$$Q^{\#}_{n,1} \geq \left(\hat{Q}\right)^{\#}_{n,1} \Rightarrow Q^{\#}_{i,1} \geq \left(\hat{Q}\right)^{\#}_{i,1} \quad \forall 1 \leq i \leq n$$

and

$$Q^{\#}_{1,1} \leq \left(\hat{Q}\right)^{\#}_{1,1} \Rightarrow Q^{\#}_{i,1} \leq \left(\hat{Q}\right)^{\#}_{i,1} \quad \forall 1 \leq i \leq n.$$

Next we examine the behavior of the Perron root of a Leslie matrix as a function of its entries on subdiagonal. From (1.5) and the preceding discussion we know that it suffices to determine the signs of the superdiagonal entries of $Q^{\#}$. As we shall see, determining expressions to represent these entries is no more difficult than determining

expressions for the entries of $Q^{\#}$ down its first column. But determining the signs of the entries on the superdiagonal, especially in the top first half, appears to be more difficult than determining the signs of $Q^{\#}$ down the first column.

THEOREM 3.3. *Let A be an $n \times n$ irreducible stochastic Leslie matrix whose top row is given by $(a_1 \cdots a_n)$. Then, providing we interpret any summation sign from a lower limit that exceeds an upper limit to be zero, the entries in the superdiagonal positions of the group inverse of $Q = I - A$ are given by*

$$Q^{\#}_{k,k+1} = \chi \left\{ \left(\frac{1 - s_k}{a_n} \right) \left[n - k - 1 + -\chi \sum_{j=0}^{n-1} \left(\frac{1 - s_j}{a_n} \right) (n - 2 - j) \right] - \sum_{j=k}^{n-1} \left(\frac{1 - s_j}{a_n} \right) \right\}, \qquad k = 1, \ldots, n-1, \tag{3.5}$$

where χ and the s_j's are given by (2.17) *and* (2.19), *respectively. The positive entries on the superdiagonal of $Q^{\#}$, if any, appear consecutively from the $(1,2)$th entry and form a nonincreasing sequence. The $(k, k+1)$th entries, where $k \geq (n-1)/2$, are all negative and for $k \geq (n+1)/2$ those entries form a nondecreasing sequence.*

Proof. For each $1 \leq k \leq n-2$, we find from (2.13) that

$$Q^{\#}_{k,k+1} = e_k^T \left[M^{-1} + \chi M^{-1} e r^T M^{-1} + \chi e r^T M^{-2} + \chi^2 \left(r^T M^{-2} e \right) e r^T M^{-1} \right] e_{k+1}. \tag{3.6}$$

Also recall the expression for $r^T M^{-2} e$ given in (3.2). Examining each term in the expansion of (3.6) and referring to (2.18) we find that

$$e_k^T M^{-1} e_{k+1}^T = \frac{1 - s_k}{a_n} - 1,$$

$$e_k^T M^{-1} e = \sum_{j=0}^{n-2} \frac{1 - s_j}{a_n} - (n - k - 1),$$

$$r^T M^{-1} e_{k+1} = -\frac{1 - s_k}{a_n},$$

and

$$r^T M^{-2} e_{k+1} = \left(r^T M^{-1} \right) \left(M^{-1} e_{k+1} \right) = - \left[\frac{1 - s_k}{a_n} \sum_{j=0}^{n-2} \frac{1 - s_j}{a_n} - \sum_{j=0}^{k-1} \frac{1 - s_j}{a_n} \right].$$

A number of algebraic reductions now yield (3.5). A similar calculation shows that (3.5) holds also when $k = n-1$.

To see that the positive entries in the superdiagonal positions of $Q^{\#}$, if any, beginning at the (1,2) entry are consecutive and form a nonincreasing sequence, we need only show that if $Q^{\#}_{k,k+1} \geq 0$, then $Q^{\#}_{k-1,k} \geq Q^{\#}_{k,k+1}$. To this end, note that if $Q^{\#}_{k,k+1} > 0$, then, in particular, from (3.5) it follows that

$$n - k - 1 - \chi \sum_{j=0}^{n-1} \frac{1 - s_j}{a_n} (n - 2 - j) > 0. \tag{3.7}$$

But from (3.5) we find that

$$Q^{\#}_{k-1,k} - Q^{\#}_{k,k+1} = \chi \frac{s_k - s_{k-1}}{a_n} \left[n - k - 1 - \chi \sum_{j=0}^{n-1} \frac{1 - s_j}{a_n} (n - 2 - j) \right]. \tag{3.8}$$

The claim now follows from (3.7).

Now let $k \geq (n-1)/2$. From (3.5) we find that

$$Q^{\#}_{k,k+1} \sim (1 - s_k) \frac{\sum_{j=k}^{n-1} (1 - s_j)(j - k) - \sum_{j=0}^{k-1} (1 - s_j)(k - j)}{\sum_{j=0}^{n-1} (1 - s_j)} - \sum_{j=k+1}^{n-1} (1 - s_j).$$

The numerator in the first term of the above expression can be rearranged as

$$\sum_{j=1}^{n-k-1} j (s_{k-j} - s_{k+j}) - \sum_{j=n-k}^{k} j (1 - s_{k-j}),$$

which is evidently nonpositive. Consequently, $Q^{\#}_{k,k+1} \leq 0$. Next, consider the difference (3.8). Note that if $Q^{\#}_{k,k+1}$ is nonpositive and if

$$n - k - 1 - \chi \sum_{j=0}^{n-1} \frac{1 - s_j}{a_n} (n - 2 - j) \leq 0, \tag{3.9}$$

then, necessarily, $Q^{\#}_{k-1,k} \leq Q^{\#}_{k,k+1} \leq 0$. But (3.9) holds if and only if

$$a_n (n - k) + \sum_{j=k}^{n-2} (1 - s_j)[j - (k-1)] \leq \sum_{j=0}^{k-1} (1 - s_j)[(k-1) - j]. \tag{3.10}$$

If $k \geq (n+1)/2$, the terms in (3.10) can be paired off to yield the equivalent expression:

$$\begin{aligned} &[(k-1) - a_n (n-k)] \\ &\quad + (s_k - s_{k-2}) \cdot 1 \\ &\quad + (s_{k+1} - s_{k-3}) \cdot 2 \\ &\quad + \cdots \\ &+ (s_{n-2} - s_{2k-n})(n - 1 - k) \\ &\quad + (1 - s_{2k-(n+1)})(n - k) \\ &\quad + \cdots \\ &\quad + (1 - s_1)(k - 2) \geq 0. \end{aligned}$$

Observe that each of the terms on the lefthand side above is nonnegative. Since we already proved that $Q^{\#}_{n-1,n} \leq 0$, the proof of the theorem is now complete. □

Note that Theorem 3.3 implies that either the $Q^{\#}_{k,k+1}$'s are all nonpositive or there is an index k_0, necessarily less than $(n-1)/2$, such that for $1 \leq k \leq k_0, Q^{\#}_{k,k+1} \geq 0$ and for $k_0 + 1 \leq k \leq n-1, Q^{\#}_{k,k+1} \leq 0$. Although we are not able to show that there exist cases where k_0 attains the value $\lfloor (n-1)/2 \rfloor$, the following example shows that

some $Q^{\#}_{k,k+1}$ can be positive when n is sufficiently large and when k is not too large compared with n. Specifically, consider the $n \times n$ stochastic Leslie matrix whose top row is given by

$$a_j = \begin{cases} 17/20 & \text{if } j = k+1, \\ 3/20 & \text{if } j = n, \\ 0 & \text{otherwise.} \end{cases} \tag{3.11}$$

From (3.5) it can be shown that $Q^{\#}_{k,k+1}$ is positive provided that

$$k < \frac{-(102n+68) + \sqrt{(102n+68)^2 + 4(119)(21n^2 - 72n + 51)}}{238}. \tag{3.12}$$

For $k = 1$, (3.12) yields that the minimal value of n that guarantees that for the matrix given by (3.11), $Q^{\#}_{1,2} > 0$, is $n = 11$. For $k = 2$, the minimal such n is 17. As $n \to \infty$, the righthand side (3.12) is asymptotic to

$$n\left(\frac{-102 + \sqrt{20400}}{238}\right) \approx 0.1715n. \tag{3.13}$$

Let us now revert to a general Leslie matrix $\tilde{A}$. From the discussion in §2 we see by (2.5) that the derivative of the Perron root with respect to the entries in the top row at $\tilde{A}$ is given by

$$\frac{\partial^2 \lambda}{\partial^2_{1,i}} = \frac{2}{\lambda}\left[\frac{1}{\sum_{j=0}^{n-1}(1-s_j)}\right]\left(\frac{P_1 P_2 \cdots P_{i-1}}{\lambda^{i-1}}\right)^2 Q^{\#}_{i,1}, \qquad i = 1, \ldots, n, \tag{3.14}$$

where $Q = I - A$ and A is as in (2.9) and (2.10). From this formula and the results of Theorem 3.1, it follows that if $\lambda > \max_{1 \le i \le n-1} P_i$, then

$$\frac{\partial^2 \lambda}{\partial^2_{1,i}} \ge 0 \Rightarrow \frac{\partial^2 \lambda}{\partial^2_{1,i}} \ge \frac{\partial^2 \lambda}{\partial^2_{1,j}}, \qquad j > i. \tag{3.15}$$

In particular, for λ as above,

$$\max_{1 \le i \le n} \frac{\partial^2 \lambda}{\partial^2_{1,i}} = \frac{\partial^2 \lambda}{\partial^2_{1,1}}. \tag{3.16}$$

Similarly, with respect to the subdiagonal entries, we have that

$$\frac{\partial^2 \lambda}{\partial^2_{k+1,k}} = \frac{2\lambda}{P_k^2} \frac{1 - s_k}{\sum_{j=0}^{n-1}(1-s_j)} Q^{\#}_{k,k+1}, \qquad k = 1, \ldots, n-1. \tag{3.17}$$

In particular, if $P_j \ge P_i$ and $j > i$, then

$$\frac{\partial^2 \lambda}{\partial^2_{i+1,i}} \ge 0 \Rightarrow \frac{\partial^2 \lambda}{\partial^2_{i+1,i}} \ge \frac{\partial^2 \lambda}{\partial^2_{j+1,j}}. \tag{3.18}$$

We now discuss the implications of the results obtained in this section on the population model that the Leslie matrix represents. First, we consider the qualitative interpretations. Theorems 3.1 and 3.3 show that the asymptotic rate of increase can only be a convex function of the fecundity and survival rates of younger age groups and that it is a concave function of these rates in older age groups. This suggests that only changes in the vital rates for younger age groups can yield a sharp change in the rate of increase of the asymptotic growth rate of the population. In attempting to compare the effects of changes in fecundity rates to changes in survival rates, we note that Theorem 3.1 always guarantees that for some younger age groups the asymptotic growth rate is a convex function of the fecundity. This is not necessarily the case for the effects of changes in the survival rates. Indeed, in our experience, it is difficult to produce examples where the asymptotic growth rate is a convex function of the survival rate of even the first age group. In this sense, the example given after Theorem 3.3 is atypical. It requires a peculiar fecundity distribution and, as we observe, a large number of age groups for changes in survival rates to have a sharp effect on the asymptotic growth rate.

Next, we consider the quantitative implications of our results on the population model. Recall that Demetrius' result [8, p. 134, Eq. (8)] asserts that if $\lambda > \max_{1\le i\le n-1} P_i$, then

$$\frac{\partial\lambda}{\partial_{1,i}} > \frac{\partial\lambda}{\partial_{1,j}}, \qquad j > i. \tag{3.19}$$

Our result in (3.15) reinforces (3.19) by showing that not only are the first partial derivatives of the asymptotic growth rate with respect to the fecundities ordered, but so are the second partial derivatives, at least for younger age groups. Next, we comment on what quantitative effects have changes in the rates of survival of the population on its growth rate? Here Demetrius' result [8, p. 134, Eq. (11)] is that

$$P_j \ge P_i \quad \text{and} \quad j > i \Rightarrow \frac{\partial\lambda}{\partial_{i+1,i}} \ge \frac{\partial\lambda}{\partial_{j+1,j}}. \tag{3.20}$$

Once again our result in (3.18) reinforces (3.20). Earlier we noted that the condition $\partial^2\lambda/\partial^2_{i+1,i} \ge 0$, which appears in (3.18), can only occur for younger age groups if at all. However, when this is the case, the hypothesis $P_j > P_i$ for $j > i$ seems reasonable when i is small because the survival rate for newborns is likely to be lower than that for slightly more mature individuals.

4. The Perron vector of a Leslie matrix as a function of top row and subdiagonal. We now examine the derivatives of the Perron vector of a stochastic Leslie matrix with respect to the entries in the top row and subdiagonal of the matrix. We then infer implications concerning the asymptotically stable age distribution vector of the general Leslie population model.

Let $\tilde{A}$ be an $n\times n$ general irreducible Leslie matrix whose Perron root is $\lambda = \lambda(\tilde{A})$. Throughout, its right Perron vector $x = x(\tilde{A})$ will be normalized so that its first entry is equal to 1. Denote by $y = y(\tilde{A})$ the left Perron vector of $\tilde{A}$ normalized so that $y^T x = 1$. Since now all the derivatives of the first entry of that Perron vector are zero, it will be convenient for us to work with the truncated form of $x = (1 \quad x_2 \ \dots \ x_n)^T$, viz., $\overline{x} = (x_2 \ \dots \ x_n)^T$. Before truncation all vectors that we work with in this section will be in $\mathbb{R}^n$ and a bar over them shall indicate their truncation to an $(n-1)$-dimensional vector by deleting their first entry. This notation is consistent with [10] and, for Leslie matrices, some results in that paper are also generalized here.

In the interest of convenience we derive a basic relation for the derivatives of the $\overline{x}$, which in essence are already contained in [10]. Put $\tilde{Q} = \lambda I - \tilde{A}$. On differentiating the matrix–vector relation $\tilde{A}x = \lambda x$ with respect to the (i,j)th entry, we obtain, on recalling (1.4), that

$$E_{i,j}x + \tilde{A}\frac{\partial x}{\partial_{i,j}} = x_j y_i x + \lambda \frac{\partial x}{\partial_{i,j}}$$

or

$$\frac{\partial \overline{x}}{\partial_{i,j}} = x_j N_{\tilde{Q}}^{-1}(\overline{e_i} - y_i \overline{x}), \tag{4.1}$$

where $N_{\tilde{Q}}$ is the $(n-1)\times(n-1)$ trailing principal submatrix of $\tilde{Q}$.

For the special case of a stochastic Leslie matrix we obtain the following.

LEMMA 4.1. *At the $n \times n$ irreducible stochastic Leslie matrix A whose top row is given by $(a_1 \cdots a_n)$,*

$$\frac{\partial \overline{x}}{\partial_{1,i}} = -\frac{1}{\sum_{j=0}^{n-1}(1-s_j)}(1 \quad 2 \quad \cdots \quad n-1)^T, \qquad 1 \le i \le n, \tag{4.2}$$

and

(4.3)

$$\frac{\partial \overline{x}}{\partial_{k+1,k}} = \frac{1}{\sum_{j=0}^{n-1}(1-s_j)} \begin{pmatrix} -(1-s_k) \\ -2(1-s_k) \\ \vdots \\ -(k-1)(1-s_k) \\ \sum_{j=0}^{n-1}(1-s_j) - k(1-s_k) \\ \vdots \\ \sum_{j=0}^{n-1}(1-s_j) - (n-1)(1-s_k) \end{pmatrix}, \qquad 1 \le k \le n-1,$$

where the s_i's are given in (2.19).

Proof. For the stochastic Leslie matrix A, we have that $x(A) = (1 \quad \cdots \quad 1)^T$ and for $Q = I - A$,

$$N_Q^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}.$$

Furthermore, if $y = y(A)$ is the left Perron vector of A normalized as above, then

$$y^T = \frac{1}{\sum_{j=0}^{n-1}(1-s_j)}((1-s_0) \quad (1-s_1) \quad \cdots \quad (1-s_{n-1})).$$

The expression for $\partial\overline{x}/\partial_{1,i}$ now follows on substituting these values of x, y, and N_Q^{-1} into (4.1) and on noting that $\overline{e_1} = 0$. The expression for $\partial\overline{x}/\partial_{k+1,k}$ follows similarly. □

Considering (4.2), we see that each entry of $\partial\bar{x}/\partial_{1,i}$ is negative for each $i = 1,\dots,n$. This fact is actually a consequence of a more general result of Elsner, Johnson, and Neumann [12, Thm. 1]. In contrast to (4.2), (4.3) shows that the signs of the entries of the $\partial\bar{x}/\partial_{k+1,k}$ are not necessarily uniform. Clearly, the first $k-1$ entries are negative and the remaining ones form a decreasing sequence. The following argument shows that $\partial\bar{x}/\partial_{k+1,k}$ always has at least one positive entry. Note that

$$\sum_{j=0}^{n-1}(1-s_j)-(k+1)(1-s_k)=\sum_{j=0}^{k}(s_k-s_j)+\sum_{j=k+1}^{n-1}(1-s_j)>0,\qquad 1\le k\le n-2.$$

Thus, we see that the kth and $(k+1)$th entries of $\partial\bar{x}/\partial_{k+1,k}$ are always positive for $1\le k\le n-2$. A similar argument shows that the last entry of $\partial\bar{x}/\partial_{n,n-1}$ is also always positive. From the above we see that either the entries of $\partial\bar{x}/\partial_{k+1,k}$ in positions k through $n-1$ are all nonnegative or there is an index $i_0\ge k+1$ such that the entries in positions k through i_0 are nonnegative and the remaining entries are nonpositive. To see that each such sign pattern can be realized, consider the stochastic Leslie matrix whose top row has $0<\alpha<1$ in the $(k+1)$th position and $1-\alpha$ in the nth position. It is readily verified that each admissible sign pattern can be obtained by a suitable choice of α.

We next investigate the sign pattern of the second partial derivatives of $\bar{x}$ with respect to the entries of the top row of a stochastic Leslie matrix.

THEOREM 4.2. *Let A be an $n\times n$ irreducible stochastic Leslie matrix whose top row is given by $(a_1\ \cdots\ a_n)$ and let $x=x(A)$ be its right Perron vector normalized so that its first entry is 1. Then*

$$\left(\frac{\partial^2\bar{x}}{\partial_{1,i}^2}\right)_l=2l\left(\frac{1}{\sum_{j=0}^{n-1}(1-s_j)}\right)^2\left[\frac{l+1}{2}+i-\frac{\sum_{j=0}^{n-1}(1-s_j)(j+1)}{\sum_{j=0}^{n-1}(1-s_j)}\right]\tag{4.4}$$

for all $1\le l\le n-1$ and $1\le i\le n$ and where the s_i's are as specified in (2.19). *Furthermore, when i is fixed, there is at most one sign change in the entries of $\partial^2\bar{x}/\partial_{1,i}^2$ from minus to plus as l increases. Similarly, when l is fixed and i is increased, there is at most one sign change in the sequence formed from the lth entries of the vectors $\partial^2\bar{x}/\partial_{1,i}^2, i=1,\dots,n$. In particular,*

$$\frac{l+1}{2}+i\ge\frac{n+1}{2}\Rightarrow\left(\frac{\partial^2\bar{x}}{\partial_{1,i}^2}\right)_l\ge 0\tag{4.5}$$

and an all pluses sign pattern is possible when either i or l is sufficiently large.

Proof. As usual let $Q=I-A$ and let y be the left Perron vector of A normalized so that $y^Tx=1$. From formula (4.13) in [10] we find that

$$\frac{\partial^2\bar{x}}{\partial_{1,i}^2}=2y_1x_i\left(y_1x_iN_Q^{-1}-Q_{i,1}^{\#}I\right)N_Q^{-1}\bar{x},\qquad i=1,\dots,n.\tag{4.6}$$

Substituting in the expressions for x, y, and N_Q^{-1} developed in Lemma 4.1 and using the formula for $Q_{i,1}^{\#}$ given in (3.1) yield after some simplification (4.4). The claims concerning the sign patterns follow readily by inspecting (4.4). That (4.5) holds can be established by the same argument given in Theorem 3.1, which was used to show that $Q_{i,1}^{\#}\le 0$ for $i\ge(n+1)/2$. □

As in our remarks following Theorem 3.1 concerning the sign pattern of the entries of the first column of $Q^{\#}$, examples can be constructed to show that when i is fixed and l is increasing, the switch from minus to nonnegative in $(\partial^2\overline{x}/\partial_{1,i}^2)_l$ can occur at any index $l_0 < n-2i$. A similar remark holds for the sequence $\partial^2\overline{x}/\partial_{1,i}^2$ when l is fixed and i is increasing.

We now return to a general Leslie matrix $\tilde{A}$. From (2.6) we have that for the first partial derivatives of the Perron vector with respect to the top row and subdiagonal,

$$\frac{\partial\overline{x(\tilde{A})}}{\partial_{1,i}} = \frac{P_1P_2\cdots P_{i-1}}{\lambda^i} N_D \frac{\partial\overline{x(A)}}{\partial_{1,i}}, \qquad i = 1,\ldots,n, \tag{4.7}$$

and

$$\frac{\partial\overline{x(\tilde{A})}}{\partial_{k+1,k}} = \frac{1}{P_k} N_D \frac{\partial\overline{x(A)}}{\partial_{k+1,k}}, \qquad k = 1,\ldots,n-1, \tag{4.8}$$

where A is given by (2.9) and (2.10). Similarly, for the second partial derivatives of the Perron vector with respect to the first row, we have from (2.7) that

$$\frac{\partial^2\overline{x(\tilde{A})}}{\partial_{1,i}^2} = \left[\frac{1}{\sum_{j=0}^{n-1}(1-s_j)}\right]\left(\frac{P_1P_2\cdots P_{i-1}}{\lambda^i}\right)^2 N_D \frac{\partial^2\overline{x(A)}}{\partial_{1,i}^2} \tag{4.9}$$

for all $i = 1,\ldots,n$. From (4.8) we can conclude that

(4.10)

$$\lambda \geq \max_{1\leq i\leq n-1} P_i \quad \text{and} \quad \left(\frac{\partial\overline{x(\tilde{A})}}{\partial_{k+1,k}}\right)_j \geq 0 \Rightarrow \left(\frac{\partial\overline{x(\tilde{A})}}{\partial_{k+1,k}}\right)_j \geq \left(\frac{\partial\overline{x(\tilde{A})}}{\partial_{k+1,k}}\right)_l, \qquad l \geq j.$$

In particular,

$$\max_{1\leq j\leq n-1}\left(\frac{\partial\overline{x(\tilde{A})}}{\partial_{k+1,k}}\right)_j = \left(\frac{\partial\overline{x(\tilde{A})}}{\partial_{k+1,k}}\right)_k. \tag{4.11}$$

We come now to interpret our results on the Perron vector for the population model. We begin with qualitative observations. For our analysis of the behavior of the asymptotically stable age distribution vector to make sense, we must choose a frame of reference, and the one we select is to compare the size of any age group to the size of the first age group.

(i) Formula (4.2) in Lemma 4.1 implies that raising the fecundity of any age group decreases the ratio of the size of any age group beyond the first to the size of the first age group.

(ii) Formula (4.3) in Lemma 4.1 shows that raising the survival rate of the kth age group has the effect of increasing the ratio of the size of the $(k+1)$th age group to the size of the first age group and possibly of raising those ratios for subsequent age groups while diminishing the those ratios for age groups $2,\ldots,k$.

In the way of quantitative interpretation of our results on the Perron vector for the population model, the only definite conclusion that we can draw follows from (4.3) and (4.11). Here we see that increasing the survival rate at the kth age group has the

effect of raising the ratios of the sizes of some age groups corresponding to ages other than 1 to the size of the group at age 1 while decreasing the ratios of sizes of other age groups to that of the size of the group at age 1. However, when $\lambda \geq \max_{1 \leq i \leq n-1} P_i$, it follows from (4.8) that the ratio that increases the most corresponds to the $(k+1)$th age group.

Acknowledgments. This research was conducted while the first author was a Postdoctoral Fellow at the Institute for Mathematics and its Applications of the University of Minnesota and while the second author was a visitor there. Both authors thank the IMA and its staff for their support.

REFERENCES

[1] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*, Academic Press, New York, 1973.

[2] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[3] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Dover Publications, New York, 1991.

[4] H. Caswell, *Matrix Population Models*, Sinauer, Sunderland, MA, 1989.

[5] J. E. Cohen, *Derivatives of the spectral radius as a function of a nonnegative matrix*, Math. Proc. Cambridge Philos. Soc., 83 (1978), pp. 183–190.

[6] ———, *Random evolutions and the spectral radius of a nonnegative matrix elements*, Math. Proc. Cambridge Philos. Soc., 86 (1979), pp. 343–350.

[7] ———, *Convexity of the dominant eigenvalue of an essentially non-negative matrix*, Proc. Amer. Math. Soc., 81 (1981), pp. 657–658.

[8] L. Demetrius, *The sensitivity of population growth rate to perturbations in the life cycle components*, Math. Biosci., 4 (1969), pp. 129–136.

[9] 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.

[10] ———, *On the first and second order derivatives of the Perron vector*, Linear Algebra Appl., 71 (1985), pp. 57–76.

[11] L. Elsner, *On convexity properties of the spectral radius of nonnegative matrices*, Linear Algebra Appl., 61 (1984), pp. 31–35.

[12] L. Elsner, C. R. Johnson, and M. Neumann, *On the effect of the perturbation of a nonnegative matrix on its Perron eigenvector*, Czechoslovak. Math. J., 32 (1982), pp. 99–109.

[13] S. Friedland, *Convex spectral functions*, Linear and Multilinear Algebra, 9 (1981), pp. 299–316.

[14] L. A. Goodman, *On the sensitivity of the intrinsic growth rate to changes in the age-specific birth and death rates*, Theoret. Population Biol., 2 (1971), pp. 339–354.

[15] G. H. Golub and C. D. Meyer, Jr., *Using the QR-factorization and group inversion to compute, differentiate, and estimate the sensitivity of stationary probabilities for Markov chains*, SIAM J. Alg. Discrete Methods, 7 (1986), pp. 273–281.

[16] M. Haviv, Y. Ritov, and U. G. Rothblum, *Taylor expansions of eigenvalues of perturbed matrices with applications to spectral radii of nonnegative matrices*, Linear Algebra Appl., 168 (1992), pp. 159–188.

[17] R. Lal and D. H. Anderson, *Calculation and utilization of component matrices in linear bioscience models*, Math. Biosci., 99 (1990), pp. 11–29.

[18] C. D. Meyer, Jr., *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Rev., 17 (1975), pp. 443–464.

[19] C. D. Meyer, Jr. and G. W. Stewart, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25 (1988), pp. 679–691.

[20] J. H. Pollard, *Mathematical Models for the Growth of Human Populations*, Cambridge University Press, Cambridge, 1973.

[21] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[22] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1108–1131, October 1994

005

STABLE NUMERICAL ALGORITHMS FOR EQUILIBRIUM SYSTEMS*

STEPHEN A. VAVASIS†

Abstract. An equilibrium system (also known as a Karush–Kuhn–Tucker (KKT) system, a saddlepoint system, or a sparse tableau) is a square linear system with a certain structure. Strang [*SIAM Rev.*, 30 (1988), pp. 283–297] has observed that equilibrium systems arise in optimization, finite elements, structural analysis, and electrical networks. Recently, Stewart [*Linear Algebra Appl.*, 112 (1989), pp. 189–193] established a norm bound for a type of equilibrium system in the case when the "stiffness" portion of the system is very ill-conditioned. This paper investigates the algorithmic implications of Stewart's result. It is shown that several algorithms for equilibrium systems appearing in applications textbooks are unstable. A certain hybrid method is then proposed, and it is proved that the new method has the right stability property.

Key words. equilibrium systems, KKT systems, stable algorithms, linear algebra

AMS subject classifications. 65F05, 65F35, 65G05

1. Equilibrium systems.

Recently, Strang [21] observed that the problem of solving the structured linear system

$$\begin{pmatrix} D & -A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix} \tag{1}$$

for $\mathbf{x}$ and $\mathbf{y}$ arises in many physical applications. Here, D is an $m \times m$ matrix, A is an $m \times n$ matrix, $\mathbf{x}$ and $\mathbf{b}$ are m-vectors, and $\mathbf{y}$ and $\mathbf{c}$ are n-vectors. We call this system an *equilibrium system.*

The focus of this paper is on the stability of algorithms for this linear system. Table 1 summarizes the applications and the interpretations of D, A, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{b}$, and $\mathbf{c}$ in these applications. In the case of finite elements, we have indicated the interpretations in the context of a heat equilibrium problem (that is, $\nabla \cdot (g(x,y)\nabla u) = -f(x,y)$ on Ω, $u = 0$ on $\partial\Omega$, where Ω is a suitable domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, and $f(x,y)$ and $g(x,y)$ are given functions, g being strictly positive).

The reader will observe that there are some similarities among the various interpretations. For example, one similarity is that, for the three physical applications, $\mathbf{x}$ and $\mathbf{c}$ are measured in the same physical units (e.g., amps), as are $\mathbf{y}$ and $\mathbf{b}$ (e.g., volts). This is also the case in certain optimization problems such as flow problems. This observation has some importance for numerical algorithms.

In all of these applications, the following two assumptions are commonplace and they are made throughout the paper.

A1. Matrix D is symmetric and positive definite.

A2. Matrix A has rank n, i.e., full column rank.

These assumptions imply that (1) is a nonsingular linear system with a unique solution.

* Received by the editors May 13, 1992; accepted for publication (in revised form) March 18, 1993. This research was supported in part by a Presidential Young Investigator Award with matching funds received from AT&T and Xerox and by the the National Science Foundation, the Air Force Office of Scientific Research, and the Office of Naval Research through NSF grant DMS-8920550.

† Department of Computer Science, Cornell University, Ithaca, New York 14853 (vavasis@cs.cornell.edu).

TABLE 1
The interpretations of the variables in (1) *for various applications.*

Variable	Optimization	Electrical networks	Finite elements	Structures
D	Objective function Hessian	Resistances	Thermal resistance	Element flexibility
A	Constraints	Node-wire adjacency	Node-element geometry	Node-element geometry
$\mathbf{x}$	Primal variables	Currents	Element heat flow	Element forces
$\mathbf{y}$	Lagrange multipliers	Voltages	Nodal temperatures	Nodal displacements
$\mathbf{b}$	Objective function gradient	Voltage sources	Applied temperature gradient	Element dilations
$\mathbf{c}$	Distance from constraints	Current sources	Applied heat sources	Applied nodal forces

The main focus of this paper is what happens when D is severely ill-conditioned. Björck [1] calls this case the "stiff" case, in analogy with systems of ordinary differential equations. Where D is well conditioned, the numerical problems associated with solving (1) are generally not as troublesome, and most standard methods will give good answers. Thus, we make the following assumption.

A3. Matrix D is very ill conditioned.

The most natural framework for this assumption is an optimization algorithm involving a barrier function. The primary example of a barrier function in optimization is the class of interior point methods for linear programming. In an interior point method, matrix D becomes very ill-conditioned when the iterate approaches the boundary of the feasible region. (See Karmarkar [13] for the first interior-point method proposed for linear programming. See Wright [26] for a description of barrier methods and linear programming and their relationship.) For linear programming, since the solution is always on the boundary of the region, ill-conditioning in D *always* occurs during the algorithm.

Ill-conditioning of D can also occur in the other three applications listed above. In electrical networks, ill-conditioning occurs, for example, when one wishes to model the leakage currents to ground through insulators in the electrical network. In this case, some of the resistances are many orders of magnitude larger than others, leading to a very ill-conditioned D. The same ill-conditioning occurs if the linear equilibrium system is a model of a time-varying system in which certain circuit elements have switched "off."

In structural analysis, ill-conditioning in D occurs when the elements of the structure have widely different rigidity properties. In the finite elements for a heat application, ill-conditioning in D would occur if part of the domain under analysis were a thermal insulator and another part were a thermal conductor.

In the case that D is severely ill-conditioned, it is not at all apparent that an accurate numerical solution to (1) is possible, since ill-conditioning in D presumably means extreme sensitivity to roundoff errors. A perturbation theorem developed in §2, based on a recent theorem due to Stewart [20], shows that, in principle, an accurate solution can be computed under further assumptions. The perturbation result in §2 leads us to propose a definition of "stable" algorithms. None of the standard algorithms are stable when compared with the perturbation bound, as we

demonstrate in §4. There is a possibility, however, that some recent techniques in the literature might lead to stable algorithms—this possibility is discussed in §5.

Our main contribution, aside from the new definition of "stable," is a new algorithm described in §§6 and 7 that can accurately solve (1).

Most of the analysis of this paper is valid only under certain further assumptions that we now state.

A4. We are more interested in recovering $\mathbf{y}$ in (1) rather than $\mathbf{x}$.

A5. In (1) the vector $\mathbf{c}$ is zero.

A6. Matrix D in (1) is a diagonal matrix.

In §9 we discuss how A4, A5, and A6 could be relaxed.

Assumption A4 is similar to obtaining error bounds on individual components of the solution of a linear system. See, for example, Chandrasekaran and Ipsen [3]. In our work, the objective is to obtain a bound for a block of components of the solution.

In the context of optimization, A5 means that the current point is feasible. In the context of electrical networks, this assumption means that no external current sources are applied—only voltage sources (e.g., batteries or generators).

Assumption A6 holds in the context of interior point methods for linear programming. It also holds for electrical networks composed of batteries and resistors. It holds for finite elements for the varying-coefficient Poisson equation above, provided A is correctly formed. It does not hold for structural analysis in the general case; one usually obtains a matrix D that is block diagonal. There are, however, special cases of structural analysis where D is diagonal that we will review in future work.

A fifth application of the equilibrium system not listed in Table 1 is a discretization of Stokes flow. Stokes flow is a linear approximation to the Navier–Stokes equations for incompressible fluid flow under the assumption of a very low Reynolds number. Assumption A6 is never satisfied for Stokes flow (even in one dimension), which is why we do not discuss it further.

2. Stewart's norm-bound result. Under the assumptions made in §1, we can apply Stewart's theorem. Because of A4, it is useful to write an explicit formula for $\mathbf{y}$ in terms of the other variables. This formula is obtained by eliminating $\mathbf{x}$ from (1) and also substituting A5:

$$A^T D^{-1} A \mathbf{y} = -A^T D^{-1} \mathbf{b}. \tag{2}$$

Because of A1 and A2, the matrix $A^T D^{-1} A$ is invertible. Thus, this linear system uniquely determines $\mathbf{y}$:

$$\mathbf{y} = -(A^T D^{-1} A)^{-1} A^T D^{-1} \mathbf{b}. \tag{3}$$

We are now in a position to state Stewart's theorem. For the sake of completeness, we replicate his proof of the theorem because we refer to some of the proof steps in our algorithm construction.

THEOREM 2.1 ([20]). *Let $\mathcal{D}$ denote the set of all positive definite $m \times m$ real diagonal matrices. Let A be an $m \times n$ real matrix of* rank n. *Then there exist constants χ_A and $\bar{\chi}_A$ such that for any $D \in \mathcal{D}$,*

(a) $\|(A^T D^{-1} A)^{-1} A^T D^{-1}\| \leq \chi_A$, *and*

(b) $\|A(A^T D^{-1} A)^{-1} A^T D^{-1}\| \leq \bar{\chi}_A$.

In this theorem, we have assumed the use of some matrix norm $\|\cdot\|$ that is induced by a vector norm (also denoted $\|\cdot\|$).

A closely related result was obtained independently by Todd [23] as part of an analysis of Karmarkar's algorithm. In particular, Todd shows that

$$\|(A^T D^{-1} A)^{-1} A^T D^{-1} \mathbf{b}\| \leq c(A, \mathbf{b}).$$

This bound can be extended to obtain Theorem 2.1 with a compactness argument involving $\mathbf{b}$. Todd's proof is geometric and completely different from Stewart's proof. We follow Stewart's proof because it contains algebraic information useful for further development.

Before beginning the proof, we discuss the interpretation of the theorem. Except for sign, the matrix in (a) is the same matrix that is applied to $\mathbf{b}$ to obtain $\mathbf{y}$ according to (3). Thus, (a) says that $\mathbf{y}$ cannot be much larger than $\mathbf{b}$, no matter how D is chosen. In the context of electrical networks, this has a very natural physical interpretation: For a fixed electrical network with no current sources, no matter how badly the resistors are out of scale, there can never be a voltage level in the circuit that is much higher than the applied battery voltages.

Statement (b) also has an algorithmic interpretation. Specifically, if we know $\mathbf{y}$ and are trying to obtain $\mathbf{x}$, we observe from (1) that

$$D\mathbf{x} - A\mathbf{y} = \mathbf{b}.$$

The second term on the left-hand side is precisely

$$A(A^T D^{-1} A)^{-1} A^T D^{-1} \mathbf{b}.$$

Thus, the matrix in (b) can occur in the process of recovering $\mathbf{x}$.

Proof ([20]). First, we observe that statements (a) and (b) imply each other. For example, if we could prove (a), then we would observe that

$$\|A(A^T D^{-1} A)^{-1} A^T D^{-1}\| \leq \|A\| \cdot \|(A^T D^{-1} A)^{-1} A^T D^{-1}\|,$$

and thus we could take $\bar{\chi}_A$ to be $\|A\| \cdot \chi_A$. Similarly, the following inequality allows us to derive (a) from (b):

$$\|(A^T D^{-1} A)^{-1} A^T D^{-1}\| \leq \|(A^T A)^{-1} A^T\| \cdot \|A(A^T D^{-1} A)^{-1} A^T D^{-1}\|.$$

Thus, we prove (b) only. We first need a preliminary lemma, which is also due to Stewart. This lemma is cited later in the paper.

LEMMA 2.2 ([20]). *Define two subsets X and Y of $\mathbb{R}^m$ as follows*:

$$X = \{\mathbf{z} : \mathbf{z} = A\mathbf{w} \text{ for some } \mathbf{w}, \text{ and } \|\mathbf{z}\| = 1\}$$

and

$$Y = \{\mathbf{z} : A^T D^{-1} \mathbf{z} = \mathbf{0} \text{ for some } D \in \mathcal{D}\}.$$

Then $X \cap \bar{Y} = \emptyset$.

Here $\bar{Y}$ denotes the closure of Y in the topological sense.

Proof ([20]). Suppose there were a $\mathbf{z} \in X \cap \bar{Y}$, so that $\mathbf{z} = A\mathbf{w}$ for some $\mathbf{w}$ and $\|\mathbf{z}\| = 1$. The statement that $\mathbf{z}$ is in $\bar{Y}$ means that there exists an infinite sequence of vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots$ converging to $\mathbf{z}$ and an infinite sequence of matrices $D_1, D_2, \ldots$ in $\mathcal{D}$ such that $A^T D_k^{-1} \mathbf{z}_k = \mathbf{0}$ for all k. Taking the inner product of this equation with $\mathbf{w}$, and substituting $\mathbf{w}^T A^T = \mathbf{z}^T$, yields $\mathbf{z}^T D_k^{-1} \mathbf{z}_k = 0$ for all k. But since $\mathbf{z}_k$ converges

to $\mathbf{z}$, there is some k large enough such that for all nonzero coordinate entries of $\mathbf{z}$, the corresponding entry of $\mathbf{z}_k$ has the same sign. This means that $\mathbf{z}^T D_k^{-1} \mathbf{z}_k > 0$, contradicting the previous equation. □

Now we use the lemma to conclude the proof of Theorem 2.1. Since X is compact and $\bar{Y}$ is closed, and since they are disjoint, there is a positive lower bound, say ρ, on the distance between these spaces measured in the $\|\cdot\|$ norm.

Next, choose some $D \in \mathcal{D}$. Choose an arbitrary vector $\mathbf{x}$. Let

$$\mathbf{y} = A(A^T D^{-1} A)^{-1} A^T D^{-1} \mathbf{x}.$$

The goal is to get an upper bound on $\|\mathbf{y}\|$ in terms of $\|\mathbf{x}\|$. A one-line calculation shows that $A^T D^{-1}(\mathbf{x} - \mathbf{y})$ is zero. Thus, let $\mathbf{v} = \mathbf{x} - \mathbf{y}$ so that $A^T D^{-1} \mathbf{v} = \mathbf{0}$. Set $t = 1/\|\mathbf{y}\|$. Then we have

$$t\mathbf{v} + t\mathbf{y} = t\mathbf{x}.$$

Note that $t\mathbf{v}$ is in Y and $-t\mathbf{y}$ is in X. Thus, the norm of the left-hand side of this equation is at least ρ. Thus, $\|t\mathbf{x}\| \geq \rho$, i.e.,

$$\rho \|\mathbf{y}\| \leq \|\mathbf{x}\|.$$

This gives us the upper bound of $1/\rho$ on $\bar{\chi}_A$ and concludes the proof. □

For the rest of the paper, given a matrix A, we define χ_A and $\bar{\chi}_A$ to be the suprema implicit in Theorem 2.1, i.e.,

$$\chi_A = \sup\{\|(A^T D^{-1} A)^{-1} A^T D^{-1}\| : D \in \mathcal{D}\}$$

and

$$\bar{\chi}_A = \sup\{\|A(A^T D^{-1} A)^{-1} A^T D^{-1}\| : D \in \mathcal{D}\}.$$

Stewart gives a bound for $\bar{\chi}_A$ that was proved by O'Leary [14] to be an exact formula. The formula seems to require an exponential number of steps to compute $\bar{\chi}_A$. Moreover, this formula is not completely constructive in finite precision arithmetic because of Stewart's observation that the parameters χ_A and $\bar{\chi}_A$ in Theorem 2.1 do not depend continuously on A. In particular, Stewart shows that $\bar{\chi}_A$ for $A = [0, 1]^T$ is one, but $\bar{\chi}_A$ for $A = [\epsilon, 1]^T$ tends to infinity as ϵ tends to zero.

It would be very interesting if there were an algorithm to compute or approximate $\chi_A, \bar{\chi}_A$ overcoming either of these difficulties (i.e., the algorithm gives the right answer in finite-precision arithmetic or the algorithm requires only a polynomial number of steps or both). In §8, we give a bound for $\chi_A, \bar{\chi}_A$ for a special class of matrices.

One would like to apply Theorem 2.1 to give algorithm stability results for finite-precision arithmetic. Here is an example of a straightforward perturbation result in this direction. For this theorem, $\|\cdot\|$ is restricted to being a p-norm. (In a p-norm, we know that the norm of a diagonal matrix is equal to its maximum absolute diagonal entry.)

THEOREM 2.3. *Let* $\mathbf{y}$ *be the second component of the exact solution to* (1) (*when* $\mathbf{c} = \mathbf{0}$). *Let* $\hat{\mathbf{y}}$ *be a computed solution to this system, such that* $\hat{\mathbf{y}}$ *is the exact solution to the perturbed system*

$$\begin{pmatrix} D+E & -A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \mathbf{b}+\mathbf{e} \\ \mathbf{0} \end{pmatrix}, \tag{4}$$

where E satisfies the bound $|E| \leq \epsilon \cdot |D|$ and $\|\mathbf{e}\| \leq \epsilon\|\mathbf{b}\|$, where $\epsilon < \frac{1}{3}$. Then

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \leq \chi_A \epsilon \cdot \|\mathbf{b}\| \cdot (\bar{\chi}_A + 8/3). \tag{5}$$

The novel property of this theorem is that the error bound in (5) is independent of the condition number of D or of the condition number of (1).

Note that we have used the matrix absolute value notation from Golub and Van Loan [11]. Thus, the assumption $|E| \leq \epsilon \cdot |D|$ implies that E is also diagonal.

Proof. By elimination as described above, we have the following formulas for $\mathbf{y}$ and $\hat{\mathbf{y}}$:

$$A^T D^{-1} A\mathbf{y} = -A^T D^{-1}\mathbf{b}$$

and

$$A^T D^{-1} A\hat{\mathbf{y}} + A^T F A\hat{\mathbf{y}} = -A^T D^{-1}(\mathbf{b} + \mathbf{e}) - A^T F(\mathbf{b} + \mathbf{e}).$$

Here F denotes $(D+E)^{-1} - D^{-1}$, i.e., $-ED^{-1}(D+E)^{-1}$. Subtracting these equations yields

$$A^T D^{-1} A(\hat{\mathbf{y}} - \mathbf{y}) + A^T F A\hat{\mathbf{y}} = -A^T D^{-1}\mathbf{e} - A^T F(\mathbf{b} + \mathbf{e}).$$

We can add and subtract $A^T F A\mathbf{y}$ on the left to obtain

$$A^T D^{-1} A(\hat{\mathbf{y}} - \mathbf{y}) + A^T F A(\hat{\mathbf{y}} - \mathbf{y}) + A^T F A\mathbf{y} = -A^T D^{-1}\mathbf{e} - A^T F(\mathbf{b} + \mathbf{e}),$$

which can be written

$$A^T (D^{-1} + F) A(\hat{\mathbf{y}} - \mathbf{y}) = -A^T F A\mathbf{y} - A^T D^{-1}\mathbf{e} - A^T F(\mathbf{b} + \mathbf{e}).$$

Define $G = D^{-1} + F = (D + E)^{-1}$, another diagonal matrix. Introduce new vectors $\mathbf{e}_1 = G^{-1} F A\mathbf{y}$, $\mathbf{e}_2 = G^{-1} D^{-1}\mathbf{e}$, and $\mathbf{e}_3 = G^{-1} F(\mathbf{b} + \mathbf{e})$. Then we can rewrite the preceding equation as

$$A^T G A(\hat{\mathbf{y}} - \mathbf{y}) = -A^T G(\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3).$$

Now we can conclude from Theorem 2.1 that

$$\|\hat{\mathbf{y}} - \mathbf{y}\| \leq \chi_A(\|\mathbf{e}_1\| + \|\mathbf{e}_2\| + \|\mathbf{e}_3\|).$$

Thus, the theorem is proved once we analyze the three terms on the right-hand side. First, we have

$$\begin{aligned} \|\mathbf{e}_1\| &= \|(D + E) F A\mathbf{y}\| \\ &\leq \|E D^{-1}\| \cdot \|A\mathbf{y}\| \\ &\leq \epsilon \|A\mathbf{y}\| \\ &\leq \epsilon \cdot \bar{\chi}_A \cdot \|\mathbf{b}\|. \end{aligned}$$

Next, we have

$$\begin{aligned} \|\mathbf{e}_2\| &= \|G^{-1} D^{-1}\mathbf{e}\| \\ &= \|(D + E) D^{-1}\mathbf{e}\| \\ &\leq \|(D + E) D^{-1}\| \cdot \|\mathbf{e}\| \\ &\leq (1 + \epsilon) \cdot \epsilon \cdot \|\mathbf{b}\|. \end{aligned}$$

Note that $1+\epsilon \leq 4/3$. Finally,

$$\begin{aligned}\|\mathbf{e}_3\| &= \|G^{-1}F(\mathbf{b}+\mathbf{e})\| \\ &\leq \epsilon \cdot (1+\epsilon) \cdot \|\mathbf{b}\|.\end{aligned}$$

As above, $1+\epsilon \leq 4/3$. This proves the theorem. □

For the rest of the paper, we regard (5) as the "ideal" bound that could be satisfied by a finite-precision algorithm. The stability result of this theorem is somewhat different from well-known perturbation results. Usually a bound on the *relative* error in $\mathbf{y}$ is obtained in terms of ϵ and the other parameters. A relative error bound on $\mathbf{y}$ is not possible for the equation at hand. Such a bound would mean that it is not possible for $\|\mathbf{y}-\hat{\mathbf{y}}\|$ to be large when $\|\mathbf{y}\|$ is small. However, one can select a nonzero $\mathbf{b}$ to make $\mathbf{y}$ arbitrarily small—zero in fact—because $\mathbf{b}$ generally has more dimensions than $\mathbf{y}$. Thus, the bound in the theorem seems to be the best form we could hope for.

We say that an algorithm for (1) is *stable* if, in the presence of finite-precision arithmetic, an error bound of the same form as (5) is satisfied, where ϵ is on the order of the machine roundoff. In particular, the error bound should have the form

$$\|\mathbf{y}-\hat{\mathbf{y}}\| \leq \epsilon \cdot f(A) \cdot \|\mathbf{b}\|,$$

where $f(A)$ is some function of A not depending on D.

If we could show that a finite-precision algorithm satisfied the hypothesis of the theorem (i.e., the computed solution satisfied (4)), then it would automatically be stable. Unfortunately, we do not know of any finite-precision algorithm to solve (1) that achieves the conditions of this theorem. In particular, it seems that the error bound for any obvious algorithm would involve a backward error term affecting not only D but also A. Since χ_A is not continuous with respect to changes in A, the theorem cannot be extended to the case when A is also perturbed. In §6 we derive a stable algorithm by other means.

3. Further development of the theory. In this section, we develop the theory further before turning to algorithms in the next section. First, we make a few remarks about the difference between χ_A and $\bar{\chi}_A$. Then we look at the relationship between $\chi_A, \bar{\chi}_A$ and the corresponding variables for the *dual* problem.

The two parameters $\chi_A, \bar{\chi}_A$ have different properties with respect to changes in A. If A is multiplied by a constant, then χ_A scales in a reciprocal manner. Parameter $\bar{\chi}_A$ is unchanged when A is scaled. It should be apparent from (5) that the ideal situation is when both constants are small. Thus, we think of A as "well conditioned" if $\bar{\chi}_A$ is on the order of unity and χ_A is on the order of $1/\|A\|$. Note that $\bar{\chi}_A$ is at least one in any norm; this is seen by taking D to be the identity matrix in Theorem 2.1(b), in which case $A(A^TA)^{-1}A^T$ is an orthogonal projection matrix with spectral radius of one.

In the preceding paragraph we observed that $\bar{\chi}_A$ is unchanged by scaling. More generally, one sees by inspection that $\bar{\chi}_A$ remains unchanged if A is replaced by AR where R is any nonsingular $n \times n$ matrix. This is the same as saying that $\bar{\chi}_A$ depends only on the span of the columns of A and not the particular basis selected for that span. The parameter χ_A, on the other hand, depends also on the degree of linear independence of the columns of A.

In the proof in §2, we bounded $\bar{\chi}_A$ in terms of $1/\rho$. We now extend that result to show that this bound is an equation: the supremum of $\|A(A^TD^{-1}A)^{-1}A^TD^{-1}\|$ taken over $D \in \mathcal{D}$ is exactly $1/\rho$. We already proved that this supremum is at most

$1/\rho$. The argument for the other direction is from O'Leary [14]. Observe that if $\mathbf{x} \in X$ and $\mathbf{y} \in Y$, where X, Y are as in Lemma 2.2, then the following argument puts a lower bound on $\mathbf{x} - \mathbf{y}$. Let D be chosen so that $A^T D^{-1}\mathbf{y} = \mathbf{0}$, and let $\mathbf{w}$ be chosen so that $\mathbf{x} = A\mathbf{w}$. Then

$$A^T D^{-1}(\mathbf{x} - \mathbf{y}) = A^T D^{-1}\mathbf{x} = A^T D^{-1} A\mathbf{w},$$

so

$$\mathbf{w} = (A^T D^{-1} A)^{-1} A^T D^{-1}(\mathbf{x} - \mathbf{y}).$$

This yields

$$\mathbf{x} = A\mathbf{w} = A(A^T D^{-1} A)^{-1} A^T D^{-1}(\mathbf{x} - \mathbf{y}),$$

so

$$\|\mathbf{x}\| \leq \bar{\chi}_A \cdot \|\mathbf{x} - \mathbf{y}\|.$$

But $\|\mathbf{x}\| = 1$ by the assumption that $\mathbf{x} \in X$, so we see that $1/\bar{\chi}_A$ is a lower bound on $\|\mathbf{x} - \mathbf{y}\|$.

We now look into the relationship between (1) and its dual problem. Let Z be a *nullspace basis* for A^T; that is, an $m \times (m - n)$ matrix Z of full column rank such that $A^T Z = 0$. We want to know the relationship between $\bar{\chi}_A$ and $\bar{\chi}_Z$. Note that A is a nullspace basis for Z^T, so any relationship between $\bar{\chi}_A, \bar{\chi}_Z$ also holds if A, Z are interchanged. Note also that $\bar{\chi}_Z$ depends only on the span of the columns of Z: in other words, it does not matter which nullspace basis Z is selected.

To obtain a bound for $\bar{\chi}_Z$, we consider two algorithms for solving the following version of (1), where $\mathbf{x}_0$ is some arbitrary m-vector.

$$\begin{pmatrix} D & -A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ A^T \mathbf{x}_0 \end{pmatrix}. \tag{6}$$

First, observe that the second equation of (6) is $A^T\mathbf{x} = A^T\mathbf{x}_0$, i.e., $A^T(\mathbf{x} - \mathbf{x}_0) = \mathbf{0}$. This is the same as saying that $\mathbf{x} - \mathbf{x}_0$ is in the nullspace of A, i.e., there is an $m - n$-vector $\mathbf{q}$ such that $\mathbf{x} = \mathbf{x}_0 + Z\mathbf{q}$. Substituting this into the first equation yields $D(\mathbf{x}_0 + Z\mathbf{q}) - A\mathbf{y} = \mathbf{0}$. Multiply by Z^T using the fact that $Z^T A = 0$ to obtain $Z^T D Z\mathbf{q} = -Z^T D\mathbf{x}_0$. Solving for $\mathbf{q}$ yields $\mathbf{q} = -(Z^T DZ)^{-1} Z^T D\mathbf{x}_0$. Substituting this formula for $\mathbf{q}$ into $\mathbf{x} = \mathbf{x}_0 + Z\mathbf{q}$ yields

$$\mathbf{x} = -Z(Z^T DZ)^{-1} Z^T D\mathbf{x}_0 + \mathbf{x}_0.$$

Since $\mathbf{x}_0$ is arbitrary, we see that

$$\|Z(Z^T DZ)^{-1} Z^T D\| = \sup\left\{ \frac{\|\mathbf{x} - \mathbf{x}_0\|}{\|\mathbf{x}_0\|} : \mathbf{x}_0 \neq \mathbf{0}; \mathbf{x} \text{ solves (6)} \right\}.$$

This means that

$$\bar{\chi}_Z = \sup\left\{ \frac{\|\mathbf{x} - \mathbf{x}_0\|}{\|\mathbf{x}_0\|} : D \in \mathcal{D}; \mathbf{x}_0 \neq \mathbf{0}; \mathbf{x} \text{ solves (6)} \right\}.$$

By applying the triangle inequality, we obtain the bound

$$\bar{\chi}_Z \leq \sup\left\{ 1 + \frac{\|\mathbf{x}\|}{\|\mathbf{x}_0\|} : D \in \mathcal{D}; \mathbf{x}_0 \neq \mathbf{0}; \mathbf{x} \text{ solves (6)} \right\}. \tag{7}$$

Now we compute this supremum by solving (6) in a different way. The first equation of (6) is $D\mathbf{x} - A\mathbf{y} = \mathbf{0}$. Multiplying by $A^T D^{-1}$ yields $A^T\mathbf{x} - A^T D^{-1} A\mathbf{y} = \mathbf{0}$. Substituting the second equation yields $A^T\mathbf{x}_0 - A^T D^{-1} A\mathbf{y} = \mathbf{0}$. Solving for $\mathbf{y}$ yields $\mathbf{y} = (A^T D^{-1} A)^{-1} A^T \mathbf{x}_0$. Use the fact that $\mathbf{x} = D^{-1} A\mathbf{y}$ to obtain $\mathbf{x} = D^{-1} A(A^T D^{-1} A)^{-1} A^T \mathbf{x}_0$. Thus,

$$\frac{\|\mathbf{x}\|}{\|\mathbf{x}_0\|} \leq \|D^{-1} A(A^T D^{-1} A)^{-1} A^T\|. \tag{8}$$

Observe that the matrix whose norm we are taking on the right-hand side of (8) is precisely the transpose of the matrix used to define $\bar{\chi}_A$ in Theorem 2.1(b). Thus, we see that the norm on the right-hand side is bounded by $\gamma_m \bar{\chi}_A$, where γ_m is the multiplicative factor that makes the following inequality a true statement for all $m \times m$ matrices B:

$$\|B^T\| \leq \gamma_m \|B\|.$$

For example, $\gamma_m = 1$ if we use the matrix 2-norm and $\gamma_m = m$ if we use the 1-norm or ∞-norm.

If we substitute (8) into (7) we arrive at the following result.

THEOREM 3.1. *Let A be an $m \times n$ matrix of rank n, and let Z be an $m \times (m-n)$ matrix of rank $m-n$ such that $A^T Z = 0$. Then*

$$\bar{\chi}_Z \leq 1 + \gamma_m \bar{\chi}_A,$$

where γ_m is defined in the last paragraph.

This gives a relationship between $\bar{\chi}_Z$ and $\bar{\chi}_A$. A relationship between χ_Z and $\bar{\chi}_A$ or χ_A is not possible without making further assumptions (for example, that $Z^T Z$ is well-conditioned).

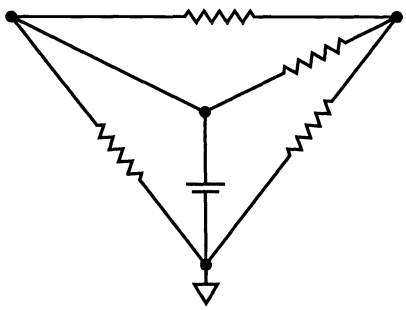
4. The standard algorithms. In this section we describe four standard algorithms for (1). All four of these algorithms are shown to be unstable in the sense of stability given in §2. By "standard" we mean that these algorithms are described in optimization textbooks such as those by Coleman [5], Gill, Murray, and Wright [9], and Fletcher [7], electrical engineering textbooks such as that by Chua, Desoer, and Kuh [4]; and civil engineering textbooks such as that by Timoshenko and Young [22]. In all four cases we have used simple implementations of the algorithms. In the next section we consider stabilizations proposed in the recent literature.

4.1. Symmetric indefinite factorization. Observe in (1) that if we replace $\mathbf{y}$ by $-\mathbf{y}$, the coefficient matrix becomes the symmetric matrix

$$\begin{pmatrix} D & A \\ A^T & 0 \end{pmatrix}.$$

Therefore, we could think about solving this equation with a standard "stable" algorithm for symmetric indefinite linear systems. (See [11] for some algorithms for this purpose.) Symmetric factorization has the disadvantage that it does not respect the block of zeros and therefore requires more operations than necessary. In the electrical engineering context, solving (1) directly is known as *sparse tableau analysis.* In optimization, this algorithm is called the *augmented system* method.

This algorithm should be rejected for the following numerical reason. The algorithm "mixes" $\mathbf{b}, \mathbf{c}$ on the right and $\mathbf{x}, \mathbf{y}$ on the left during the forward substitution



(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 10^{-15} & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 10^{-15} & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(b)

FIG. 1. *The circuit depicted in* (a) *yields an equilibrium system of the form* (1) *with* $D, A, \mathbf{b}$, *as in* (b). *Here all the resistors have size* 1 *and wires without an indicated resistor have resistance* 10^{-15}. *The correct values of the voltages of the three nodes are* $(1.00, 1.00, 0.67)$ *accurate to two digits. The Parlett–Reid algorithm, however, yields* $(1.31, 1.31, 0.88)$ *when* D *is replaced by* $10^{20}D$ *and yields* $(0.86, 0.98, 0.58)$ *when* D *is replaced by* $10^{25}D$.

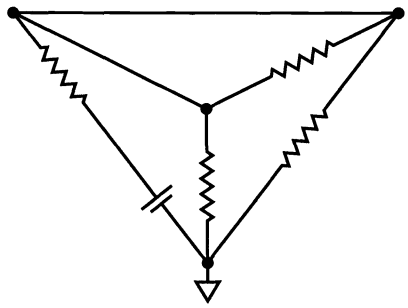
and back substitution process. This does not make sense physically, because $\mathbf{b}, \mathbf{c}$ and $\mathbf{x}, \mathbf{y}$ have different physical units and could be on vastly different scales numerically.

To put it another way, this algorithm is sensitive to scaling matrix D by a constant multiple. On the other hand, $\mathbf{y}$ is mathematically unchanged by such a scaling. Indeed, for a simple electrical circuit depicted in Fig. 1, we scaled D by varying amounts and applied the Parlett–Reid [17] symmetric factorization. Substantially different values of $\mathbf{y}$ (the voltages) were recovered, some accurate to less than one decimal place.

These tests, as well as the others in this paper, were conducted in MATLAB™. MATLAB, a software package for interactive numerical computation, is a trademark of The Mathworks, Inc. All the computations were done in IEEE double-precision arithmetic, with about 15 decimal digits of accuracy.

Standard stability results for algorithms like the Parlett–Reid algorithm do not apply because of our stated interest to recover $\mathbf{y}$ accurately, which is only a part of the solution vector. Moreover, the accuracy of these algorithms depends on the condition number of the equilibrium system (1), which depends on the condition number of D (another reason for rejecting this approach).

For the problem in this example, $\chi_A, \bar{\chi}_A$ are both small because A is a reduced



(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 10^{-15} & & \\ & & & & 10^{-15} & \\ & & & & & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(b)

FIG. 2. *The circuit depicted in* (a) *yields an equilibrium system of the form* (1) *with* $D, A, \mathbf{b}$, *as in* (b). *Here all the resistors have size* 1 *and wires without an indicated resistor have resistance* 10^{-15}. *The correct values of the voltages of the three nodes are* $(0.33, 0.33, 0.33)$ *accurate to two digits. The range-space algorithm, however, yields* $(0.28, 0.28, 0.28)$.

node-arc incidence matrix (see §8). Thus, the failure of this algorithm to satisfy something like (5) indicates that the algorithm is unstable, not that the problem is poorly posed.

4.2. The range-space method. This method obtains $\mathbf{y}$ by solving (2) explicitly. Since $A^T D^{-1} A$ is positive definite, (2) can be solved with Cholesky factorization. Note that this method, as well as the nullspace method described below, is insensitive to scaling of D by a constant multiple.

This method is called the *range-space* method in optimization. In structural analysis it is known as the *displacement* method. In electrical engineering it is called the *nodal analysis* method. In finite elements, the matrix $A^T D^{-1} A$ is known as the *assembled stiffness matrix.* This matrix is known as the *Schur complement* in optimization.

It should be apparent that if D is severely ill-conditioned, then $A^T D^{-1} A$ can also be severely ill-conditioned. (Indeed, in finite precision arithmetic, $A^T D^{-1} A$ may not even be positive definite.) This means that inaccurate answers (i.e., answers not respecting (5)) may be obtained. An example of a circuit solution obtained with the range-space method is indicated in Fig. 2. Observe that not even one significant decimal place is obtained in the answer.

4.3. The nullspace method. This method obtains $\mathbf{x}$ first and then $\mathbf{y}$. The essential ingredients of the nullspace method were described in §3. Following Fletcher [7], we can describe this method as follows. First, a pair of matrices Y, Z is obtained, where Y is an $m \times n$ matrix such that $A^T Y = I$, the identity, and Z is a nullspace basis; that is, an $m \times (m-n)$ of full column rank such that $A^T Z = 0$. Matrix Y is called a *right-inverse* of A^T.

The most stable way to obtain Y, Z is via a QR factorization of A, say $A = QR$. Then Z is set to the last $m-n$ columns of Q and Y is set to the product $Q_1 R_1^{-T}$, where Q_1 is the first n columns of Q and R_1 is the first n rows of R.

Once Y, Z are obtained, we observe that the equation $A^T\mathbf{x} = \mathbf{0}$ means $\mathbf{x} = Z\mathbf{q}$ for some $(m-n)$-vector $\mathbf{q}$. This is because Z spans the nullspace of A. Substituting $\mathbf{x} = Z\mathbf{q}$ into the equation $D\mathbf{x} - A\mathbf{y} = \mathbf{b}$ yields $DZ\mathbf{q} - A\mathbf{y} = \mathbf{b}$. Now multiply by Z^T, using the fact that $Z^T A = 0$, to obtain $Z^T DZ\mathbf{q} = Z^T\mathbf{b}$. The matrix $Z^T DZ$ is known as the *reduced Hessian* in the optimization literature.

The linear system $Z^T DZ\mathbf{q} = Z^T\mathbf{b}$ is symmetric and positive definite, and hence may be solved with Cholesky factorization to obtain $\mathbf{q}$. Once $\mathbf{q}$ is obtained, $\mathbf{x}$ can be obtained with the formula $\mathbf{x} = Z\mathbf{q}$. Finally, $\mathbf{y}$ is obtained from the equation $D\mathbf{x} - A\mathbf{y} = \mathbf{b}$ by multiplying through by Y^T, yielding $\mathbf{y} = Y^T(D\mathbf{x} - \mathbf{b})$.

This method is known as the *nullspace method* in optimization. It is called the *force method* in civil engineering, and *loop analysis* in electrical engineering.

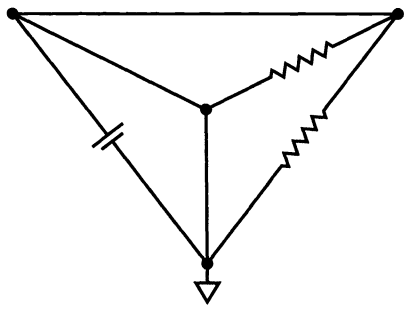
It should be apparent that this method suffers from the same flaw as the range-space method. The matrix $Z^T DZ$ could be arbitrarily ill-conditioned, and hence solving a linear system with this matrix could give a highly inaccurate answer. Indeed, for the example in Fig. 3, the nullspace method returned an answer for $\mathbf{y}$ without any digits of accuracy.

4.4. Gaussian elimination with partial pivoting. Gaussian elimination with partial pivoting is generally not recommended for equilibrium systems because it ignores the special structure. Furthermore, it suffers from the same disadvantages as symmetric indefinite factorization as far as stably solving (1) is concerned. Nonetheless, one might be tempted to apply it directly to (1) since it is widely and easily accessible in many software packages.

As the reader might expect by now, Gaussian elimination with partial pivoting (as implemented by the '\' operation in MATLAB) also failed to give an accurate answer as depicted in Fig. 4.

5. Stabilizing the standard algorithms. The implementations in the last section of the standard algorithms were straightforward, without any special techniques applied. In this section we describe some techniques known from recent literature that may stabilize these algorithms. Additional algorithms known in the literature are described below. As far as we know, none of these stabilizations, nor any of the other algorithms known in the literature, are provably stable. (The only provably stable algorithm of which we are aware is the NSH algorithm described in the upcoming sections.) On the other hand, we do not have any counterexamples showing that these algorithms are unstable. Therefore, it remains an interesting open question to determine the stability of these algorithms.

As for the augmented system method, one might hypothesize the existence of a special value of the scaling parameter that will give an accurate solution for $\mathbf{y}$ in (1) and therefore constitute a stable algorithm. It should be noted that there are simple examples of (1) such that, no matter how D is scaled, the augmented system matrix



(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 10^{-16} & & & & & \\ & 1 & & & & \\ & & 10^{-16} & & & \\ & & & 10^{-16} & & \\ & & & & 10^{-16} & \\ & & & & & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(b)

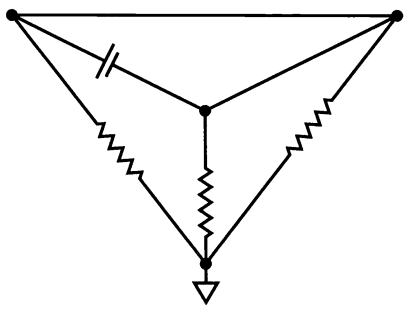
FIG. 3. *The circuit depicted in* (a) *yields an equilibrium system of the form* (1) *with* $D, A, \mathbf{b}$, *as in* (b). *Here, all the resistors have size* 1, *and wires without an indicated resistor have resistance* 10^{-16}. *The correct values of the voltages of the three nodes are* $(0.67, 0.33, 0.67)$ *accurate to two digits. The nullspace algorithm, however, yields* $(0.89, 0.52, 1.16)$.

is ill-conditioned (e.g., $A = [1, 1, 0]^T$ and $D = \text{diag}(1, 1, \epsilon)$). The possibility remains that in spite of the ill-conditioning, the system could still be accurately solved for $\mathbf{y}$.

In particular, Björck [1] has analyzed symmetric factorization of (1); his work attempts to identify the proper selection of the scaling factor to be applied to D to obtain $\mathbf{x}$ or $\mathbf{y}$ accurately. Even with Björck's choice of scaling factor, however, it is not clear that the resulting algorithm is stable.

As for the range-space method, it is possible that reordering of the variables might help. In our testing of the range-space method, we observed that the choice of numbering for edges and nodes affected the resulting answer and some numberings for the example in Fig. 2 gave better answers than the numbering depicted. It is known in the optimization community (Y. Li, private communication) that a good ordering can be obtained with diagonal pivoting. This may be related to orderings for least-squares problems. See Van Loan [24] and the analysis of Golub's [10] algorithm by Powell and Reid [18]. Nonetheless, even with a different ordering, we do not see any evidence that the resulting algorithm will be stable in the sense of §2, because $A^T D^{-1} A$ is ill-conditioned regardless of the ordering.

We do not know of any stabilization proposed in the literature for the nullspace method.



(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 0.3 & & & & & \\ & 0.8 & & & & \\ & & 0.3 & & & \\ & & & 10^{-17} & & \\ & & & & 10^{-17} & \\ & & & & & 10^{-17} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

(b)

FIG. 4. *The circuit depicted in* (a) *yields an equilibrium system of the form* (1) *with* $D, A, \mathbf{b}$, *as in* (b). *Here two resistors have resistance* 0.3 *and the third* (*on the right in the figure*) *has resistance* 0.8. *The wires without an indicated resistor have resistance* 10^{-17}. *The correct values of the voltages of the three nodes are* $(-0.33, 0.33, 0)$ *accurate to two digits. Gaussian elimination with partial pivoting, however, yields* $(0.45, 1.11, 0.78)$.

There are other algorithms for (1) in the recent literature; we do not know whether they are stable. Here is a partial list.

1. Björck and Duff [2] propose an elimination-based approach with a special kind of pivoting.

2. Gulliksson and Wedin [12] propose to factor $A^T D^{-1} A$ using scaled Householder transformations; that is, transformations that are orthogonal in a scaled inner product.

3. Paige [15] applies orthogonal factorization to an augmented system (not the same augmented system as (1)).

6. A stable hybrid algorithm. We now describe a stable algorithm for (1). We call this algorithm "hybrid" because it works with both the range-space of A and nullspace of A^T. The scaling matrix D is applied to the nullspace. Thus, the particular method described in this section is referred to as the "nullspace-scaled hybrid" (NSH) method, to distinguish it from other hybrid methods that we introduce later.

The first step of the NSH method is to compute a nullspace basis V for $A^T D^{-1}$. Thus, V is an $m \times (m-n)$ matrix of rank $m-n$ such that $A^T D^{-1} V = 0$. The matrix V must have a number of special properties, as will become apparent from the lemma

and theorem that follow. Our algorithm for V is described in §7.

Once V is obtained, the linear system

$$[A, V]\begin{pmatrix} \mathbf{y} \\ \mathbf{q} \end{pmatrix} = -\mathbf{b} \tag{9}$$

is solved for $\mathbf{y}$ and $\mathbf{q}$. We claim that $\mathbf{y}$ is indeed a solution for the original problem. This linear system can be rewritten as $A\mathbf{y}+V\mathbf{q} = -\mathbf{b}$. Multiplying by A^TD^{-1} causes the second term on the left to drop out, yielding $A^TD^{-1}A\mathbf{y} = -A^TD^{-1}\mathbf{b}$. Thus, this method solves (2) without explicitly forming $A^TD^{-1}A$. Notice that D does not appear at all in the linear system (9).

The NSH method does not appear in standard textbooks, but it has appeared in the literature. For example, Coleman and Li [6] suggest a similar approach (called the "full-space" method) for optimization, but with the scaling done in a different manner.

We now investigate the numerical stability of this method. We start with a preliminary lemma. For the rest of this section $\|\cdot\|$ is some p-norm.

LEMMA 6.1. *Assume the null basis V computed above is normalized so that $\|V\| = \|A\|$. Suppose also that V is well-conditioned, in the sense that there is a constant $\theta > 0$ such that*

$$\|V\mathbf{x}\| \geq (\|V\| \cdot \|\mathbf{x}\|)/\theta \tag{10}$$

for all $\mathbf{x}$. Let $M = [A, V]$. Then

$$\kappa(M) \leq 2\chi_A \cdot \|A\| + 2\theta \cdot (1 + \bar{\chi}_A), \tag{11}$$

where $\kappa(M)$ denotes the condition number *of M, that is, $\|M\| \cdot \|M^{-1}\|$.*

Proof. The basic idea of this proof is that the condition number of $M = [A, V]$ depends on three things: the condition of A, the condition of V (i.e., the value of θ), and the angle between the span of A and the span of V. But this angle cannot be too small because of Stewart's lemma in §2.

Suppose that $M\mathbf{z} = \mathbf{c}$ for an arbitrary $\mathbf{c}$. Split $\mathbf{z}$ into two components, say $\mathbf{y}$ and $\mathbf{q}$. Then $A\mathbf{y} + V\mathbf{q} = \mathbf{c}$. Multiply by A^TD^{-1} to obtain $A^TD^{-1}A\mathbf{y} = A^TD^{-1}\mathbf{c}$. Thus, $\|\mathbf{y}\| \leq \chi_A\|\mathbf{c}\|$ and $\|A\mathbf{y}\| \leq \bar{\chi}_A\|\mathbf{c}\|$ by Stewart's theorem. Next, $V\mathbf{q} = \mathbf{c} - A\mathbf{y}$, so

$$\|V\mathbf{q}\| \leq \|\mathbf{c}\| + \bar{\chi}_A\|\mathbf{c}\|.$$

Finally,

$$\begin{aligned} \|\mathbf{q}\| &\leq \frac{\theta}{\|V\|} \cdot \|V\mathbf{q}\| \\ &\leq \frac{\theta}{\|A\|} \cdot (1 + \bar{\chi}_A) \cdot \|\mathbf{c}\|. \end{aligned}$$

Thus, we have

$$\begin{aligned} \|\mathbf{z}\| &\leq \|\mathbf{y}\| + \|\mathbf{q}\| \\ &\leq \|\mathbf{c}\| \cdot \left(\chi_A + \frac{\theta}{\|A\|} \cdot (1 + \bar{\chi}_A)\right). \end{aligned}$$

Since $\|M^{-1}\|$ is the supremum of $\|\mathbf{z}\|$ in the case that $\|\mathbf{c}\| = 1$, the parenthesized factor on the right-hand side is an upper bound on $\|M^{-1}\|$.

Now finally, observe that $\|M\| \leq 2\|A\|$. Thus, multiplying through by $2\|A\|$ gives the result. □

Note that the condition that $\|A\| = \|V\|$ does not have to be satisfied exactly; it suffices in the above argument to have A and V on the same order.

Let us denote with κ_A the expression on the right-hand side of (11). This expression has a complicated form, but one sees that it is independent of D except insofar as it depends on V, and is invariant if A is multiplied by a constant. We now can state the main theorem about the stability of the NSH method.

THEOREM 6.2. *Assume V satisfies the conditions of Lemma* 6.1. *Suppose that the algorithm to solve the linear system $A\mathbf{y} + V\mathbf{q} = -\mathbf{b}$ in floating point arithmetic returns approximate solution $(\hat{\mathbf{y}}, \hat{\mathbf{q}})$ such that*

$$(A+E)\hat{\mathbf{y}} + (V+F)\hat{\mathbf{q}} = -\mathbf{b} + \mathbf{e}, \tag{12}$$

where $\|[E,F]\| \leq \epsilon\|[A,V]\|$ and $\|\mathbf{e}\| \leq \epsilon\|\mathbf{b}\|$. Assume that $\epsilon \leq 1/(3\kappa_A)$. Then we have the following error bound:

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \leq 3\epsilon \frac{\kappa_A^2}{\|A\|} \cdot \|\mathbf{b}\|, \tag{13}$$

where $\mathbf{y}$ is the true solution.

Remark 1. Note that (13) indicates that the NSH algorithm is stable, because D does not enter the bound.

Remark 2. The theorem seemingly requires that an *exact* nullspace basis V of $A^T D^{-1}$ be computed. We remark further on this in §7.

Remark 3. The condition that the floating-point algorithm satisfies an error bound like (12) is very natural, and any well-known stable algorithm, say Gaussian elimination with partial pivoting, satisfies such a bound.

Proof. Let $\mathbf{z} = (\mathbf{y}^T, \mathbf{q}^T)^T$ and $\hat{\mathbf{z}} = (\hat{\mathbf{y}}^T, \hat{\mathbf{q}}^T)^T$. Following the standard analysis of [11, Thm. 2.7.2] and using the previous lemma, we obtain the bound

$$\|\mathbf{z} - \hat{\mathbf{z}}\| \leq 3\epsilon\kappa_A \cdot \|\mathbf{z}\|.$$

But

$$\begin{aligned} \|\mathbf{z}\| &\leq \|M^{-1}\| \cdot \|\mathbf{b}\| \\ &\leq \frac{\kappa_A}{\|A\|} \cdot \|\mathbf{b}\|. \end{aligned}$$

Thus,

$$\|\mathbf{z} - \hat{\mathbf{z}}\| \leq 3\epsilon \frac{\kappa_A^2}{\|A\|} \cdot \|\mathbf{b}\|.$$

Finally, $\|\mathbf{y} - \hat{\mathbf{y}}\|$ is at most $\|\mathbf{z} - \hat{\mathbf{z}}\|$. □

7. Obtaining V for Theorem 6.2. As mentioned in §6, Theorem 6.2 seemingly requires V to be computed exactly. In fact, it is implicit in the theorem that it suffices to compute a matrix $\hat{V}$ such that $\hat{V} + F'$ is a nullspace basis for $A^T D^{-1}$, where F' is a small error matrix. This is because using $\hat{V}$ instead of V in (9) is equivalent to compounding the error F in (12) with the further small error F', so the same bound holds with a larger constant.

The standard general-purpose algorithms for nullspace bases do not guarantee that the computed nullspace basis is only a small distance from a true nullspace basis (i.e., a forward-error bound). Instead, these algorithms guarantee that each column of the computed nullspace basis is an exact nullspace vector for a perturbed matrix $A^T D^{-1} + E$ (i.e., a backward-error bound). This is not useful in the present context because adding E to $A^T D^{-1}$ spoils the special structure that allows Stewart's theorem to be applied.

We now describe a technique to obtain a V with a forward-error bound. First, we obtain a nullspace basis Z for A^T. Then we obtain a nullspace basis V for $A^T D^{-1}$ with the formula $V = DZR$, where R is another diagonal matrix described below.

To compute Z, find a subset of rows of n rows of A that form a *basis*, that is, an $n \times n$ nonsingular submatrix of A. Now, for each of the remaining $m - n$ "nonbasic" rows, solve for the nonbasic row in terms of the basis rows. The coefficients of this dependence form a column vector with m entries, of which at most $n+1$ are nonzero. This yields a *fundamental* nullspace basis Z; that is, a nullspace basis for A^T with an embedded $(m-n) \times (m-n)$ identity matrix.

If we assume that $\bar{\chi}_A$ is reasonably small, then it turns out that this nullspace basis has a special property, namely, all of the nonzero entries have roughly the same magnitude. If we think of a vector $\mathbf{x}$ as being one column of Z, we can state this as a lemma. In this lemma, we assume that the ∞-norm is used throughout. The constant $\bar{\chi}_Z$ arises in this lemma; recall that $\bar{\chi}_Z$ is bounded in terms of $\bar{\chi}_A$, as demonstrated in Theorem 3.1.

LEMMA 7.1. *Let* $\mathbf{x}$ *be a nonzero vector such that* $A^T\mathbf{x} = \mathbf{0}$. *Assume that* $\mathbf{x}$ *has nonzero entries in positions* $\{i_1, \ldots, i_p\}$, *where* $\{i_1, \ldots, i_p\} \subset \{1, \ldots, m\}$. *Suppose also that for every subset* I *of size* $p-1$ *of* $\{i_1, \ldots, i_p\}$, *the rows of* A *indexed by* I *are linearly independent. Then the magnitude of the smallest nonzero entry in* $\mathbf{x}$ *is no smaller than* $\|\mathbf{x}\|_\infty / \bar{\chi}_Z$.

Proof. Without loss of generality, we can take $\|\mathbf{x}\| = 1$. Let δ be the magnitude of the entry in $\mathbf{x}$ with the smallest nonzero magnitude and suppose this entry is in position i_p. Since rows $i_1, \ldots, i_{p-1}$ of A are linearly independent, there is a vector $\mathbf{w}$ such that $A\mathbf{w}$ attains arbitrarily specified entries in these $p-1$ positions. In particular, there is a vector $\mathbf{w}$ such that $A\mathbf{w}$ agrees with $\mathbf{x}$ in positions $i_1, \ldots, i_{p-1}$.

Now let C be a diagonal matrix with entry "1" in positions $i_1, \ldots, i_{p-1}$, and very small entries in the other diagonal positions. Observe that $CA\mathbf{w}$ agrees with $\mathbf{x}$ in positions $i_1, \ldots, i_{p-1}$, and has nearly zeros in all the other positions, including i_p. Let $\mathbf{y} = CA\mathbf{w}$. Then we observe that if we define X, Y as in Lemma 2.2 for Z, i.e.,

$$X = \{\mathbf{z} : \mathbf{z} = Z\mathbf{w} \text{ for some } \mathbf{w}, \text{ and } \|\mathbf{z}\| = 1\}$$

and

$$Y = \{\mathbf{z} : Z^T C^{-1} \mathbf{z} = \mathbf{0} \text{ for some } C \in \mathcal{D}\},$$

then $\mathbf{x} \in X$ and $\mathbf{y} \in Y$. This means that $\|\mathbf{x} - \mathbf{y}\|$ is at least $1/\bar{\chi}_Z$. But $\|\mathbf{x} - \mathbf{y}\|$ is arbitrarily close to δ (depending on how close to zero the diagonal entries of C other than $i_1, \ldots, i_{p-1}$ are). Thus, $\delta \geq 1/\bar{\chi}_Z$. This proves the lemma, since we assumed $\|\mathbf{x}\| = 1$. □

This lemma has two important consequences. First, the lemma implies that the nullspace basis Z is well conditioned in the sense of (10). In particular, with suitable

ordering of the rows and columns of Z, we can write it in the form

$$Z = \begin{pmatrix} I \\ W \end{pmatrix}, \tag{14}$$

where W is a matrix whose entries are at most $\bar{\chi}_Z$ in magnitude.

Second, the lemma has the following more subtle consequence: if the columns of Z are calculated with a small forward error, this means in fact that these entries have a small *componentwise* forward error. We can express this formally as follows. Let Z be the nullspace basis of A computed with the above algorithm using exact arithmetic, and let $\hat{Z}$ be the nullspace basis computed in the presence of floating point errors. If we assume that the columns of Z are computed with a small forward error, that is,

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \epsilon \|\mathbf{x}\|,$$

where $\mathbf{x}$ is a column of Z and $\hat{\mathbf{x}}$ is a column of $\hat{Z}$, then we immediately obtain the result

$$|\mathbf{x} - \hat{\mathbf{x}}| \leq \epsilon \bar{\chi}_Z |\mathbf{x}|.$$

The componentwise error bound is important for the following reason: to obtain V, recall that we form the product DZR, where D is the diagonal matrix from (1), and R is a diagonal matrix not yet specified. Because D, R are both diagonal and Z has a small componentwise error, then DZR also will have a small componentwise forward error, and therefore, a small norm forward error, no matter how D, R are chosen.

The argument in the last two paragraphs presupposed that the columns of Z would have a small normwise forward error. A small normwise forward error is obtained if the basis rows of A are well-conditioned, because the columns of Z are solutions to linear systems involving the basis rows. Now we ask whether the basis rows of A are well-conditioned. In fact, there is a bound on the condition number of the basis rows; we state another lemma in this regard. This lemma is based on similar results in [14] and [20].

LEMMA 7.2. *Let A be an $m \times n$ matrix. Let B be a nonsingular $n \times n$ submatrix of A. Then $\|B^{-1}\| \leq \chi_A$.*

Proof. Let $\mathbf{b}$ be an arbitrary n-vector and suppose $\mathbf{x}$ is the solution to $B\mathbf{x} = \mathbf{b}$. We require an upper bound on $\|\mathbf{x}\|$ in terms of $\|\mathbf{b}\|$. Let $\hat{\mathbf{b}}$ be the extension of $\mathbf{b}$ to $\mathbb{R}^m$ obtained by inserting zeros in the $m-n$ positions corresponding to rows of A not in B.

Let C be the diagonal matrix with ones in the n diagonal positions corresponding to rows of B and very small entries in other positions. Then $A^TCA \approx B^TB$ and $A^TC\hat{\mathbf{b}} \approx B^T\mathbf{b}$, where the approximations can be arbitrarily close as we make diagonal entries of C close to zero.

Thus, if vector $\mathbf{x}$ is the solution to $A^TCA\mathbf{x} = A^TC\hat{\mathbf{b}}$, then $B^TB\mathbf{x} \approx B^T\mathbf{b}$. Since B is square and invertible, this last equation means that $B\mathbf{x} \approx \mathbf{b}$. But we also know that $\|\mathbf{x}\| \leq \chi_A \|\hat{\mathbf{b}}\|$ by definition of χ_A. This proves the lemma. □

Recall that for A "well-conditioned," we should have $\chi_A = \phi/\|A\|$, where ϕ is a small constant; thus $\|B^{-1}\| \leq \phi/\|A\|$. Finally, $\|A\| \geq \|B\|$ so

$$\|B^{-1}\| \cdot \|B\| \leq \phi,$$

thus obtaining a condition number bound on B.

Thus, we conclude that we can compute Z using the above algorithm such that Z is simultaneously well conditioned and has a small forward componentwise error. Now, we apply D on the left. As we see from (14), the upper $(m-n) \times (m-n)$ block of DZ is some diagonal matrix. Finally, choose a diagonal matrix R so that the upper $(m-n) \times (m-n)$ block of $V = DZR$ is once again the identity matrix. (This uniquely determines R.)

This algorithm gives a nullspace basis V for $A^T D^{-1}$ with a small forward error. Moreover, V has the form

$$V = \begin{pmatrix} I \\ W' \end{pmatrix}. \tag{15}$$

There is no reason to believe, however, that V is well-conditioned. In fact, W' can have arbitrarily large entries if the basic rows of A have large corresponding entries in D.

We now show how to maintain control over the size of W' by correctly choosing the basis rows in A. (Until now, we have not made any assumptions about the basis rows except that they are independent.) The algorithm for obtaining B is the following "greedy" approach. Assign *weights* to the rows, where the weight of the ith row is the (i,i) entry of D. Now, select the row with the lowest weight and insert it in B. Continue appending the lowest-weighted remaining row of A to B. Before appending a row to B, check that the row is linearly independent from the rows already in B. If a dependence is found, then the row is discarded. Continue this process until B has n rows.

Note that this algorithm requires a "yes/no" test concerning linear dependence among rows of A. In general, such tests are considered to be numerically hazardous. In our application, however, this is a safe procedure; a consequence of the preceding lemmas and arguments is that if $\chi_A, \bar{\chi}_A$ are well bounded, then there can be no "near" dependence among subsets of rows of A.

Since A is sparse for many applications, this process of testing rows for independence from previous rows should involve both combinatorial and numerical tests for dependence, such as the inner loop of the Gilbert and Heath [8] algorithm for nullspace bases. Briefly, the Gilbert–Heath algorithm maintains a bipartite graph indicating the nonzero structure of the current basis and a complete matching of that bipartite graph. There is a simple combinatorial technique to detect many cases of dependency among rows. Indeed, for RNAI matrices (described in §8), the process of selecting B is purely combinatorial and very efficient.

We now claim that if B is selected with this greedy approach, then W' in (15) has a small bound. Consider the (i,j) entry of V. Assume $i > m-n$, so that the entry in question falls in the W' portion in (15). Assume that this entry is nonzero. This means that the jth row of A is expressed in terms of rows $m-n+1, \ldots, m$ with a nonzero coefficient for the ith row. We claim that this implies that $d_{ii} \leq d_{jj}$. Suppose not; suppose that $d_{jj} < d_{ii}$. Observe that, by the positions of the rows, i is basic and j is nonbasic. This means that j was passed over by the greedy algorithm for forming the basis, since row i was added to B even though it has a higher weight than row j. This in turn means that row j must have been linearly dependent on rows already in B when it was encountered by the greedy algorithm. But this is impossible, because we already know that row j can be expressed in terms of the basis rows with a nonzero coefficient for row i, so row j cannot be dependent on B until after row i is added to B. Thus, this contradiction shows that $d_{ii} \leq d_{jj}$.

Finally, when we scale Z on the left by D and on the right by R, it is easy to see that the (i,j) entry of V is equal to the (i,j) entry of Z multiplied by d_{ii}/d_{jj}, which we now have shown to be a positive scalar no larger than one. This shows that the entries of W' in (15) are no larger than the corresponding entries of W in (14), and we already have argued that W has a small bound.

If V has the form (15) with a bound on W', then it is well conditioned in the sense of Theorem 6.2. Let $\mathbf{x}$ be an arbitrary vector, and consider $\mathbf{y} = V\mathbf{x}$. Observe that $\mathbf{y}$ agrees with $\mathbf{x}$ in its first $m-n$ positions. Thus, $\|\mathbf{y}\| \geq \|\mathbf{x}\|$. On the other hand, $\|V\| \leq 1 + \|W'\|$. Hence if we take $\theta = 1 + \|W'\|$, then we have the bound

$$\|V\mathbf{x}\| \geq (\|V\| \cdot \|\mathbf{x}\|)/\theta$$

needed for Theorem 6.2.

The only property remaining that V must have is the equation $\|V\| = \|A\|$. This is easily obtained by scaling V uniformly.

8. Reduced node-arc incidence (RNAI) matrices. A special class of matrices A arising in many applications are RNAI matrices. Let G be an directed graph, weakly connected, with no parallel edges and no self-loops. (The assumptions about connectivity and parallel edges could be easily removed at the expense of a more complicated exposition.) Assume G has m arcs and $n+1$ nodes.

The *node-arc incidence* matrix for G is a matrix A_0 with one row for each edge of G and one column for each node. Each row contains two nonzero entries, a 1 and a -1. The -1 entry occurs in the column corresponding to the tail of the arc, and the 1 entry occurs in the column corresponding to its head.

The RNAI matrix A for G is obtained by deleting a column of A_0 corresponding to an arbitrary node. The deleted node is called "ground" in an electrical engineering context. In an RNAI matrix, each row has either a $1/-1$ pair of entries, or a lone 1 or lone -1 in rows corresponding to arcs with one end grounded.

Electrical networks are the main application in which A of (1) turns out to be an RNAI matrix. Indeed, the matrix A occurring in Figs. 1– 4 is an RNAI matrix. Optimization problems with flow constraints can give rise to RNAI or related matrices. Structural analysis in civil engineering can give rise to matrices A that have a block-RNAI structure.

It is a well-known fact of algebraic graph theory (see, for example, Welsh [25]) that an RNAI matrix A has full column rank. Furthermore, the process of finding a basis among the rows of A corresponds to identifying a spanning tree of G_0, where G_0 is the undirected graph that results when the arcs of G are stripped of their orientation. A *spanning tree* is a subgraph T of G_0 that is incident upon every vertex of G_0, is connected, and has no cycles. The resulting nullspace basis Z has the form (14), with the additional property that every entry in W is either 0, 1, or -1. Thus, there is a very good upper bound on $\|W\|$.

The greedy algorithm described in the previous section corresponds to finding a *minimum-weight* spanning tree. The minimum-weight spanning tree can be computed very efficiently; see, for example, Papadimitriou and Steiglitz [16]. From this spanning tree we construct Z as described above, and then V by multiplying DZR. The process of obtaining Z and the basis is completely combinatorial for an RNAI matrix.

We have tried this version of the NSH algorithm in MATLAB on all the problems in §4. We used an $O(m \log n)$ algorithm for constructing minimum-weight spanning trees. We obtained answers with approximately 15 digits of accuracy in all cases.

These results show that we can easily construct a V with the right properties for Theorem 6.2 for RNAI matrices. A natural follow-up question is whether the constants in the theorem, namely χ_A and $\bar{\chi}_A$, have reasonable bounds for RNAI matrices. Such bounds are also important because we want to argue that the incorrect answers obtained with the standard methods in §4 are due to numerical problems with the algorithms and not ill-conditioning in A.

In fact, both constants $\chi_A, \bar{\chi}_A$ have small bounds for RNAI matrices, as stated in the following theorem.

THEOREM 8.1. *Let A be an* RNAI *matrix of size $m \times n$. Then $\chi_A \leq n$ and $\bar{\chi}_A \leq n$ if the ∞-norm is used.*

Proof. Suppose the equilibrium equations hold, i.e.,

$$\begin{pmatrix} D & -A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \tag{16}$$

for an arbitrarily chosen $\mathbf{b}$ such that $\|\mathbf{b}\|_\infty \leq 1$. Parameter χ_A is defined to be the maximum possible value of $\|\mathbf{y}\|_\infty$. Label each node in the underlying graph with its y value and label the ground node with 0. In analogy with electrical engineering, we call these labels the "voltages."

Let $p_1, \ldots, p_{n+1}$ be the voltages written in descending order, i.e., $p_1, \ldots, p_{n+1}$ is a sorted version of $\{y_1, \ldots, y_n, 0\}$. A simple averaging argument shows that there must be a μ such that

$$p_\mu - p_{\mu+1} \geq (p_1 - p_{n+1})/n.$$

Here, p_1 is the maximum voltage and p_{n+1} the minimum. Because ground is labeled, $p_1 \geq 0 \geq p_{n+1}$.

Let S be the nodes that were numbered 1 to μ in this sorted list, and let T be nodes numbered $\mu + 1$ to $n + 1$ in the sorted list. Then (S, T) is the partition of the nodes such that each node in S has a voltage at least g higher than every node in T, where $g = (p_1 - p_{n+1})/n$.

Assume from now on that ground is in S (an interchanging of S and T in the following argument would handle the opposite case). Let $\mathbf{v}$ be an n-vector that, for each i, has 1 in position i if node i lies in T, and a 0 otherwise. (From now on "node i" refers to the original numbering of the nodes corresponding to the ordering of columns in A.) Consider the product $\mathbf{w} = A\mathbf{v}$; $\mathbf{w}$ has one entry for each edge in the graph. It follows from the definition of an RNAI matrix that w_i is 1 if edge i is a forward edge from S to T, it is -1 if the edge is backward, and 0 if the edge is internal to either S or T. Since the graph is connected, $\mathbf{w} \neq \mathbf{0}$.

We have the equation $A^T\mathbf{x} = \mathbf{0}$; multiplying by $\mathbf{v}^T$ shows that $\mathbf{w}^T\mathbf{x} = 0$. This means that there must exist an i such that w_i is nonzero and $w_i x_i \geq 0$.

Now examine the ith equation from the block $D\mathbf{x} - A\mathbf{y} = \mathbf{b}$. Since w_i is nonzero, one endpoint of edge i is in S and the other endpoint in T. Let the endpoint in T be numbered j. (Since ground is not in T, j corresponds to a true column of A.) Thus, the (i, j) entry of A is ± 1.

There are two cases: either w_i is positive or w_i is negative. If w_i is positive, then by definition of $\mathbf{w}$, we must have that the (i, j) entry of A is $+1$. By choice of i, $x_i \geq 0$, so $d_{ii}x_i \geq 0$. There are two subcases: either the other endpoint of edge i is ground, or it is some other node k of the graph, which lies in S. If it is ground, then the ith equation must be $d_{ii}x_i - y_j = b_i$, i.e., $-y_j \leq b_i$, i.e., $y_j \geq -b_i$. If the other endpoint is node k, then $d_{ii}x_i - y_j + y_k = b_i$, i.e., $y_j - y_k \geq -b_i$

If w_i is negative, then by definition of $\mathbf{w}$, we must have that the (i,j) entry of A is -1. By choice of i, $x_i \leq 0$, so $d_{ii}x_i \leq 0$. There are two subcases: either the other endpoint of edge i is ground, or it is some other node k of the graph, which lies in S. If it is ground, then the ith equation must be $d_{ii}x_i + y_j = b_i$, i.e., $y_j \geq b_i$. If the other endpoint is node k, then $d_{ii}x_i + y_j - y_k = b_i$, i.e., $y_j - y_k \geq b_i$.

Thus, in all four subcases, we have obtained an inequality of the form $y_j - y' \geq -|b_i|$, where y_j is the voltage of a node in T and y' is the voltage of a node in S (possibly ground). Since we already know by construction of S and T that $y_j - y' \leq -g$, we have $-g \geq -|b_i|$, i.e., $g \leq |b_i|$. By assumption on $\mathbf{b}$, we have $|b_i| \leq 1$. Thus $(p_1 - p_{n+1})/n \leq 1$, i.e.,

$$p_1 - p_{n+1} \leq n. \tag{17}$$

Since 0 lies between p_1 and p_{n+1}, and the values of the y' are contained in p's, we conclude from (17) that $|y_i| \leq n$ for all i. This shows that $\chi_A \leq n$.

Next, let us obtain a bound on $\bar{\chi}_A$. Recall that $\bar{\chi}_A$ is the maximum value of $\|A\mathbf{y}\|_\infty$, where $\mathbf{y}$ is the solution to (16) and where $\|\mathbf{b}\|_\infty = 1$. But for an RNAI matrix, each entry of $A\mathbf{y}$ is the difference between two entries in $\mathbf{y}$, or is simply an entry in $\mathbf{y}$ (for edges with one endpoint grounded). But we have shown that the maximum difference between any of the $y's$ is n—this is (17). □

9. Generalizing the problem. In this section we comment on the possibility of lifting some of the assumptions made in §1. One assumption made throughout the paper is that we are interested in recovering $\mathbf{y}$ in (1) rather than $\mathbf{x}$. Assuming we have $\mathbf{y}$, $\mathbf{x}$ can be recovered from the equation $D\mathbf{x} - A\mathbf{y} = \mathbf{b}$. It is easy to see that we can compute $D\mathbf{x}$ stably (i.e., with only small normwise error). Then $\mathbf{x}$ is obtained by scaling the computed $D\mathbf{x}$ with D^{-1}. The recovered $\mathbf{x}$ would have a relative error that could be on the order of the condition number of D. This is inherent in the problem—(1) is ill-conditioned with respect to $\mathbf{x}$ when D is ill-conditioned, and there is no theorem like Theorem 2.1 that holds for $\mathbf{x}$. Thus, we see that recovering $D\mathbf{x}$ accurately is the best that could be hoped for in the case that $\mathbf{b} \neq \mathbf{0}$, $\mathbf{c} = \mathbf{0}$.

This instability is not merely a numerical artifact—there is a corresponding physical instability. For example, in the context of electrical networks with batteries, it is possible to construct a circuit as follows. One arc of the circuit has very low resistance, but also very small current because the battery voltages are balanced so that the potential across the arc is small. Then a small perturbation to the battery voltages (i.e., $\mathbf{b}$) could cause a great multiplication of the current (i.e., $\mathbf{x}$). On the other hand, $D\mathbf{x}$, the voltage drops across the resistors, would suffer only a small perturbation.

Another assumption was that $\mathbf{c} = \mathbf{0}$. Can we drop this assumption? One special case, which is dual to the case considered for most of this paper, is when $\mathbf{c} \neq \mathbf{0}$, $\mathbf{b} = \mathbf{0}$, and we are interested in recovering $\mathbf{x}$ instead of $\mathbf{y}$. In this case, let Z be a nullspace basis for A^T, and let $\mathbf{x}_0$ be an arbitrary solution to the underdetermined system $A^T\mathbf{x} = \mathbf{c}$. Then the formula for $\mathbf{x}$ is obtained by solving $Z^TDZ\mathbf{q} = Z^TD\mathbf{x}_0$, and then setting $\mathbf{x} = Z\mathbf{q} + \mathbf{x}_0$. The equation $Z^TDZ\mathbf{q} = Z^TD\mathbf{x}_0$ is of the exact format of (2). Thus, we can apply the machinery developed in this paper, except with Z in place of A, $\mathbf{x}_0$ in place of $\mathbf{b}$, and D in place of D^{-1}. Notice that the nullspace of Z^T is the same as the range of A. Thus, the version of the hybrid algorithm for this problem would be the range-space scaled hybrid algorithm, or RSH algorithm. Note also that we have derived relationships between $\bar{\chi}_A$ and $\bar{\chi}_Z$ in §3.

In the most general case that $\mathbf{b} \neq \mathbf{0}, \mathbf{c} \neq \mathbf{0}$, it seems unlikely that we could obtain an algorithm that computes either $\mathbf{x}$ or $\mathbf{y}$ accurately when D is ill-conditioned because

both variables become ill-conditioned. A compromise algorithm might be as follows. Let Z be a basis for the nullspace of A^T. Then solve the linear system

$$D^{-1/2}A\mathbf{y} + D^{1/2}Z\mathbf{q} = -D^{-1/2}\mathbf{b} + D^{1/2}\mathbf{x}_0,$$

where $\mathbf{x}_0$ is as in the previous paragraph. Multiplying through by $A^T D^{-1/2}$ shows that this solves the general case of (1) for $\mathbf{y}$. This method has the advantage that it "splits" the ill-conditioning between the nullspace and the range-space. We could call this the biscaled hybrid algorithm, or BSH algorithm.

The last restrictive assumption made in §1 is that D is diagonal. As mentioned earlier, in some finite element and structural problems in two and three dimensions, we can generally only assume that D is block diagonal. Stewart's theorem does not generalize to nondiagonal matrices, as shown in his paper. In the applications of finite elements and structures, however, A has a certain sparsity pattern that is correlated to the blocks of D. Perhaps a result could be established for special structured matrices A.

10. Open questions and future directions. Perhaps the number of open questions raised by this work exceeds the number of results we have obtained. Here are some examples of open questions.

1. Is there a good algorithm to obtain $\chi_A, \bar{\chi}_A$? Here "good" means either that the algorithm is polynomial time, or that it gives an approximate answer in finite precision arithmetic, or both.

2. What are useful bounds for $\chi_A, \bar{\chi}_A$ in the case that A has the special structure arising in applications more general than RNAI matrices?

3. Can Stewart's theorem be generalized to a block-diagonal D in the case that the structure of A is correlated to the structure of D, as in structural analysis?

4. As mentioned in §5, there are techniques known in the literature that can be applied to the standard algorithms. Could it be proved that one of the standard algorithms is stable if one of these techniques is used? The standard algorithms are much simpler than the NSH method.

5. This paper has not dealt at all with issues of algorithmic efficiency, but there are many. For instance, in the NSH method, must we explicitly solve the $m \times m$ system (9) or can we approach it implicitly? The algorithm for computing Z described in §7 was geared solely towards numerical stability. Recently, Stern and Vavasis [19] proposed a way to compute Z for an RNAI matrix based on *separators* of G_0. The Stern–Vavasis algorithm aims for sparsity of Z rather than stability. Is there some combination of the algorithm here with the ideas from the other paper that attains both numerical stability and a sparse nullspace basis Z?

6. For the general case of recovering both $\mathbf{x}$ and $\mathbf{y}$, is the BSH method described in §9 any better than the range-space or nullspace methods? We tried the BSH algorithm using a nullspace basis Z generated by QR factorization described in §4 (in particular, not any special spanning-tree basis). We obtained very accurate answers for all the problems in which the standard methods failed. Is there an explanation for the success of the BSH method?

7. Are there good iterative methods for (1)? For instance, is there an iterative method whose convergence rate depends only on A and gives a good solution for $\mathbf{y}$ in the sense of a bound like (5)?

Acknowledgments. The author benefitted from discussion of these ideas with Tom Coleman, Yuying Li, Danny Ralph, Mike Todd, Nick Trefethen, and Charlie

Van Loan of Cornell; Alan Edelman of Berkeley; Nick Higham of Manchester; and Ilse Ipsen of Yale. Helpful comments were received on the first version of this paper from these people as well as Gene Golub of Stanford, Des Higham of Dundee, Dianne O'Leary of Maryland, Gilbert Strang of MIT, and two anonymous referees.

REFERENCES

[1] A. BJÖRCK, *Pivoting and stability in the augmented system method*, in Numerical Analysis 1991: Proceedings of the 14th Dundee Conference, D. F. Griffiths and G. A. Watson, eds., Longman Scientific and Technical, Harlow, Essex, U.K., 1992.

[2] A. BJÖRCK AND I. S. DUFF, *A direct method for the solution of sparse linear least squares problems*, Linear Algebra Appl., 34 (1980), pp. 43–67.

[3] S. CHANDRASEKARAN AND I. IPSEN, *Perturbation theory for the solution of systems of linear equations*, Tech. Rep. RR-866, Department of Computer Science, Yale University, New Haven, CT, 1991.

[4] L. O. CHUA, C. A. DESOER, AND E. S. KUH, *Linear and Nonlinear Circuits*, McGraw–Hill, New York, 1987.

[5] T. F. COLEMAN, *Large Sparse Numerical Optimization*, Lecture Notes in Computer Science 165, Springer-Verlag, Berlin, New York, 1984.

[6] T. F. COLEMAN AND Y. LI, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Math. Programming, 56 (1992), pp. 189–222.

[7] R. FLETCHER, *Practical Methods of Optimization*, Second ed., J. Wiley and Sons, Chichester, 1987.

[8] J. GILBERT AND M. HEATH, *Computing a sparse basis for the null space*, SIAM J. Algebraic Discrete Meth., 8 (1987), pp. 446–459.

[9] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, London, 1981.

[10] G. GOLUB, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[11] G. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Second ed., Johns Hopkins University Press, Baltimore, 1989.

[12] M. GULLIKSSON AND P.-Å. WEDIN, *Modifying the QR decomposition to constrained and weighted linear least squares*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1298–1313.

[13] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[14] D. P. O'LEARY, *On bounds for scaled projections and pseudoinverses*, Linear Algebra Appl., 132 (1990), pp. 115–117.

[15] C. C. PAIGE, *Fast numerically stable computations for generalized least squares problems*, SIAM J. Numer. Anal., 16 (1979), pp. 165–171.

[16] C. H. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[17] B. N. PARLETT AND J. K. REID, *On the solution of a system of linear equations whose matrix is symmetric but not definite*, BIT, 10 (1980), pp. 386–397.

[18] M. J. D. POWELL AND J. K. REID, *On applying Householder transformations to linear least squares problems*, in Proc. IFIP Congress, 1968, Elsevier–North Holland, Amsterdam, 1969, pp. 122–126.

[19] J. M. STERN AND S. A. VAVASIS, *Nested dissection for sparse nullspace bases*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 766–775.

[20] G. W. STEWART, *On scaled projections and pseudoinverses*, Linear Algebra Appl., 112 (1989), pp. 189–193.

[21] G. STRANG, *A framework for equilibrium equations*, SIAM Rev., 30 (1988), pp. 283–297.

[22] S. P. TIMOSHENKO AND D. H. YOUNG, *Theory of Structures*, Second ed., McGraw-Hill, New York, 1965.

[23] M. J. TODD, *A Dantzig–Wolfe-like variant of Karmarkar's interior-point linear programming algorithm*, Oper. Res., 38 (1990), pp. 1006–1018.

[24] C. F. VAN LOAN, *On the method of weighting for equality-constrained least-squares problems*, SIAM J. Numer. Anal., 22 (1985), pp. 851–864.

[25] D. WELSH, *Matroid Theory*, London Math. Soc. Monograph, Vol. 8, Academic Press, London, 1976.

[26] M. H. WRIGHT, *Interior methods for constrained optimization*, in Acta Numerica 1992, A. Iserles, ed., Cambridge University Press, Cambridge, 1992, pp. 341–407.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1132–1140, October 1994

006

SOME CONVERGENCE PROPERTIES OF MATRIX SETS *

DAVID P. STANFORD† AND JOSÉ MIGUEL URBANO‡

Abstract. A set $\mathcal{A} = \{A_j : j \in J\}$ of $n \times n$ matrices is *pointwise convergent* provided each n-vector x can be steered to zero by iterated multiplication by matrices in $\mathcal{A}$. The convergence is *uniform* if the sequence of multipliers may be chosen independently of x. This paper discusses conditions related to convergence for sets of diagonal, triangular, and general matrices, real and complex. It generalizes known conditions for convergence of a single matrix and characterizes convergence of a set of diagonal matrices in terms of semipositivity of a matrix derived from the set.

Key words. convergence, matrix norm, precontractive, eigenvalue, singular value, semipositive

AMS subject classifications. primary 15A48, 15A60; secondary 93D15

1. Introduction.

The convergence properties of an $n \times n$ complex matrix A are well known. Among them is the equivalence of the following three statements.

(i) $\lim_{k\to\infty} A^k = 0$,

(ii) If x is a complex n-vector, then

$$\lim_{k\to\infty} A^k x = 0,$$

(iii) $\rho(A) < 1$, where ρ denotes spectral radius.

The notion of convergence of a matrix can be extended to that of convergence of a set of matrices in a variety of ways. In this paper we study a definition of convergence of a set of matrices that is motivated by the theory of multirate sampled-data control systems and the multimodal linear control systems to which they give rise [6]. Intuitively, the set $\mathcal{A} = \{A_j : j \in J\}$ of $n \times n$ matrices is "pointwise" convergent if each n-vector x can be steered to zero by repeated multiplication by matrices selected from $\mathcal{A}$. The convergence is "uniform" if the sequence of multipliers can be chosen independently of x. Hence, a set is uniformly convergent if and only if there is an infinite product of matrices in the set that left converges to the zero matrix. Although the sets arising from the control systems in [6] are finite, we also consider here infinite sets. We see, however, that the convergence of an infinite set of matrices is equivalent to that of some finite subset.

Our definition of convergence is substantially different from other generalizations of the convergence of a matrix. For example, in [3] Daubechies and Lagarias consider a definition that requires *all* (left or right) infinite products of matrices in the set to converge (but not necessarily to zero). This work is continued in [1]. The fact that neither definition of convergence contains the other can be seen by the examples

$$\mathcal{S} = \left\{ \operatorname{diag}\left\{\frac{3}{2}, \frac{1}{2}\right\}, \operatorname{diag}\left\{\frac{2}{3}, \frac{-3}{2}\right\} \right\} \quad \text{and} \quad \mathcal{T} = \{I\}.$$

*Received by the editors March 24, 1992; accepted for publication (in revised form) June 1, 1993.

†Department of Mathematics, The College of William & Mary, Williamsburg, Virginia 23187-8795 (stanford@cs.wm.edu).

‡Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal (jmurb@mat.uc.pt).

$\mathcal{S}$ is uniformly, and hence pointwise, convergent (it is a slight modification of the example before Theorem 5.3) but is neither right nor left convergent in the sense of [3]. On the other hand, the set $\mathcal{T}$ whose only member is the identity matrix is both right and left convergent in the sense of [3] but is neither pointwise nor uniformly convergent in our sense.

Section 2 lists the necessary definitions. In §3 we present necessary and sufficient conditions for pointwise and uniform convergence of a matrix set and relate convergence of a set to that of some finite subset. We then present further necessary conditions for each type of convergence. In §4 we show the equivalence of pointwise and uniform convergence for sets of diagonal matrices and relate the convergence of a finite set of diagonal matrices to the semipositivity of a matrix constructed from the set. Finally, in §5 we discuss convergence of sets of triangular matrices.

2. Definitions. Throughout the paper $\mathbb{F}$ denotes either the field $\mathbb{C}$ of complex numbers or the field $\mathbb{R}$ of real numbers. The set of $m \times n$ matrices over $\mathbb{F}$ are denoted by $\mathbb{F}^{m\times n}$, and $\mathbb{F}^n$ denotes $\mathbb{F}^{n\times 1}$. General vector norms and matrix norms are discussed and denoted by $\|\cdot\|_V$ and $\|\cdot\|_M$, respectively. We reserve the symbol $\|\cdot\|_2$ for the euclidean vector norm on $\mathbb{F}^n$. Throughout the paper, J denotes a nonempty index set.

DEFINITION 2.1. *A set $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$ is* convergent at a point $x \in \mathbb{F}^n$ *provided there is a sequence $\{p(x)_i\}_{i=1}^{\infty}$ such that $p(x)_i \in J$ for all i, and the sequence*

$$x, \qquad A_{p(x)_1}x, \qquad A_{p(x)_2}A_{p(x)_1}x, \ldots$$

converges to the zero vector. We assert this convergence by writing

$$\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} A_{p(x)_i}\right]x\right\} = 0.$$

$\mathcal{A}$ is pointwise convergent *provided that it is convergent at each point in $\mathbb{F}^n$. $\mathcal{A}$ is* uniformly convergent *provided the sequence $\{p(x)_i\}_{i=1}^{\infty}$ may be chosen independently of x; that is, there is a sequence $\{p_i\}_{i=1}^{\infty}$ with $p_i \in J$ for all i and*

$$\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} A_{p_i}\right]x\right\} = 0$$

for all $x \in \mathbb{F}^n$.

Clearly our definition of uniform convergence is unchanged if we require the stated limit to be zero only for x in the unit sphere of $\mathbb{F}^n$. Furthermore, the limit is zero for all such x if and only if

$$\lim_{k\to\infty}\left[\prod_{i=k}^{1} A_{p_i}\right] = 0.$$

These observations imply that our use of the term "uniform convergence" is consistent with its usual meaning of "convergence at the same rate for all x."

Of course, for a single matrix pointwise and uniform convergence are the same. We shall see, however, that the set

$$\mathcal{A} = \left\{\begin{bmatrix}\frac{1}{2} & 0\\ 0 & 2\end{bmatrix}, \begin{bmatrix}\frac{\sqrt{3}}{2} & \frac{1}{2}\\ -\frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\right\} \subset \mathbb{R}^{2\times 2}$$

is pointwise convergent but is not uniformly convergent.

The notion of precontractiveness introduced in [6] is useful.

DEFINITION 2.2. *Let $\|\cdot\|_V$ be any norm on $\mathbb{F}^n$, and let $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$. $\mathcal{A}$ is* precontractive relative to $\|\cdot\|_V$ *provided that if $x \in \mathbb{F}^n/\{0\}$, then there is a finite sequence $\{q(x)_i\}_{i=1}^{n(x)}$, with $q(x)_i \in J$ for $i = 1, 2, \ldots, n(x)$, and*

$$\left\| \left[\prod_{i=n(x)}^{1} A_{q(x)_i} \right] x \right\|_V < \|x\|_V .$$

3. Conditions for convergence.

LEMMA 3.1. *$\mathcal{A}$ is precontractive relative to $\|\cdot\|_V$ if and only if some finite subset $\mathcal{B}$ of $\mathcal{A}$ is precontractive relative to $\|\cdot\|_V$.*

Proof. Suppose that $\mathcal{A}$ is precontractive relative to $\|\cdot\|_V$, and let P denote the set of all finite sequences $p = \{p_i\}_{i=1}^{k(p)}$ of members of J. For p in P, let

$$S_p = \left\{ x \in \mathbb{F}^n : \|x\|_V = 1, \left\| \left(\prod_{i=k(p)}^{1} A_{p_i} \right) x \right\|_V < 1 \right\}.$$

Then each S_p is an open subset of the unit V-sphere in $\mathbb{F}^n$ and, by precontractiveness, the sphere is covered by $\{S_p : p \in P\}$. By compactness, there is a finite set $\{p^{(1)}, p^{(2)}, \ldots, p^{(m)}\} \subseteq P$ such that the set $\{S_{p^{(i)}} : i = 1, 2, \ldots, m\}$ covers the sphere. It follows that the finite subset $\mathcal{B} = \bigcup_{i=1}^{m} \bigcup_{j=1}^{k(p^{(i)})} A_{p_j^{(i)}}$ of $\mathcal{A}$ is precontractive. The reverse implication is obvious. □

THEOREM 3.2. *Let $\|\cdot\|_V$ be a norm on $\mathbb{F}^n$ and let $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$. Then $\mathcal{A}$ is pointwise convergent if and only if $\mathcal{A}$ is precontractive relative to $\|\cdot\|_V$.*

Proof. A proof for the case $\mathbb{F} = \mathbb{R}$ and $\mathcal{A}$ finite appears in [6, Thm. 1]. The proof remains valid when $\mathbb{R}$ is replaced by $\mathbb{F}$, leaving only the case that $\mathcal{A}$ is infinite. If $\mathcal{A}$ is precontractive, then by Lemma 3.1 there is a finite subset $\mathcal{B}$ of $\mathcal{A}$ that is precontractive and so is pointwise convergent. Clearly then, $\mathcal{A}$ is pointwise convergent. Conversely, if $\mathcal{A}$ is pointwise convergent, then for each x in $\mathbb{F}^n$ we have

$$\lim_{k\to\infty} \left\{ \left[\prod_{i=k}^{1} A_{p(x)_i} \right] x \right\} = 0,$$

so for some k,

$$\left\| \left[\prod_{i=k}^{1} A_{p(x)_i} \right] x \right\|_V < \|x\|_V$$

and $\mathcal{A}$ is precontractive relative to $\|\cdot\|_V$. □

COROLLARY 3.3. *Precontractiveness is norm independent.*

Lemma 3.1 and Theorem 3.2 yield the following corollary.

COROLLARY 3.4. *$\mathcal{A}$ is pointwise convergent if and only if some finite subset of $\mathcal{A}$ is pointwise convergent.*

The condition $\rho(A) < 1$, necessary and sufficient for convergence of the single matrix A, generalizes to the following necessary and sufficient condition for uniform convergence of a matrix set.

THEOREM 3.5. *The set* $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$ *is uniformly convergent if and only if there is a finite sequence* $\{p_i\}_{i=1}^m$ *with* $p_i \in J$ *for* $i = 1, 2, \ldots, m$, *such that*

$$\rho\left(\prod_{i=m}^{1} A_{p_i}\right) < 1.$$

Proof. If the sequence $\{p_i\}_{i=1}^m$ exists as described, let $\{t_i\}_{i=1}^\infty$ be the periodic sequence $\{p_1, \ldots, p_m, p_1, \ldots, p_m, \ldots,\}$, let $\|\cdot\|_M$ be a (submultiplicative) matrix norm on $\mathbb{F}^{n\times n}$ such that

$$\left\|\prod_{i=m}^{1} A_{p_i}\right\|_M < 1,$$

let $\|\cdot\|_V$ be a norm on $\mathbb{F}^n$ consistent with $\|\cdot\|_M$, and let

$$C = \max\{1, \max\ \{\|A_{p_i}\|_M : 1 \le i \le m\}\}.$$

It follows that for any positive integer $k = mq + r$, with $0 \le r < m$, and any $x \in \mathbb{F}^n$, we have

$$\left\|\left[\prod_{i=k}^{1} A_{p_i}\right] x\right\|_V \le \left\|\prod_{i=m}^{1} A_{p_i}\right\|_M^q C^m \|x\|_V,$$

and hence $\mathcal{A}$ is uniformly convergent.

Conversely, if $\mathcal{A}$ is uniformly convergent with

$$\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} A_{p_i}\right] x\right\} = 0$$

for each $x \in \mathbb{F}^n$, then $\lim_{k\to\infty} \prod_{i=k}^{1} A_{p_i} = 0$, so for some sufficiently large m, $\rho(\prod_{i=m}^{1} A_{p_i}) < 1$. □

COROLLARY 3.6. *$\mathcal{A}$ is uniformly convergent if and only if there is a finite subset $\mathcal{B}$ of $\mathcal{A}$ that is uniformly convergent.*

COROLLARY 3.7. *If the set* $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$ *is uniformly convergent, then there is a* j *in* J *such that* $|\det A_j| < 1$.

Proof. Suppose that $|\det A_j| \ge 1$ for $j \in J$. Then, for any sequence $\{p_i\}_{i=1}^k$ with $p_i \in J$,

$$\left|\det\left(\prod_{i=k}^{1} A_{p_i}\right)\right| = \prod_{i=k}^{1} |\det A_{p_i}| \ge 1.$$

Hence

$$\rho\left(\prod_{i=k}^{1} A_{p_i}\right) \ge 1,$$

so by the theorem, $\mathcal{A}$ is not uniformly convergent. □

For $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$, the condition $|\det A_j| < 1$ for some j is not necessary for pointwise convergence. For example, if

$$A_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix},$$

then $\det A_1 = \det A_2 = 1$. However, as shown in [6], the set $\mathcal{A} = \{A_1, A_2\}$ is pointwise convergent. Intuitively, the matrix A_1 shrinks vectors in a cone centered on the x-axis (relative to the euclidean norm), whereas A_2 is a rotation that can be applied to vectors not in that cone to bring them into the cone without changing their length in the process. Hence, the set is precontractive and so pointwise convergent. In fact, small perturbations of A_1 and A_2 produce a set in which each matrix has determinant strictly greater than one, but which is still pointwise convergent.

A necessary condition for pointwise convergence is provided by the singular values of the A_i. We need the following lemma, which follows easily from the singular value decomposition of A.

LEMMA 3.8. *Let $A \in \mathbb{F}^{n\times n}$. A has a singular value less than one if and only if there is an $x \in \mathbb{F}^n$ such that $\|Ax\|_2 < \|x\|_2$.*

THEOREM 3.9. *Let $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$. If each singular value of each A_j is greater than or equal to 1, then for each $x \in \mathbb{F}^n$ and each sequence $\{p_i\}_{i=1}^k$ with $p_i \in J$, we have*

$$\left\| \left[\prod_{i=k}^{1} A_{p_i} \right] x \right\|_2 \geq \|x\|_2 .$$

Proof. Using the lemma,

$$\left\| \left[\prod_{i=k}^{1} A_{p_i} \right] x \right\|_2 \geq \left\| \left[\prod_{i=k-1}^{1} A_{p_i} \right] x \right\|_2 \geq \cdots \geq \|x\|_2 . \qquad \square$$

COROLLARY 3.10. *Let $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$. If $\mathcal{A}$ is pointwise convergent, then some A_j has a singular value less than one.*

Proof. The corollary follows from Theorems 3.2 and 3.9. $\square$

The existence of an eigenvalue of an A_j with modulus less than one is not necessary for pointwise convergence in the real case. Consider the example

$$\mathcal{A} = \{A_1, A_2\} \subset \mathbb{R}^{2\times 2},$$

with

$$A_1 = \begin{bmatrix} 1.0047 & -.7734 \\ -.0027 & -1.7896 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix},$$

and with θ selected as follows. The singular values of A_1 are .9 and 2, so by Lemma 3.8 there is an $x \in \mathbb{R}^2$ such that $\|A_1 x\|_2 < \|x\|_2$. It follows that there is a cone of vectors in $\mathbb{R}^2$ that contains x, each vector of which is shrunk by A_1 relative to $\|\cdot\|_2$. We choose θ so that each nonzero $x \in \mathbb{R}^2$ can be rotated into that cone, and we have $\mathcal{A}$ pointwise convergent. However, the eigenvalues of A_1 are 1.0054 and -1.7903, whereas those of A_2 both have modulus 1. Again, by applying a small perturbation, one can construct a pointwise convergent set in which each eigenvalue of each matrix exceeds one in absolute value.

Finally, in this section we observe that convergence is invariant under simultaneous similarity.

LEMMA 3.11. *Let $\mathcal{A} = \{A_j : j \in J\} \subset \mathbb{F}^{n\times n}$, and let $S \in \mathbb{F}^{n\times n}$ be nonsingular. Then $\mathcal{A}$ is pointwise (uniformly) convergent if and only if $\mathcal{A}_S = \{SA_jS^{-1} : j \in J\}$ is pointwise (uniformly) convergent.*

Proof. The proof for either pointwise or uniform convergence follows from the fact that if $y = S^{-1}x$ and $\{p_i\}_{i=1}^{\infty}$ is any index sequence, then

$$\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} SA_{p_i}S^{-1}\right]x\right\} = S\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} A_{p_i}\right]y\right\}. \qquad \square$$

4. Sets of diagonal matrices. We first establish the equivalence of pointwise and uniform convergence for sets of diagonal matrices.

THEOREM 4.1. *Let $\mathcal{D} = \{D_j : j \in J\} \subset \mathbb{F}^{n\times n}$ with each D_j a diagonal matrix. If $\mathcal{D}$ is convergent at $x = [x_1 \quad x_2 \cdots x_n]^T$ with $x_i \neq 0$ for $i = 1, 2, \ldots, n$, then $\mathcal{D}$ is uniformly convergent.*

Proof. We first consider the case $x = e = [1 \quad 1 \cdots 1]^T$. If $\mathcal{D}$ is convergent at e, there is a finite product D of matrices in $\mathcal{D}$ such that $\|De\|_\infty < \|e\|_\infty$, where $\|x\|_\infty = \max\{|x_i| : 1 \leq i \leq n\}$ is the infinity norm of x. Since D is diagonal, the entries of De are the eigenvalues of D. Thus, the inequality $\|De\|_\infty < \|e\|_\infty$ may be written $\rho(D) < 1$, and so $\mathcal{D}$ is uniformly convergent by Theorem 3.5.

Now consider a general $x = [x_1 \quad x_2 \cdots x_n]^T$ with $x_i \neq 0$ for $i = 1, 2, \ldots, n$ and suppose $\mathcal{D}$ converges at x; that is,

$$\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} D_{p(x)_i}\right]x\right\} = 0.$$

Let $D_x = \operatorname{diag}\{x_1, x_2, \ldots, x_n\}$, so that $D_x e = x$.

Then

$$\begin{aligned}\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} D_{p(x)_i}\right]e\right\} &= \lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} D_{p(x)_i}\right]D_x^{-1}D_x e\right\}\\ &= D_x^{-1}\lim_{k\to\infty}\left\{\left[\prod_{i=k}^{1} D_{p(x)_i}\right]x\right\} = 0,\end{aligned}$$

so $\mathcal{D}$ converges at e and hence converges uniformly. $\square$

COROLLARY 4.2. *A set of diagonal matrices is pointwise convergent if and only if it is uniformly convergent.*

Accordingly, we do not modify the word *convergence* when referring to a set of diagonal matrices.

We now develop a test for convergence of a finite set of diagonal matrices. We remark that, because of Corollary 3.6, the test is also relevant to infinite sets of diagonal matrices. Through the remainder of this section, we assume that $\mathcal{D}$ is finite.

DEFINITION 4.3. *The matrix $A \in \mathbb{R}^{m\times n}$ is* semipositive *provided there is an x in $\mathbb{R}^n$ with $x \geq 0$ such that $Ax > 0$, the inequalities denoting entrywise inequality.*

A more general semipositivity is discussed in [2] and semipositivity as defined here is discussed in [5]. In the following discussion, $\mathbb{N}_+^n$ denotes the set of n-tuples of nonnegative integers and $\mathbb{R}_+^n$ denotes the nonnegative orthant in $\mathbb{R}^n$.

LEMMA 4.4. *Suppose that $A \in \mathbb{R}^{m\times n}$. Then A is semipositive if and only if there is a $k \in \mathbb{N}_+^n$ such that $Ak > 0$.*

Proof. Clearly the existence of k implies A semipositive.

Suppose A is semipositive; choose $x \in \mathbb{R}_+^n$ such that $Ax > 0$. Since the map $y \to Ay$ is continuous, there is an open set $U \subset \mathbb{R}^n$ containing x such that $y \in U$

implies $Ay > 0$. We may choose a member q of $U \cap \mathbb{R}^n_+$, all of whose entries are rational, and for an appropriately chosen positive integer c, $k = cq$ is in $\mathbb{N}^n_+$, and we have $Ak = cAq > 0$. □

THEOREM 4.5. *Let* $\mathcal{D} = \{D_1, D_2, \ldots, D_m\} \subset \mathbb{F}^{n\times n}$ *with* $D_j = \operatorname{diag}\{d_{1j}, d_{2j}, \ldots, d_{nj}\}$, $j = 1, 2, \ldots, m$, *and suppose* $d_{ij} \neq 0$ *for* $1 \leq i \leq n, 1 \leq j \leq m$. *Let* $L(\mathcal{D}) = [l_{ij}] \in \mathbb{R}^{n\times m}$ *be defined by*

$$\ell_{ij} = -\ln\ |d_{ij}|\,.$$

Then $\mathcal{D}$ *is convergent if and only if* $L(\mathcal{D})$ *is semipositive. Furthermore, if this is the case and if* $k \in \mathbb{N}^n_+$ *satisfies* $L(\mathcal{D})k > 0$, *then* $\rho(D_1^{k_1} D_2^{k_2}, \ldots, D_m^{k_m}) < 1$.

Proof. It holds that $\mathcal{D}$ is convergent if and only if there is a sequence $\{p_i\}_{i=1}^t$ such that $1 \leq p_i \leq m$ and

$$\rho\left(\prod_{i=t}^{1} D_{p_i}\right) < 1.$$

Since $\mathcal{D}$ is a commutative set, $\{p_i\}_{i=1}^t$ exists if and only if there is a $k = [k_1, k_2, \ldots, k_m]^T \in \mathbb{N}^m_+$, with $\sum_{j=1}^m k_j = t$, such that

$$\rho\left(D_1^{k_1} D_2^{k_2} \cdots D_m^{k_m}\right) < 1.$$

This inequality is equivalent to

$$\prod_{j=1}^{m} |d_{ij}|^{k_j} < 1 \quad \text{for } i = 1, 2, \ldots, n.$$

Taking logarithms, we obtain the equivalent linear system

$$\sum_{j=1}^{m} k_j \ln |d_{ij}| < 0, \qquad i = 1, 2, \ldots, n$$

or, equivalently, $L(\mathcal{D})k > 0$. Thus, by Lemma 4.4, $\mathcal{D}$ is convergent if and only if $L(\mathcal{D})$ is semipositive, and in this case the stated inequality holds. □

It is easily seen that a set $\mathcal{D} = \{D_1, D_2, \ldots, D_m\}$ of diagonal matrices, in which there are some zero entries on diagonals, is convergent if and only if the set $\hat{\mathcal{D}}$ is convergent. Here $\hat{\mathcal{D}}$ is obtained from $\mathcal{D}$ by deleting from each D_j the ith row and ith column provided there is a k such that $d_{ik} = 0$. This observation leads to the following corollary in which $L(\mathcal{D})$ is defined consistently with its definition in Theorem 4.5.

COROLLARY 4.6. *Let* $\mathcal{D} = \{D_1, D_2, \ldots, D_m\} \subset \mathbb{F}^{n\times n}$ *with* $D_j = \operatorname{diag}\{d_{1j}, d_{2j}, \ldots, d_{nj}\}$, $1 \leq j \leq m$. *Let* $L(\mathcal{D}) = [l_{ij}] \in \mathbb{F}^{n\times m}$ *be defined by*

$$l_{ij} = \begin{cases} -\ln\ |d_{ij}| & \text{if } d_{ik} \neq 0 \quad \text{for } k = 1, 2, \ldots, m, \\ 1 & \text{if there is a } k \text{ such that } d_{ik} = 0. \end{cases}$$

Then $\mathcal{D}$ *is convergent if and only if* $L(\mathcal{D})$ *is semipositive and the final sentence of Theorem* 4.5 *remains valid.*

Proof. $\mathcal{D}$ is convergent if and only if $\hat{\mathcal{D}}$, described above, is convergent. $\hat{\mathcal{D}}$ is convergent if and only if $L(\hat{\mathcal{D}})$ is semipositive. Now $L(\mathcal{D})$ may be obtained from $L(\hat{\mathcal{D}})$ by adjoining rows all of whose entries are one. Hence, for $x \in \mathbb{R}^n_+$, $L(\mathcal{D})x > 0$ if and only if $L(\hat{\mathcal{D}})x > 0$, and the corollary follows. □

We note here that, in view of Lemma 3.11, sets $\mathcal{A} = \{A_1, A_2, \ldots, A_m\} \subset \mathbb{F}^{n\times n}$, which are simultaneously diagonalizable, can be tested for convergence using Corollary 4.6.

5. Sets of triangular matrices. In this section we may interpret "triangular" as "upper triangular" throughout or as "lower triangular" throughout.

DEFINITION 5.1. *If $A \in \mathbb{F}^{n\times n}$, then* diag (A) *is the matrix obtained from A by replacing all off-diagonal entries with zeros.*

THEOREM 5.2. *The set $\mathcal{T} = \{T_j : j \in J\}$ of triangular matrices is uniformly convergent if and only if the set*

$$\text{diag } (\mathcal{T}) = \{\text{diag } (T_j) : j \in J\}$$

is convergent.

Proof. The result follows from Theorem 3.5 and the fact that for any sequence $\{p_i\}_{i=1}^k$, the eigenvalues of $\prod_{i=k}^1 T_{p_i}$ are the same as those of $\prod_{i=k}^1 \text{diag } (T_{p_i})$. □

We remark, however, that for sets of triangular matrices, pointwise and uniform convergence are not the same. Let $\mathcal{T} = \{T_1, T_2\} \subset \mathbb{R}^{2\times 2}$, where

$$T_1 = \begin{bmatrix} \frac{3}{2} & 0 \\ 0 & \frac{2}{3} \end{bmatrix} \quad \text{and} \quad T_2 = \begin{bmatrix} \frac{2}{3} & 0 \\ 1 & -\frac{3}{2} \end{bmatrix}.$$

We show that $\mathcal{T}$ is precontractive relative to $\|\cdot\|_2$ and hence pointwise convergent. In fact, we claim that for $x \in \mathbb{R}^2$ and $x \neq 0$, one of the following is true:

(1) $\|T_1 x\|_2 < \|x\|_2$,
(2) $\|T_2 x\|_2 < \|x\|_2$,
(3) $\|T_2 T_1 T_2 x\|_2 < \|x\|_2$,
(4) $\|T_2 T_1^2 T_2 x\|_2 < \|x\|_2$.

It is sufficient to show that one of (1)–(4) holds when $x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ or $x = \begin{bmatrix} 1 \\ y \end{bmatrix}$ for some $y \in \mathbb{R}$. Clearly, (1) holds when $x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. For $i = 1, 2, 3, 4$, let $S_i = \{y \in \mathbb{R} : (i) \text{ holds for } x = \begin{bmatrix} 1 \\ y \end{bmatrix}\}$. Computation shows that

$$S_1 = \left\{ y : |y| > \frac{3}{2} \right\},$$

$$S_2 = \left\{ y : \frac{6}{5} - \frac{2\sqrt{61}}{15} < y < \frac{6}{5} + \frac{2\sqrt{61}}{15} \right\} \supset \left\{ y : \frac{1}{2} < y < 2 \right\},$$

$$S_3 = \left\{ y : |y| < \frac{3}{2} \right\},$$

$$S_4 = \left\{ y : y < -\frac{5}{12} \right\}.$$

Hence, $\bigcup_{i=1}^4 S_i = \mathbb{R}$ and $\mathcal{T}$ is pointwise convergent.

However, $\mathcal{T}$ is not uniformly convergent, as we see by use of Theorems 5.2 and 4.5. Let $D_i = \text{diag } (T_i)$ for $i = 1, 2$, so that diag $(\mathcal{T}) = \{D_1, D_2\}$. Then

$$L(\text{diag } (\mathcal{T})) = \begin{bmatrix} \ln \frac{2}{3} & \ln \frac{3}{2} \\ \ln \frac{3}{2} & \ln \frac{2}{3} \end{bmatrix}$$

is not semipositive, since it has sign pattern $\begin{bmatrix} - & + \\ + & - \end{bmatrix}$ and its determinant is zero [5]. Hence, diag $(\mathcal{T})$ is not convergent, so $\mathcal{T}$ is not uniformly convergent.

We obtain a necessary condition for pointwise convergence of a set of triangular matrices.

THEOREM 5.3. *Suppose* $\mathcal{T} = \{T_j : j \in J\} \subset \mathbb{F}^{n\times n}$ *with each* T_j *triangular. If* $\mathcal{T}$ *is pointwise convergent, then for* $1 \leq i \leq n$, *there is a* j *in* J *such that* $|\mathrm{ent}_{ii}\,(T_j)| < 1$.

Proof. Suppose there exists an i such that

$$|\mathrm{ent}_{ii}\,(T_j)| \geq 1 \quad \text{for } j \in J.$$

Then no finite product P of the T_j's would satisfy $\|Pe_i\|_2 < \|e_i\|_2$, where e_i is the ith canonical basis vector in $\mathbb{F}^n$, so $\mathcal{T}$ is not precontractive relative to $\|\cdot\|_2$ and hence is not pointwise convergent. □

Finally, we note that results obtained for sets of triangular matrices can be used for any commuting set of matrices via simultaneous triangularization and Lemma 3.11 (see [4, Thm. 2.3.3].)

REFERENCES

[1] M. A. BERGER AND Y. WANG, *Bounded semigroups of matrices*, Linear Algebra Appl., 166 (1992), pp. 21–27.

[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[3] I. DAUBECHIES AND J. C. LAGARIAS, *Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 161 (1992), pp. 227–263.

[4] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, London, 1985.

[5] C. R. JOHNSON, M. K. KERR, AND D. P. STANFORD, *Semipositivity of Matrices*, Linear Multilinear Algebra, to appear.

[6] D. P. STANFORD, *Stability for a multi-rate sampled-data system*, SIAM J. Control Optim., 17 (1979), pp. 390–399.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1141–1156, October 1994

007

A HYBRID ALGORITHM FOR OPTIMIZING EIGENVALUES OF SYMMETRIC DEFINITE PENCILS*

JEAN-PIERRE A. HAEBERLY† AND MICHAEL L. OVERTON‡

Abstract. An algorithm is presented for the optimization of the maximum eigenvalue of a symmetric definite pencil depending affinely on a vector of parameters. The algorithm uses a hybrid approach, combining a scheme based on the method of centers, developed by Boyd and El Ghaoui [*Linear Algebra Appl.*, 188 (1993), pp. 63–112], with a new quadratically convergent local scheme. A convenient expression for the generalized gradient of the maximum eigenvalue of the pencil is also given, expressed in terms of a dual matrix. The algorithm computes the dual matrix that establishes the optimality of the computed solution.

Key words. nonsmooth optimization, generalized eigenvalue problem, matrix pencil, Lyapunov equations

AMS subject classifications. 15A22, 15A42, 49A52, 65F15, 90C26

1. Introduction.

In this paper we consider the problem of minimizing the maximum eigenvalue of a symmetric definite pencil $(A(x), B(x))$ depending on a vector parameter $x \in \Re^m$. Many problems arising in control theory can be formulated in these terms. Most notable among these are the computation of bounds for structured singular values and the computation of structured Lyapunov functions [3]. The salient feature of this problem is the lack of smoothness. Indeed, it is well known that the eigenvalues of a matrix are not differentiable as functions of the entries of the matrix when their multiplicity exceeds one. Furthermore, at an optimum point, the multiplicity of the maximum eigenvalue is often greater than one. Thus standard optimization techniques cannot be applied.

The special case $B = I$, which is the problem of minimizing the maximum eigenvalue of a symmetric matrix $A(x)$, has been studied extensively (see [14] and the references therein, as well as [1], [5], [9]). The case where $B(x)$ is constant is entirely similar and is briefly discussed in [14]. Algorithms have been developed in [2], [11] and [12] to solve the problem when the pencil depends affinely on the parameter vector and in [8] for the general case. The algorithm of Boyd and El Ghaoui in [2] is based on the method of centers and exhibits very good global behavior but slow local convergence.

We propose a hybrid algorithm, combining the robustness of the method of centers with rapid local convergence, to efficiently solve the affine case. More precisely, we propose to follow the path of centers to the vicinity of a solution and then to switch to a quadratically convergent local scheme. Such an approach was suggested in [2]. It should be noted that any algorithm exhibiting good global behavior could be used in place of the method of centers, e.g., any algorithm based on interior point methods. The local algorithm results from an extension of the work presented in [13], [14], and [16] to a pencil-valued function $(A(x), B(x))$. The algorithm is implemented to take

* Received by the editors February 12, 1993; accepted for publication (in revised form) June 11, 1993.

† Department of Mathematics, Fordham University, Bronx, New York 10458 (haeberly@murray.fordham.edu).

‡ Computer Science Department, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, New York 10012 (overton@cs.nyu.edu). The work of this author was supported in part by National Science Foundation grant CCR-9101649.

full advantage of a block diagonal structure, if present. Block diagonal pencils occur frequently in applications from control theory [3].

The paper is organized as follows. Some notation and conventions are introduced in §2. The generalized gradient and directional derivatives of the maximum eigenvalue are derived in §3 and optimality conditions are stated. The local algorithm is described in §4 and the global algorithm in §5. Two numerical examples are presented in §6.

2. Preliminaries. Let $\mathcal{SR}^{n\times n}$ denote the set of $n \times n$ real symmetric matrices. For $B \in \mathcal{SR}^{n\times n}$, the notation $B \geq 0$ means that B is nonnegative definite, and $B > 0$ means that B is positive definite. A symmetric definite pencil (A, B) consists of a pair of matrices A, B in $\mathcal{SR}^{n\times n}$ with $B > 0$. An eigenvalue λ of (A, B) satisfies $\det(A - \lambda B) = 0$, i.e., there is a nonzero eigenvector q satisfying $Aq = \lambda Bq$.

We write $\langle\, , \rangle$ for the Frobenius inner product on the space of $n \times n$ matrices. Thus

$$\langle M, N\rangle = \operatorname{tr} M^T N.$$

Let "vec" denote the operator mapping $\mathcal{SR}^{n\times n}$ into $\mathcal{R}^{n(n+1)/2}$ defined by

$$\operatorname{vec} A = (a_{11}, \sqrt{2}a_{12}, \ldots, \sqrt{2}a_{1n}, a_{22}, \sqrt{2}a_{23}, \ldots, \sqrt{2}a_{2n}, a_{33}, \sqrt{2}a_{34}, \ldots, a_{nn})$$

for $A = (a_{ij}) \in \mathcal{SR}^{n\times n}$. Observe that for two symmetric matrices M and N, we have

$$(\operatorname{vec} M)^T (\operatorname{vec} N) = \langle M, N\rangle.$$

Also, let

$$C_1 \oplus \cdots \oplus C_k$$

denote the block diagonal matrix with blocks $C_1, \ldots, C_k$.

Given a symmetric definite pencil (A, B), let G denote the Choleski factor of B. Hence G is a lower triangular matrix with positive diagonal entries such that $GG^T = B$. Then the symmetric matrix $G^{-1}AG^{-T}$ has the same eigenvalues as the pencil. Let us write $\lambda_{\max} = \lambda_1 \geq \cdots \geq \lambda_n$ for these eigenvalues and Λ for the diagonal matrix

$$\Lambda = \operatorname{Diag}(\lambda_1, \ldots, \lambda_n).$$

Let $p_1, p_2, \ldots, p_n$ denote a set of orthonormal eigenvectors for $G^{-1}AG^{-T}$, and let $P = [p_1 \ldots p_n]$ denote the orthogonal matrix with columns p_i, $1 \leq i \leq n$. Then the columns of the matrix $Q = G^{-T}P = [G^{-T}p_1, \ldots, G^{-T}p_n]$ are eigenvectors for the pencil. Hence we have

$$P^T G^{-1} A G^{-T} P = \Lambda, \qquad P^T P = I,$$

and

$$AQ = BQ\Lambda, \qquad Q^T B Q = I.$$

We have the following characterization of $\lambda_{\max}$ [14]:

$$\lambda_{\max} = \max\{\langle U, G^{-1}AG^{-T}\rangle \mid U \in \mathcal{SR}^{n\times n},\ \operatorname{tr} U = 1,\ U \geq 0\}. \tag{1}$$

In particular, we see that $\lambda_{\max}$ is a convex function of A; however, it is only quasi-convex as a function of B [2].

3. Optimality conditions. Consider a symmetric definite pencil-valued function $(A(x), B(x))$ of a vector of real parameters $x \in \Re^m$. We assume that $A(x)$ and $B(x)$ are twice continuously differentiable in x and we write $A_j(x)$, $B_j(x)$ for the partial derivatives of $A(x)$ and $B(x)$ with respect to x_j. Note that a vector subscript (e.g., x_j) denotes a component, while a matrix subscript indicates differentiation. Later, we shall use matrix superscripts to denote diagonal blocks.

Let $G(x)$ denote the Choleski factor of $B(x)$ and let $L_j(x)$ denote the partial derivative of $G(x)$ with respect to x_j. Thus $L_j(x)$ is the unique lower triangular matrix that solves the equation

$$L_j(x)G^T(x) + G(x)L_j^T(x) = B_j(x). \tag{2}$$

Note that this is a particularly simple Lyapunov equation since the matrix $G(x)$ is lower triangular. Since

$$\frac{\partial}{\partial x_j} G^{-1}(x) = -G^{-1}(x)L_j(x)G^{-1}(x),$$

we have

$$\frac{\partial}{\partial x_j}(G^{-1}(x)A(x)G^{-T}(x)) = G^{-1}(x)Z_j(x)G^{-T}(x),$$

where the matrix $Z_j(x)$ is given by

$$Z_j(x) \equiv A_j(x) - L_j(x)G^{-1}(x)A(x) - A(x)G^{-T}(x)L_j^T(x). \tag{3}$$

For future reference, we also define

$$L_{ij}(x) = \frac{\partial}{\partial x_i}(L_j(x)). \tag{4}$$

Thus $L_{ij}(x)$ is that lower triangular matrix that solves the following equation:

$$L_{ij}(x)G^T(x) + G(x)L_{ij}^T(x) = B_{ij}(x) - [L_i(x)L_j^T(x) + L_j(x)L_i^T(x)].$$

Clearly, $L_{ij}(x) = L_{ji}(x)$.

Finally, write

$$\lambda_{\max}(x) = \lambda_1(x) \geq \cdots \geq \lambda_n(x)$$

for the eigenvalues of $(A(x), B(x))$, let $P(x)$ be an orthogonal matrix of eigenvectors for $G^{-1}(x)A(x)G^{-T}(x)$, so that

$$P^T(x)G^{-1}(x)A(x)G^{-T}(x)P(x) = \Lambda(x)$$

and define $Q(x) = G^{-1}(x)P(x)$.

We are now ready to derive the generalized gradient of $\lambda_{\max}(x)$. (See Gollan [18, Thm. 6.1] for a related result.)

THEOREM 3.1. *Assume that the multiplicity of $\lambda_{\max}(x)$ is t and let $Q_{\max}(x)$ be the submatrix of $Q(x)$ whose columns form a complete set of eigenvectors for $\lambda_{\max}(x)$. Then the generalized gradient of $\lambda_{\max}(x)$ is the set*

$$\partial\lambda_{\max}(x) = \left\{v \in \Re^m \mid v_j = \langle U, Q_{\max}^T(x)\,(A_j(x) - \lambda_{\max}(x)B_j(x))\,Q_{\max}(x)\rangle\right\}, \tag{5}$$

where U runs over all nonnegative definite symmetric $t \times t$ matrices with $\operatorname{tr} U = 1$.

Proof. Let C denote an $n \times n$ symmetric matrix. The generalized gradient of the maximum eigenvalue $\lambda_{\max} = \lambda_{\max}(C)$ viewed as a function of C was given in [14]. If the multiplicity of $\lambda_{\max}(C)$ is t and if $P_{\max}$ is an $n \times t$ matrix whose columns form a complete set of orthonormal eigenvectors for $\lambda_{\max}(C)$, then

$$\partial\lambda_{\max}(C) = \{V \in \mathcal{SR}^{n\times n} \mid V = P_{\max} U P_{\max}^T\},$$

where U runs over the $t \times t$ symmetric matrices with $U \geq 0$ and $\operatorname{tr} U = 1$. We call U a *dual* matrix. The generalized gradient of the maximum eigenvalue $\lambda_{\max}(x)$ of the symmetric matrix $G^{-1}(x)A(x)G^{-T}(x)$ now follows from the chain rule [4]. We get

$$\partial\lambda_{\max}(x) = \{v \in \mathcal{R}^m \mid v_j = \langle V, G^{-1}(x)Z_j(x)G^{-T}(x)\rangle\}, \tag{6}$$

where $V \in \partial\lambda_{\max}(A(x))$. We have an equality in (6) instead of a mere inclusion because $\lambda_{\max}(x)$ is a regular function [4]. Indeed, (1) expresses $\lambda_{\max}(x)$ as the maximum over the matrices V of $\langle V, G^{-1}(x)A(x)G^{-T}(x)\rangle$, and, for a fixed V, the function

$$x \mapsto \langle V, G^{-1}(x)A(x)G^{-T}(x)\rangle$$

is differentiable, hence regular [4, Thms. 2.8.2 and 2.8.6.]. We write $P_{\max} = P_{\max}(x)$ and $Q_{\max} = Q_{\max}(x) = G^{-T}(x)P_{\max}(x)$. Observe that

$$\langle P_{\max} U P_{\max}^T, G^{-1}(x)Z_j(x)G^{-T}(x)\rangle = \langle U, P_{\max}^T G^{-1}(x)Z_j(x)G^{-T}(x)P_{\max}\rangle.$$

Now the columns of $Q_{\max}$ are generalized eigenvectors for $\lambda_{\max} \equiv \lambda_{\max}(x)$ and we have

$$\begin{aligned}
Q_{\max}^T Z_j(x) Q_{\max} &= Q_{\max}^T [A_j(x) - L_j(x)G^{-1}(x)A(x) \\
&\quad - A(x)G^{-T}(x)L_j^T(x)]Q_{\max} \\
&= Q_{\max}^T [A_j(x) - \lambda_{\max}(L_j(x)G^{-1}(x)B(x) \\
&\quad + B(x)G^{-T}(x)L_j^T(x))]Q_{\max} \\
&= Q_{\max}^T [A_j(x) - \lambda_{\max}(L_j(x)G^T(x) + G(x)L_j^T(x))]Q_{\max} \\
&= Q_{\max}^T [A_j(x) - \lambda_{\max} B_j(x)]Q_{\max}.
\end{aligned}$$

The result follows. □

A necessary condition for x to minimize $\lambda_{\max}$ is that $0 \in \partial\lambda_{\max}(x)$ [4]. By Theorem 3.1 this can be rewritten as follows. Let t denote the multiplicity of $\lambda_{\max}(x)$. Then a necessary condition for x to minimize $\lambda_{\max}$ is that there exists a dual matrix $U \in \mathcal{SR}^{t\times t}$ such that

$$\operatorname{tr} U = 1, \tag{7}$$

$$U \geq 0, \tag{8}$$

and, for $1 \leq j \leq m$,

$$\langle U, Q_{\max}^T(x)\,(A_j(x) - \lambda_{\max}(x)B_j(x))\,Q_{\max}(x)\rangle = 0, \tag{9}$$

where $Q_{\max}(x)$ is as in Theorem 3.1.

As a corollary of Theorem 3.1, we can evaluate the directional derivative of $\lambda_{\max}(x)$.

COROLLARY 3.2. *Let $\lambda'_{\max}(x;d)$ denote the directional derivative of $\lambda_{\max}$ in the direction d, i.e.,*

$$\lambda'_{\max}(x;d) = \lim_{\epsilon \downarrow 0} \frac{\lambda_{\max}(x+\epsilon d) - \lambda_{\max}(x)}{\epsilon}.$$

Then $\lambda'_{\max}(x;d)$ is equal to the largest eigenvalue of the matrix

$$\sum_{j=1}^{m} d_j Q_{\max}^T(x) \left(A_j(x) - \lambda_{\max}(x) B_j(x)\right) Q_{\max}(x).$$

Proof. Since $\lambda_{\max}$ is regular at x the directional derivative $\lambda'_{\max}(x;d)$ exists and is given by [4, Prop. 2.1.2]

$$\lambda'_{\max}(x;d) = \max\{\langle v, d\rangle \mid v \in \partial\lambda_{\max}(x)\}.$$

Now for $v \in \partial\lambda_{\max}(x)$ we have, from (5),

$$\begin{aligned}
\langle v, d\rangle &= \sum_{j=1}^{m} v_j d_j \\
&= \sum_{j=1}^{m} d_j \langle U, Q_{\max}^T(x) \left(A_j(x) - \lambda_{\max}(x) B_j(x)\right) Q_{\max}(x)\rangle \\
&= \langle U, \sum_{j=1}^{m} d_j Q_{\max}^T(x) \left(A_j(x) - \lambda_{\max}(x) B_j(x)\right) Q_{\max}(x)\rangle.
\end{aligned}$$

The result follows from (1). □

Now consider the three optimality conditions (7)–(9). Suppose that x and U are known to satisfy the first and third condition, but not necessarily the second condition $U \geq 0$. The following result shows the relationship between certain directional derivatives of $\lambda_{\max}(x)$ and the eigenvalues of the dual matrix U.

THEOREM 3.3. *Let x and U satisfy* (7) *and* (9), *and let θ be an eigenvalue of U with normalized eigenvector v. Suppose $d \in \Re^m$ and $\delta \in \Re$ form a solution of the following equation:*

$$\sum_{j=1}^{m} d_j Q_{\max}^T(x) \left(A_j(x) - \lambda_{\max}(x) B_j(x)\right) Q_{\max}(x) - \delta I = -vv^T. \tag{10}$$

Then the directional derivative $\lambda'_{\max}(x;d)$ is given by

$$\lambda'_{\max}(x;d) = \theta.$$

Proof. Let us write

$$M(d) = \sum_{j=1}^{m} d_j Q_{\max}^T(x) \left(A_j(x) - \lambda_{\max}(x) B_j(x)\right) Q_{\max}(x).$$

Then (10) gives

$$M(d) = \delta I - vv^T,$$

so that all the eigenvalues of $M(d)$ are equal to δ except one that equals $\delta - 1$. Thus $\lambda_1'(x;d) = \delta$ by Corollary 3.2. Now, taking the inner product of both sides of (10) with the matrix U yields

$$\sum_{j=1}^{m} d_j \langle U, Q_{\max}^T(x)\,(A_j(x) - \lambda_{\max}(x)B_j(x))\,Q_{\max}(x)\rangle - \delta \operatorname{tr} U = -\theta.$$

Since optimality conditions (7) and (9) are satisfied, this equation reduces to

$$\delta = \theta,$$

which completes the proof. □

The essential idea here is that the direction d splits the multiple eigenvalue $\lambda_{\max}$, but, to first order, the multiplicity is reduced only by one. Among other things, this result shows how to generate a descent direction for $\lambda_{\max}(x)$ if all optimality conditions are satisfied except $U \geq 0$: set θ equal to a negative eigenvalue of U. For further results of this kind, together with a discussion of solvability conditions for systems of the form (10), see [15].

4. The local algorithm. In this section we develop a quadratically convergent local algorithm to minimize the maximum eigenvalue of the pencil $(A(x), B(x))$, under the assumption that $A(x)$ and $B(x)$ are affine functions of x. Thus

$$A(x) = A_0 + x_1A_1 + \cdots + x_mA_m$$

and

$$B(x) = B_0 + x_1B_1 + \cdots + x_mB_m.$$

We discuss the nonlinear case briefly at the end of this section. We further assume, as is the case in many applications, that the pencil $(A(x), B(x))$ is block diagonal, with l denoting the number of blocks. Thus

$$A(x) = A^1(x) \oplus \cdots \oplus A^l(x), \qquad B(x) = B^1(x) \oplus \cdots \oplus B^l(x)$$

with $A^i(x)$, $B^i(x)$ in $\mathcal{SR}^{n_i \times n_i}$, $B^i(x) > 0$, for $1 \leq i \leq l$. Let $n = n_1 + \cdots + n_l$ be the dimension of the pencil and let $\lambda_{\max}(x)$ denote the largest eigenvalue of $(A(x), B(x))$.

Given an initial point x^0 in a neighborhood of a local minimizer x^* of $\lambda_{\max}(x)$, we show how to generate a sequence of iterates converging quadratically to x^*. Let $\hat{x}$ denote the current point. The new iterate is set to $\hat{x} + \hat{d}$ where $\hat{d}$ is the solution of a certain equality-constrained quadratic program. Let $t = (t_1, \ldots, t_l)$ be the vector of multiplicities for $\lambda_{\max}^* \equiv \lambda_{\max}(x^*)$, i.e., $t_j \geq 0$ is the multiplicity of $\lambda_{\max}^*$ in the block $(A^j(x^*), B^j(x^*))$. Of course, t is not known and needs to be estimated. This is done with the use of a multiplicity tolerance τ, based on the eigenvalues at the current iterate $\hat{x}$. More precisely, we write $\hat{\lambda}_{\max} = \lambda_{\max}(\hat{x})$, $\hat{\lambda}_i = \lambda_i(\hat{x})$, we order the eigenvalues of each block in decreasing order,

$$\hat{\lambda}_1^j \geq \cdots \geq \hat{\lambda}_{n_j}^j, \qquad 1 \leq j \leq l,$$

and we define $t_j \geq 0$ by

$$\hat{\lambda}_{\max} - \hat{\lambda}_{t_j+1}^j > \tau \max(1, |\hat{\lambda}_{\max}|),$$

and, if $t_j \neq 0$,

$$\hat{\lambda}_{\max} - \hat{\lambda}^j_{t_j} \leq \tau \max(1, |\hat{\lambda}_{\max}|).$$

We write

$$\hat{\Lambda}^j_{\max} = \mathrm{Diag}(\hat{\lambda}^j_1, \ldots, \hat{\lambda}^j_{t_j}),$$

for the diagonal matrix consisting of the first t_j eigenvalues of the jth block. If $t_j = 0$, $\hat{\Lambda}^j_{\max}$ is an empty matrix. Let

$$\hat{\Lambda}^j_{\mathrm{rest}} = \mathrm{Diag}(\hat{\lambda}^j_{t_j+1}, \ldots, \hat{\lambda}^j_{n_j})$$

and

$$\hat{\Lambda}^j = \hat{\Lambda}^j_{\max} \oplus \hat{\Lambda}^j_{\mathrm{rest}}.$$

Let $\hat{Q}$ be a matrix whose columns form a complete set of eigenvectors for the pencil at $\hat{x}$. Clearly $\hat{Q}$ is also block diagonal,

$$\hat{Q} = \hat{Q}^1 \oplus \cdots \oplus \hat{Q}^l.$$

We now write $\hat{Q}_{\max}$ for the matrix obtained from $\hat{Q}$ by discarding all but the first t_j eigenvectors of the jth block, for $1 \leq j \leq l$. It is also block diagonal, i.e.,

$$\hat{Q}_{\max} = \hat{Q}^1_{\max} \oplus \cdots \oplus \hat{Q}^l_{\max},$$

where $\hat{Q}^j_{\max}$, of size $n_j \times t_j$, corresponds to the jth block and is empty if $t_j = 0$. Finally let $\hat{Q}^j_{\mathrm{rest}}$ be such that

$$\hat{Q}^j = [\hat{Q}^j_{\max}\ \hat{Q}^j_{\mathrm{rest}}].$$

The quadratic program that is solved to compute the step $\hat{d}$ is given as follows:

$$\min_{d,\omega}\ \omega + \frac{1}{2} d^T W d \tag{11}$$

subject to

$$K \begin{pmatrix} \omega \\ d \end{pmatrix} = h, \tag{12}$$

where $\omega \in \Re$ and W is a positive definite matrix that is described below. The matrix K is given by

$$K = \begin{bmatrix} K^1 \\ \vdots \\ K^l \end{bmatrix},$$

where K^j is the matrix

$$[\mathrm{vec}I \quad -\mathrm{vec}(\hat{Q}^{j\,T}_{\max}(\hat{A}^j_1 - \hat{\lambda}_{\max}\hat{B}^j_1)\hat{Q}^j_{\max}) \cdots \quad -\mathrm{vec}(\hat{Q}^{j\,T}_{\max}(\hat{A}^j_m - \hat{\lambda}_{\max}\hat{B}^j_m)\hat{Q}^j_{\max})],$$

with I the identity matrix of size $t_j \times t_j$ and $\hat{A}_i^j = \hat{A}_i^j(\hat{x})$, $\hat{B}_i^j = \hat{B}_i^j(\hat{x})$. Clearly, K^j is empty if $t_j = 0$. The vector h is given by

$$h = \begin{bmatrix} h^1 \\ \vdots \\ h^l \end{bmatrix},$$

where

$$h^j = \operatorname{vec}(\hat{\Lambda}^j_{\max} - \hat{\lambda}_{\max} I)$$

if $t_j > 0$ and h^j is empty otherwise.

The formulation of the quadratic program can be motivated as follows. The constraints (12) consist of

$$\sum_{j=1}^{l} \frac{t_j(t_j+1)}{2} \tag{13}$$

scalar constraints. The multipliers of these scalar constraints can be assembled into a block diagonal dual matrix estimate

$$U = U^1 \oplus \cdots \oplus U^l$$

with U^j empty if $t_j = 0$. If (d, ω) is a solution of (11), (12), then the optimality conditions for quadratic programs give

$$\sum_{j=1}^{l} \langle U^j, \hat{Q}^{j\,T}_{\max} \left(\hat{A}_i^j - \hat{\lambda}_{\max} \hat{B}_i^j \right) \hat{Q}^j_{\max} \rangle + (Wd)_i = 0 \tag{14}$$

for $1 \le i \le m$ and

$$\operatorname{tr} U = \sum_{j=1}^{l} \operatorname{tr} U^j = 1. \tag{15}$$

Now observe that if the multiplicity tolerance $\tau = 0$, the vector h is zero, and the constraint (12) can be rewritten as

$$\sum_{i=1}^{m} d_i \, \hat{Q}^{j\,T}_{\max} \left(\hat{A}_i^j - \hat{\lambda}_{\max} \hat{B}_i^j \right) \hat{Q}^j_{\max} = \omega I, \qquad 1 \le j \le l.$$

Since the matrices $\hat{Q}^{j\,T}_{\max}(\hat{A}_i^j - \hat{\lambda}_{\max}\hat{B}_i^j)\hat{Q}^j_{\max}$ are the diagonal blocks of the matrix $Q^T_{\max}(\hat{x})(A_i(\hat{x}) - \lambda_{\max}(\hat{x})B_i(\hat{x}))Q_{\max}(\hat{x})$ of Corollary 3.2, we conclude that a solution (d, ω) of (12) satisfies

$$\lambda'_{\max}(\hat{x}; d) = \omega.$$

If, moreover, (d, ω) is a solution of (11) then

$$\omega + \frac{1}{2} d^T W d \le 0,$$

since $(0,0)$ is feasible. But W is positive definite so that

$$\lambda'_{\max}(\hat{x};d) \leq -\frac{1}{2}d^T W d \leq 0.$$

Hence d is a descent direction for $\lambda_{\max}$ unless $d = 0$. But if $d = 0$ then (14) and (15) show that all optimality conditions for $\lambda_{\max}$ are satisfied at $\hat{x}$ except possibly the nonnegative definite condition on U. If U does have a negative eigenvalue, Theorem 3.3 shows how to obtain a descent direction by splitting $\lambda_{\max}$.

If the multiplicity vector t of $\lambda_{\max}$ at the optimal value x^* is known, the problem of minimizing $\lambda_{\max}$ over x can be rewritten as

$$\min_{x,\omega} \quad \omega$$

subject to

$$\lambda_1^j(x) = \cdots = \lambda_{t_j}^j(x) = \omega, \qquad 1 \leq j \leq l. \tag{16}$$

The objective function is now a smooth function of x and ω, but the constraints, of course, are not. Following [16], we replace these nondifferentiable constraints with a system of nonlinear equations to get the following nonlinear program:

$$\min_{x,\omega} \quad \omega \tag{17}$$

subject to

$$F(x,Y,\omega,\theta) = 0, \tag{18}$$

where $F(x,Y,\omega,\theta)$ is a block diagonal symmetric $n \times n$ matrix

$$F(x,Y,\omega,\theta) = F^1(x,Y,\omega,\theta) \oplus \cdots \oplus F^l(x,Y,\omega,\theta).$$

The matrices $F^j(x,Y,\omega,\theta)$ are defined as follows. The vector θ is given by

$$\theta = (\theta_{t_1+1}^1, \ldots, \theta_{n_1}^1, \ldots, \theta_{t_l+1}^l, \ldots, \theta_{n_l}^l).$$

We write

$$\Theta^j = \mathrm{Diag}(\theta_{t_j+1}^j, \ldots, \theta_{n_j}^j).$$

Let

$$Y = Y^1 \oplus \cdots \oplus Y^l,$$

where Y^j is a skew symmetric $n_j \times n_j$ matrix, and let

$$D = D^1 \oplus \cdots \oplus D^l$$

with $D^j = \omega I \oplus \Theta^j$, where I is the $t_j \times t_j$ identity matrix. Then

$$F^j(x,Y,\omega,\theta) = D^j - e^{-Y^j}\hat{Q}^{jT}(G^j)^{-1}(x)A^j(x)(G^j)^{-T}(x)\hat{Q}^j e^{Y^j},$$

where $G^j(x)$ is the Choleski factor of $B^j(x)$ and $\hat{Q}^j$ is the matrix of eigenvectors for the pencil $(A^j(\hat{x}), B^j(\hat{x}))$. Observe that the definition of F depends on $\hat{x}$ through $\hat{Q}$.

The matrix exponential e^{Y^j} is orthogonal since Y^j is skew symmetric, and it follows that (x, Y, ω, θ) solves (18) if and only if the pencil $(A^j(x), B^j(x))$ has eigenvalues

$$(\underbrace{\omega, \ldots, \omega}_{t_j}, \theta^j_{t_j+1}, \ldots, \theta^j_{n_j}).$$

In the case $t_j = 0$, the latter condition is an empty one, indicating that the block F^j can be omitted from F in this case.

The idea of replacing the nondifferentiable constraints (16) by an equation of the form (18) based on a matrix exponential formulation goes back to [7]. Indeed, consider the case where the number of variables and constraints in the quadratic program (11), (12) are the same, i.e., K is square, or equivalently, the number of constraints, as given by (13), equals $m+1$. Then, provided K is nonsingular, the solution of the quadratic program is completely defined by the constraint (12). In the case that $B(x) = I$, this essentially reduces to the step defined by Modified Method I in [7], the only difference being that in [7], the multiple eigenvalue $\lambda_{\max}$ is prescribed. The method in [7] can be viewed as a variation on Newton's method for solving (18), but the analysis is somewhat complicated because (a) the definition of F, which depends on $\hat{Q}$, changes at every step of the iteration, and (b) it is necessary to remove the leading t_j by t_j block of the variables Y^j from the formulation of F to obtain a well-posed iteration, leading to a sequence of nonlinear equations that are solvable only in the limit.

The matrix W in the quadratic objective (11) is derived from the Hessian of the Lagrangian function associated with the nonlinear program (17), (18), evaluated at $x = \hat{x}$, $\omega = \hat{\lambda}_{\max}$, $Y^j = 0$, and $\Theta^j = \hat{\Lambda}^j_{\text{rest}}$. A detailed discussion in the case $B(x) = I$ is given in [16]. Specifically, W is an $m \times m$ symmetric matrix whose (p, q)-entry is given by

$$w_{pq} = \sum_{j=1}^{l} \langle U^j, H^j_{pq} \rangle, \tag{19}$$

where H^j_{pq} is a symmetric $t_j \times t_j$ matrix computed as follows. It is empty if $t_j = 0$. Otherwise let $\hat{G}^j = G^j(\hat{x})$ be the Choleski factor of $B^j(\hat{x})$, let $\hat{L}^j_p$ denote the matrix defined by (2) for the block $B^j(\hat{x})$ and similarly $\hat{L}^j_{pq}$ for the matrix defined by (4). Finally write $\hat{Z}^j_p = Z^j_p(\hat{x})$. Then

$$H^j_{pq} = H^j_{pq,1} + H^j_{pq,2} + H^j_{pq,3} + H^j_{pq,4},$$

where:

1. $H^j_{pq,1} = M_1 + M_1^T$ with

$$M_1 = -\hat{Q}^{j\,T}_{\max} \hat{L}^j_{pq} (\hat{G}^j)^T \hat{Q}^j_{\max} \hat{\Lambda}^j_{\max};$$

2. $H^j_{pq,2} = M_2 + M_2^T$ with

$$M_2 = -\hat{Q}^{j\,T}_{\max} \hat{L}^j_p (\hat{G}^j)^{-1} \hat{A}^j (\hat{G}^j)^{-T} (\hat{L}^j_q)^T \hat{Q}^j_{\max};$$

3. $H^j_{pq,3} = M_3 + M_3^T$ with

$$M_3 = -\hat{Q}^{j\,T}_{\max} \hat{L}^j_p (\hat{G}^j)^{-1} \hat{Z}^j_q \hat{Q}^j_{\max} - \hat{Q}^{j\,T}_{\max} \hat{L}^j_q (\hat{G}^j)^{-1} \hat{Z}^j_p \hat{Q}^j_{\max};$$

4. $H^j_{pq,4} = M_4 + M_4^T$ with $M_4 = 0$ if $t_j = n_j$, and otherwise

$$M_4 = \hat{Q}^j_{\max}{}^T \hat{Z}^j_p \hat{Q}^j_{\text{rest}} (S^j)^{-1} \hat{Q}^j_{\text{rest}}{}^T \hat{Z}^j_q \hat{Q}^j_{\max};$$

where

$$S^j = \text{Diag}(\hat{\lambda}_{\max} - \hat{\lambda}^j_{t_j+1}, \ldots, \hat{\lambda}_{\max} - \hat{\lambda}^j_{n_j}).$$

Observe that the computation of M_4 seems to require explicit knowledge of all the eigenvalues and eigenvectors of $(\hat{A}^j, \hat{B}^j)$ rather than just the first t_j. However, a clever observation of Xianjian Ye [17] allows us to compute M_4 using only $\hat{\Lambda}^j_{\max}$ and $\hat{Q}^j_{\max}$. Let

$$T^j = \hat{\lambda}_{\max} \hat{B}^j - \hat{A}^j + \hat{B}^j \hat{Q}^j_{\max} \hat{Q}^j_{\max}{}^T \hat{B}^j.$$

Then the term $\hat{Q}^j_{\text{rest}} (S^j)^{-1} \hat{Q}^j_{\text{rest}}{}^T$ in M_4 is given by

$$\hat{Q}^j_{\text{rest}} (S^j)^{-1} \hat{Q}^j_{\text{rest}}{}^T = (T^j)^{-1} - \hat{Q}^j_{\max} \hat{Q}^j_{\max}{}^T.$$

Hence all references to $\hat{Q}^j_{\text{rest}}$ and $\hat{\Lambda}^j_{\text{rest}}$ can be eliminated from the equations for H^j_{pq}.

In the case $B(x) = I$ there is only one term contributing to the Hessian, namely, $H_{pq,4}$ with $\hat{Z}_p$ and $\hat{Z}_q$ replaced by $\hat{A}_p$ and $\hat{A}_q$, respectively, [13], [16]. All other terms vanish. When $B(x)$ is not constant, even though the second derivatives of $A(x)$ and $B(x)$ are zero since $A(x)$, $B(x)$ are affine, the second derivatives of the Choleski factor $G(x)$ are not. Indeed, they are the matrices $L_{ij}(x)$. This explains the presence of the terms $H_{pq,1}$, $H_{pq,2}$, and $H_{pq,3}$.

We now summarize the local algorithm.

Step 0. Initialize $\hat{x}$.

Step 1. Compute $A(\hat{x})$, $B(\hat{x})$, the Choleski factor $G(\hat{x})$, the multiplicity estimate t, the eigenvalues $\Lambda_{\max}(\hat{x})$, and the eigenvectors $Q_{\max}(\hat{x})$.

Step 2. Compute the matrix K and the vector h from (18).

Step 3. Obtain an estimate $U = U^1 \oplus \cdots \oplus U^l$ of the dual matrix. This is done by computing the least square solution v to the equation

$$K^T v = e_1,$$

where $e_1 = (1, 0, \ldots, 0) \in \Re^{m+1}$. Note that the estimate from a previous iterate is useless because the basis of eigenvectors may have rotated an arbitrary amount. The vector v has dimension

$$\sum_{j=1}^{l} \frac{t_j(t_j+1)}{2}$$

and can be assembled into the matrix U. Compute the eigenvalues of U. If U has a negative eigenvalue, split the maximum eigenvalue as explained in Theorem 3.3 and go to Step 1.

Step 4. Use the dual matrix estimate U to define the matrix W using (19) and solve the equality-constrained quadratic program (11), (12) to obtain a step $\hat{d}$. If $\|\hat{d}\| < 1$, a second order correction is computed to avoid the Maratos effect [6]. This is unnecessary in most cases but the cost is negligible.

Step 5. If $\|\hat{d}\|$ is less than a certain convergence tolerance, stop; otherwise, set $\hat{x} = \hat{x} + \hat{d}$, and go to Step 1.

The iteration just described is locally quadratically convergent to a minimizer of $\lambda_{\max}(x)$. This property is demonstrated by the numerical results in §6. The proof of this assertion, given the appropriate nonsingularity condition, requires extension of the results in [16] and is beyond the scope of this paper.

Observe that no line search is performed. If the new iterate fails to produce a reduction of $\lambda_{\max}$, the step $\hat{d}$ is rejected altogether, and a new step is computed using the method of centers. This is discussed further in the next section.

We conclude this section with a brief discussion of the case that the pencil $(A(x), B(x))$ depends nonlinearly on x. The linear approximations used in the constraints of the quadratic program remain valid, but the matrix W is not. The formula for W is easily generalized to apply to the case where A is nonlinear and B is affine, by including a second derivative term for A [16]. However, the formula for the case that B is nonlinear is far more complicated. Rapidly convergent methods could instead be constructed using quasi-Newton or limited memory quasi-Newton techniques [10] to approximate W. Indeed, because of the complicated form of W and the fact that it is a dense matrix, these may be preferable even for the affine case for moderately large sized problems.

5. The global algorithm. The global algorithm is a two-stage process. In the first stage, a sequence of iterates x^ν is computed using the method of centers (Boyd and El Ghaoui [2]) until the norm of the step $x^{\nu+1} - x^\nu$ is reduced below a certain threshold. In the second stage, we proceed as follows. First, we compute a step d^ν by solving the quadratic problem described in the previous section. If the point $x^\nu + d^\nu$ is feasible, i.e., $B(x^\nu + d^\nu) > 0$, and if d^ν is such that $\lambda_{\max}(x^\nu + d^\nu) < \lambda_{\max}(x^\nu)$, then we set $x^{\nu+1} = x^\nu + d^\nu$. Otherwise, $x^{\nu+1}$ is computed via the method of centers.

The algorithm of Boyd and El Ghaoui is thoroughly discussed in [2], so we sketch only the basic step. Consider the matrix inequality $C(x) > 0$ where $C(x)$ is the block diagonal matrix

$$C(x) = (\rho B(x) - A(x)) \oplus (B(x) - \mu I) \oplus (x_{\sup} I - \mathrm{Diag}(x)) \oplus (x_{\sup} I + \mathrm{Diag}(x)).$$

Here μ is a small constant used to ensure that $B(x)$ remains in the interior of the positive definite cone, while $x_{\sup}$ is a large constant used to ensure that x remains bounded. The quantity ρ is an upper bound on the value of the maximum eigenvalue of the pencil $(A(x), B(x))$. For a fixed ρ let $x^*(\rho)$ denote the analytic center of the inequality $C(x) > 0$, i.e., $x^*(\rho) = \operatorname{argmax} \det C(x)$ [2]. Now for ρ^ν and x^ν with $C(x^\nu) > 0$ let

$$x^{\nu+1} = x^*(\rho^\nu)$$

and

$$\rho^{\nu+1} = (1 - \sigma)\lambda_{\max}(x^\nu) + \sigma\rho^\nu,$$

where $0 < \sigma < 1$ is a parameter kept fixed throughout the algorithm. The analytic center is computed via Newton's method applied to the logarithmic barrier function $\log\det C^{-1}(x)$, using an exact line search for the computation of the step length.

The algorithm [2] applies only to the case that $(A(x), B(x)$ is affine. It should be possible to generalize some of these ideas to the nonlinear case, but as yet this has not been investigated.

TABLE 1
Hybrid.

ν	$\lambda_{\max}$	$\|d\|$	Mflops	N
1	2.56696348918400	2.937e-01	1.20	7
2	1.65996563727620	1.597e-01	1.54	9
3	1.08305393696925	1.002e-01	1.54	9
4	0.75949101184420	8.061e-02	1.54	9
5	0.66424777849618	6.591e-02	1.37	8
6	0.66109151988020	3.149e-02	1.18	7
7	0.66063603170038	2.594e-03	1.19	7
8	0.66055966458506	1.785e-03	0.14	0
9	0.66055960982024	1.358e-06	0.14	0
10	0.66055960981957	5.684e-11	0.14	0

TABLE 2
Method of centers.

ν	$\lambda_{\max}$	$\|d\|$	Mflops	N
1	2.56696348918400	2.937e-01	1.20	7
⋮	⋮	⋮	⋮	⋮
7	0.66063603170038	2.594e-03	1.19	7
8	0.66057058873535	4.167e-04	1.19	7
9	0.66056118766552	5.915e-05	1.19	7
10	0.66055983654089	1.960e-05	1.36	8

6. Numerical results. We now present some numerical results. The algorithms were implemented in MATLAB.

We first consider the example presented in [2]. The matrices $A(x)$ and $B(x)$ are block diagonal with four blocks of dimension 4. There are nine variables. We set $\mu = 0.0001$, $x_{\sup} = 50$, $\sigma = 0.001$. The threshold value κ, determining when to attempt to switch to the local algorithm, is set to 0.01. The vector x is initialized to $(1,1,1,0,0,0,0,0,0)$. The results of the hybrid algorithm are given in Table 1 while Table 2 contains the output from the method of centers alone. In both cases the results use our implementation of the method of centers, as described in [2]. The number N denotes the number of Newton steps required by the inner iteration that computes the analytic center. This is zero once the hybrid algorithm has switched to the local scheme. The expression $\|d\|$ refers to the norm of $x^{\nu+1} - x^{\nu}$. The hybrid algorithm terminates when $\|d\| < 10^{-10}$. In the case of Table 2, the stopping criterion is $\bar{\rho} - \lambda_{\max}(\bar{x}) < 10^{-10}$. The Mflops column displays the number of floating point operations required in millions. The large number of operations reflects the fact that the inner iteration required by the method of centers is being performed accurately; this number could undoubtedly be reduced. The significant point, however, is the following: the global algorithm finds the neighborhood of the solution very reliably, while the use of the local algorithm, once in the neighborhood of the solution, rapidly locates the solution to full precision. Note, specifically, the quadratic convergence of $\|d\|$ to zero once the local scheme is in effect. The eigenvalue $\lambda_{\max}$ has block multiplicities 1, 0, 0, 1 at the computed solution x^*. The hybrid algorithm reduces the gap between $\lambda_1(x^*)$ and $\lambda_2(x^*)$ to 1.4×10^{-15}, while the method of centers reduces it only to 5.4×10^{-7}. The dual matrix U has block dimensions $(1,0,0,1)$, corresponding to the multiplicity of $\lambda_{\max}$, and is computed by the hybrid algorithm in Step 3 of the

local scheme. Its final value is found to be Diag(.583, .417) (to three digits), verifying that the optimality condition (8) is satisfied. Optimality conditions (7) and (9) hold to machine precision.

TABLE 3
Hybrid.

ν	$\lambda_{\max}$	$\|d\|$	Mflops	N
1	0.7989467570479351	2.889803	41.0	9
2	0.5279030947496347	1.161417	41.0	9
3	0.4096247700327728	5.185782e-01	41.0	9
4	0.3564700009737637	2.556177e-01	41.0	9
5	0.3313368545063169	1.409038e-01	40.9	9
6	0.3190742531948093	8.384906e-02	40.9	9
7	0.3130620662624993	5.438315e-02	40.9	9
8	0.3101212297584327	3.939520e-02	40.9	9
9	0.3086835410941962	2.939005e-02	40.9	9
10	0.3079797136097306	2.094047e-02	40.9	9
11	0.3076340074134891	1.419033e-02	40.9	9
12	0.3074633984606852	9.392075e-03	40.9	9
13	0.3073787506822123	6.222999e-03	46.3	9
14	0.3073365132846426	4.171214e-03	46.3	9
15	0.3072932213520559	8.185445e-03	5.7	0
16	0.3072931684710006	3.461869e-05	5.7	0
17	0.3072931684689409	1.382263e-09	5.7	0
18	0.3072931684689404	2.081514e-15	5.7	0

TABLE 4
Method of centers.

ν	$\lambda_{\max}$	$\|d\|$	Mflops	N
1	0.7989467570479351	2.889803	41.0	9
⋮	⋮	⋮	⋮	⋮
13	0.3073787506822123	6.222999e-03	40.9	9
14	0.3073365132846426	4.171214e-03	40.9	9
15	0.3073153048002088	2.824038e-03	40.9	9
16	0.3073045769895835	1.916467e-03	40.9	9
17	0.3072991033588865	1.290257e-03	40.9	9
18	0.3072962829954534	8.516237e-04	40.9	9
19	0.3072948149058472	5.446555e-04	40.9	9
20	0.3072940435712944	3.348214e-04	40.9	9
21	0.3072936352938653	1.976833e-04	40.9	9
22	0.3072934180681651	1.127953e-04	40.9	9
23	0.3072933021081565	6.278650e-05	40.9	9
24	0.3072932400776711	3.439867e-05	40.9	9

The second example provides a better test of the new algorithm, since $\lambda_{\max}(x^*)$ has block multiplicities greater than one. There are fifteen variables and the matrices $A(x)$ and $B(x)$ are block diagonal with three blocks, each of dimension 10. The matrix B_0 was set to the identity matrix and the matrices $A_0, \ldots, A_{15}$ and $B_1, \ldots, B_{15}$ were generated using the MATLAB random number generator, and symmetrizing. The matrices $B_1, \ldots, B_{15}$ were then scaled by a factor 0.05 to ensure the existence of a reasonable-sized domain with $B(x) > 0$. All parameters have the same values as in the previous example. The vector x is initialized with a random vector. The results are given in Tables 3 and 4. The hybrid algorithm attempted to switch to the local scheme at the thirteenth step since $\|d^{12}\| < \kappa$. The step was rejected, however, because it

failed to achieve a reduction of $\lambda_{\max}$. The operation count for $\nu = 13$ is the sum of the cost of one local step and one iteration of the method of centers. The same phenomenon occurred at the fourteenth step. The algorithm switched permanently to the local scheme at the fifteenth step and quadratic convergence was established. The last line indicates the limits of double precision computation. The data for this example is available from the authors.

The multiplicity of $\lambda_{\max}$ at the computed solution x^* is 6, with block multiplicities 1, 2, and 3. The hybrid algorithm reduces the gap between λ_1 and λ_6 to 3.3×10^{-16}, while the method of centers reduces the gap to 1.4×10^{-5}. The final dual matrix U has block dimensions $(1, 2, 3)$, satisfies the optimality conditions (7) and (9) to machine precision, and has eigenvalues 0.195 (first block), 0.074, 0.126 (second block), and 0.002, 0.244, and 0.359 (third block), demonstrating that (8) is also satisfied.

7. Conclusion. We have presented an algorithm for the optimization of the maximum eigenvalue of a symmetric definite pencil depending affinely on a vector of parameters. The algorithm combines a scheme based on the method of centers developed by Boyd and El Ghaoui [2] and a new local scheme exhibiting quadratically convergent behavior. The local scheme is an extension of the methods introduced in [13], [14], and [16] to the case of matrix pencils.

REFERENCES

[1] F. Alizadeh, *Optimization over positive semi-definite cone: Interior-point methods and combinatorial applications*, in Advances in Optimization and Parallel Computing, P. Pardalos, ed., North-Holland, Amsterdam, 1992.

[2] S. Boyd and L. El Ghaoui, *Method of centers for minimizing generalized eigenvalues*, Linear Algebra Appl.,188 (1993), pp. 63–112.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 1994.

[4] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983. Reprinted by Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[5] M. K. H. Fan and B. Nekooie, *On minimizing the largest eigenvalue of a symmetric matrix*, Proc. 31st IEEE Conference on Decision and Control, Tuscon, AZ, 1992.

[6] R. Fletcher, *Second order corrections for non-differentiable optimization*, in Numerical Analysis, Dundee 1981, G. A. Watson, ed., Lecture Notes in Math. 912, Springer-Verlag, New York, 1982, pp. 85–114.

[7] 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.

[8] J.-P. Haeberly, *On shape optimizing the ratio of the first two eigenvalues of the Laplacian*, Tech. Report 586, Computer Science Dept., New York University, New York, 1991.

[9] F. Jarre, *An interior point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control Optim., 31 (1993), pp. 1360–1377.

[10] D. C. Liu and J. Nocedal, *On the limited memory method for large scale optimization*, Math. Programming B, 45, pp. 503–528.

[11] Y. Nesterov and A. Nemirovsky, *An interior point method for generalized linear-fractional programming*, Tech. report, USSR Acad. Sci. Centr. Econ. and Math. Inst., 32 Krasikova St., Moscow 117418, 1991.

[12] ———, *Interior Point Polynomial Algorithms in Convex Programming*, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[13] M. L. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.

[14] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[15] M. L. Overton and R. S. Womersley, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Math. Programming, 62 (1993), pp. 321–357.

[16] ———, *Second derivatives for optimizing eigenvalues of symmetric matrices*, SIAM J. Matrix Anal. Appl., to appear.

[17] X. YE, Private communication, New York University, July 1991.
[18] B. GOLLAN, *Eigenvalue perturbations and nonlinear parametric optimization*, Math. Programming Stud., 30 (1987), pp. 67–81.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1157–1166, October 1994

008

ROW SUMS AND INVERSE ROW SUMS FOR NONNEGATIVE MATRICES *

SHMUEL FRIEDLAND[†1], ROHAN HEMASINHA[‡], HANS SCHNEIDER[§1], JEFFREY STUART[¶1], AND JAMES WEAVER[‡1]

Abstract. For a nonnegative, irreducible matrix A, the grading of the row sums vector and the grading of the Perron vector are used to predict the grading of the row sums vector of $(I - A)^{-1}$. This has applications to Markov chains.

Key words. row sums, inverse row sums, Markov chain, nonnegative matrix

AMS subject classifications. 15A48, 15A42

0. Motivation. Let T be a row stochastic matrix. It is well known that the matrix T is the transition matrix associated with an absorbing Markov chain if and only if T is permutation similar to a matrix of the form

$$T = \begin{bmatrix} I & 0 \\ B & A \end{bmatrix},$$

where A is a square matrix with $\rho(A) < 1$ [BP, Thm. 8.3.21]. Furthermore, if F is the set of indices corresponding to the nonabsorbing, i.e., transient, states then the expected number of steps until absorption when starting in the nonabsorbing state i is given by

$$\sum_{j \in F} [(I - A)^{-1}]_{ij}$$

[BP, Thm. 8.4.27]. This leads to the natural question of what can be said about the row sums of the matrix $(I - A)^{-1}$ given some knowledge about the matrix A. In particular, what can we predict about the maximum and minimum row sums of $(I - A)^{-1}$ given the row sums of A and the Perron vector for A?

1. Notation. For an $n \times n$ matrix A, let $\rho = \rho(A)$ denote the spectral radius of A. The real matrix A will be called nonnegative, denoted $A \geq 0$, if each entry of A is nonnegative. If A is nonnegative and irreducible, let $X = X_A$ denote the Perron vector of euclidean norm one for A; that is, X is the unique strictly positive eigenvector of norm one corresponding to the eigenvalue $\rho(A)$. Unless otherwise specified, the matrix A will always be an $n \times n$ nonnegative matrix.

---

* Received by the editors July 27, 1992; accepted for publication December 8, 1993.

† Department of Mathematics, University of Illinois at Chicago, Chicago, Illinois 60680

‡ Department of Mathematics and Statistics, University of West Florida, Pensacola, Florida 32514

§ Department of Mathematics, University of Wisconsin, Madison, Wisconsin 53706. The work of this author was partially supported by National Science Foundation grants DMS 8901445, DMS 9123318, and ECS 8718971

¶ Department of Mathematics, University of Southern Mississippi, Hattiesburg, Mississippi 39406

[1] The research for this paper was performed while these authors were visiting the Institute for Mathematics and Its Applications, University of Minnesota, Minneapolis, Minnesota 55455

Let $v = (v_1, v_2, \dots, v_n)^t \in \mathbb{R}^n$. There exists a permutation σ such that $v_{\sigma(1)} \geq v_{\sigma(2)} \geq \cdots \geq v_{\sigma(n)}$. The integer vector $(\sigma^{-1}(1), \sigma^{-1}(2), \dots, \sigma^{-1}(n))^t$ is called a grading of v. If the entries of v are pairwise distinct, then v has a unique grading and v is called a strictly graded vector. A set of vectors is said to share a common grading if the intersection of their sets of gradings is nonempty.

For $1 \leq i \leq n$, let e_i denote the ith standard basis vector for $\mathbb{R}^n$. Let $u = u_n$ denote the vector of ones. That is,

$$u_n = \sum_{i=1}^{n} e_i.$$

Let $D = D_n$ denote the cone in $\mathbb{R}^n$ generated by the vectors $e_1, e_1 + e_2, e_1 + e_2 + e_3, \dots, u_n$; that is, $D = \{v \in \mathbb{R}^n\colon\ v_1 \geq v_2 \geq \cdots \geq v_n \geq 0\}$. Let $\Pi(D)$ denote the class of D-preserving matrices: $\Pi(D) = \{A \in \mathcal{M}_n(\mathbb{R})\colon\ A(D) \subseteq D\}$.

Note that a nonnegative vector $v \in \mathbb{R}^n$ has its entries in decreasing order if and only if $v \in D$, and that v has its entries in strictly decreasing order if and only if $v \in \operatorname{int}(D)$, where $\operatorname{int}(D)$ denotes the interior of D. Also note that if $A \in \Pi(D)$, then $A^k \in \Pi(D)$ for all positive integers k. Finally note that the row sums of the matrix A are precisely the entries of the vector Au.

LEMMA 1.1. *If A is a nonnegative, primitive matrix with $\rho(A) < 1$, such that $Au, (I - A)^{-1}u$, and X_A share a common (strict) grading, then there exists a permutation matrix P such that $PAP^tu, (I - PAP^t)^{-1}u$, and PX_A are all in $(\operatorname{int}(D))D$. Furthermore, $PX_A = X_{PAP^t}$.*

Proof. Let $v = Au$. Let the permutation matrix P correspond to the common permutation σ in the definition of grading. Then $PAP^tu = PAu \in D$. Since $\rho < 1, (I-A)^{-1}$ exists. Since $P(I-A)^{-1}P^t = (I-PAP^t)^{-1}$, and since Au and $(I-A)^{-1}u$ share the common grading $\sigma, (I-PAP^t)^{-1}u = (I-PAP^t)^{-1}P^tu = P(I-A)^{-1}u \in D$. Finally, X is an eigenvector for ρ for A if and only if PX is an eigenvector for ρ for PAP^t. Since multiplication by P is norm preserving and since PAP^t is nonnegative and primitive, $X_{PAP^t} = PX_A$. Note that σ is a common grading for Au and X_A, so $PX_A \in D$. □

One immediate consequence of this lemma is that we can always assume that a graded vector has its entries in decreasing order. Thus questions about graded vectors are transformed to questions about vector membership in the cone D.

Finally, recall the Neumann expansion for the inverse of the matrix $I - A$.

THEOREM 1.2 [O]. *Let A be an $n \times n$ real matrix with $\rho(A) < 1$. Then $(I - A)^{-1}$ exists, and*

$$(I - A)^{-1} = I_n + \sum_{k=1}^{\infty} A^k.$$

2. Empirical evidence. If A is nonnegative and primitive, then by the power method, $A^ku \approx c_kX_A$ for large k. Furthermore, if $\rho(A) < 1$, then $c_k \to 0$ as $k \to \infty$. Also, $\rho(A) < 1$ implies

$$(I - A)^{-1} u = u + \sum_{k=1}^{\infty} A^k u.$$

This suggests that the grading for $(I - A)^{-1}u$ should be linked to the grading for X_A, and that the early terms in the summation should be the most important. Since

$(I - A)^{-1}u$ and $(I - A)^{-1}u - u$ have the same grading, and since

$$(I - A)^{-1} u - u = Au + \sum_{k=2}^{\infty} A^k u,$$

the importance of the grading of Au is immediately apparent. When Au and X share a common grading, it remains to be seen how much of an effect the remaining low order summands have on the grading of $(I - A)^{-1}u$.

Motivated by numerical experiments conducted using MATLAB on an APOLLO workstation, we were led to several conjectures. The first was that if Au, X, and $(I - A)^{-1}u$ all share a common grading, then that grading is shared by A^k for all positive k. The second and more interesting conjecture was that if Au and X share a common grading, then $(I - A)^{-1}u$ also shares that grading. Unfortunately, neither conjecture holds.

If

$$A = \begin{bmatrix} 0.0783 & 0.2999 & 0.2421 & 0.0089 \\ 0.0305 & 0.0003 & 0.1814 & 0.2272 \\ 0.0013 & 0.1196 & 0.1426 & 0.1305 \\ 0.0008 & 0.0005 & 0.0009 & 0.0003 \end{bmatrix},$$

then

$$Au = \begin{bmatrix} 0.6292 \\ 0.4394 \\ 0.3940 \\ 0.0025 \end{bmatrix}, \quad X = \begin{bmatrix} 0.8954 \\ 0.3166 \\ 0.3131 \\ 0.0043 \end{bmatrix}, \quad \text{and} \quad (I - A)^{-1} u = \begin{bmatrix} 2.0101 \\ 1.5695 \\ 1.5411 \\ 1.0041 \end{bmatrix}.$$

Hence, Au, X, and $(I - A)^{-1}u$ are all in D. However,

$$A^2 u = \begin{bmatrix} 0.2765 \\ 0.0914 \\ 0.1099 \\ 0.0011 \end{bmatrix},$$

which is not in D since its second and third entries are not in decreasing order.

Discovering a counterexample to the second conjecture proved to be very difficult. The following matrix was one of only four counterexamples generated during a run of 25,000 randomly generated, rank four, strictly positive 4×4 matrices with spectral radius less than one. In fact, for 67% of the matrices generated in that run, all three vectors—Au, X, and $(I - A)^{-1}u$—shared a common grading.

$$A = \begin{bmatrix} 0.2042 & 0.2837 & 0.0196 & 0.2356 \\ 0.1522 & 0.2149 & 0.0320 & 0.2848 \\ 0.1060 & 0.2072 & 0.1289 & 0.2415 \\ 0.2750 & 0.1958 & 0.0805 & 0.0187 \end{bmatrix},$$

then

$$Au = \begin{bmatrix} 0.7431 \\ 0.6839 \\ 0.6836 \\ 0.5700 \end{bmatrix}, \quad X = \begin{bmatrix} 0.5513 \\ 0.4991 \\ 0.4987 \\ 0.4452 \end{bmatrix}, \quad \text{and} \quad (I - A)^{-1} u = \begin{bmatrix} 3.2356 \\ 3.0330 \\ 3.0338 \\ 2.7799 \end{bmatrix}.$$

Hence Au and X are in D, however, $(I - A)^{-1}u$ is not in D.

A run of 100,000 randomly generated, rank three, strictly positive 4×4 matrices with spectral radius less than one yielded only one counterexample. Furthermore, for this run, 91% of the matrices had all vectors sharing a common grading, and an additional 3% had the Perron vector and the inverse row sums vector (but not the row sums vector) sharing a common grading.

Extensive numerical experiments with matrices of sizes up to 50×50 lead to the following observations. First, for low rank matrices, the grading of X is a good predictor for the grading of $(I - A)^{-1}u$. Second, even when the vectors do not share a common grading, they share a roughly blocked common grading in the sense that the grading vectors differ within blocks corresponding to closely sized entries of the vectors. In particular, the set of indices for the smallest (largest) row sums of A correspond roughly to the set of indices of the smallest (largest) entries of the Perron vector and to the set of indices of the smallest (largest) row sums of $(I - A)^{-1}$.

3. An analytic approach. In this section, we present several different types of results including an examination of certain classes of matrices for which the grading of the Perron vector and the row sums vector do determine the grading for the inverse row sums vector.

PROPOSITION 3.1. *Let A be an $n \times n$ nonnegative, irreducible matrix with $\rho = \rho(A) < 1$. Suppose that $X = X_A \in D$. For $1 \leq i \leq n$,*

$$(1 - \rho)^{-1} \frac{x_i}{x_1} \leq [(I - A)^{-1} u]_i \leq (1 - \rho)^{-1} \frac{x_i}{x_n}.$$

Proof. Since $A \geq 0$, and $\rho < 1, I - A$ is an invertible M-matrix. Thus $(I - A)^{-1} \geq 0$ [BP, Thm. 6.2.3]. Since $X \in D$ and X is strictly positive, $x_1 \geq \cdots \geq x_n > 0$. Note that $(I - A)^{-1}X = (1 - \rho)^{-1}X$. Thus for $1 \leq i \leq n$,

$$\begin{aligned}(1 - \rho)^{-1} x_i &= \sum_j [(I - A)^{-1}]_{ij} x_j \\ &\leq \sum_j [(I - A)^{-1}]_{ij} x_1 = [(I - A)^{-1} u]_i x_1.\end{aligned}$$

Similarly, the other bound holds. □

THEOREM 3.2. *Let A be a nonnegative, irreducible matrix with $\rho(A) < 1$. If $A \in \Pi(D)$, then Au, X_A and $(I - A)^{-1}u$ are all in D.*

Proof. Since $u \in D$, and since $A^k \in \Pi(D)$ for all positive $k, A^k u \in D$ for all positive k. Since $\rho(A) < 1$, it follows from Theorem 1.2 that $(I - A)^{-1}u = u + \sum_{k=1}^{\infty} A^k u \in D$. Finally since A is nonnegative and irreducible, X_A exists, and by the Krein–Rutman Theorem [BP, Thm. 1.3.2], $A \in \Pi(D)$ implies $X_A \in D$. □

COROLLARY 3.3. *Let A be a nonnegative, irreducible $n \times n$ matrix with $\rho(A) < 1$. If $a_{ij} \geq a_{i+1,j}$ for $1 \leq i \leq n - 1$ and $1 \leq j \leq n$, then Au, X_A, and $(I - A)^{-1}u$ are all in D.*

Proof. Pick $z \in D$. Note that $z \geq 0$ and $A \geq 0$. Thus for $1 \leq i \leq n - 1$,

$$(Az)_i = \sum_j a_{ij} z_j \geq \sum_j a_{i+1,j} z_j = (Az)_{i+1} \geq 0.$$

Hence $Az \in D$. □

THEOREM 3.4. *Let A be a nonnegative, irreducible $n \times n$ matrix with $\rho = \rho(A) < 1$. Suppose that the minimum polynomial of A is $m_A(\lambda) = \lambda^k(\lambda - \rho)$. Then*

$(I - A)^{-1} = I + A + \cdots + A^{k-1} + (1-\rho)^{-1}A^k$. *Suppose that* $Au, \ldots, A^{k-1}u$ *are in* D. *If either of* X_A *and* $A^k u$ *is in* D, *then all three of* $X_A, A^k u$, *and* $(I-A)^{-1}u$ *are in* D. *Finally, if at least one of* $X_A, Au, \ldots, A^k u$ *is in* $\operatorname{int}(D)$, *then* $(I-A)^{-1}u$ *is in* $\operatorname{int}(D)$.

Proof. Let $X = X_A$. Then $A = \rho XY^t + N$ where Y^t is the strictly positive row eigenvector for ρ such that $Y^t X = 1$, and where N is the nilpotent matrix of index k satisfying $NX = 0$ and $Y^t N = 0^t$. For all nonnegative r, $A^{k+r} = \rho^{k+r} XY^t = \rho^r A^k$. Hence

$$\sum_{r=k}^{\infty} A^r = \sum_{r=0}^{\infty} \rho^r A^k = (1-\rho)^{-1} A^k.$$

Thus $(I-A)^{-1} = I + A + \cdots + (1-\rho)^{-1}A^k$. Since $A^k u = \rho^k (Y^t u)X$, X is in D if and only if $A^k u$ is in D. Since $u \in D$, it follows that $(I-A)^{-1}u \in D$ when $Au, \ldots, A^k u$ are in D. Furthermore, if one of the summands is in $\operatorname{int}(D)$, then it is clear that $(I-A)^{-1}u$ is in $\operatorname{int}(D)$. □

COROLLARY 3.5. *Let* A *be a nonnegative, irreducible* $n \times n$ *matrix with* $\rho = \rho(A) < 1$. *If* $\operatorname{rank}(A) = 1$ *and if* $Au \in D$, *then* X_A *and* $(I-A)^{-1}u$ *are in* D.

THEOREM 3.6. *Let* A *be a nonnegative, irreducible* $n \times n$ *matrix with* $\rho = \rho(A) < 1$. *Suppose that the minimum polynomial of* A *is either* $m_A(\lambda) = \lambda(\lambda - \rho)(\lambda - \lambda_1)$ *or else* $m_A(\lambda) = (\lambda - \rho)(\lambda - \lambda_1)$, *where* $\lambda_1 \neq 0$. *If* X_A *and* Au *are in* D, *then* $(I-A)^{-1}u$ *is in* D. *Furthermore, if* Au *is in* $\operatorname{int}(D)$, *then* $(I-A)^{-1}u$ *is in* $\operatorname{int}(D)$.

Proof. Let $X = X_A$. Then $A = \rho XY^t + \lambda_1 E$, where Y^t is the strictly positive row eigenvector for ρ satisfying $Y^t X = 1$, and where $E^2 = E$, $EX = 0$, and $Y^t E = 0^t$. For each positive k, $A^k = \rho^k XY^t + \lambda_1^k E$. Since $\rho < 1$, $(I-A)^{-1} = I + \rho(1-\rho)^{-1}XY^t + \lambda_1(1-\lambda_1)^{-1}E$. Then $(I-A)^{-1}u = u + \rho(1-\rho)^{-1}(Y^t u)X + \lambda_1(1-\lambda_1)^{-1}Eu$. Since $\lambda_1 < \rho < 1, 0 < (1-\lambda_1)^{-1} < (1-\rho)^{-1}$. Clearly, $Au \geq 0$. Now $Au \in D$ if and only if for $1 \leq i \leq n-1$,

$$\begin{aligned}
&(Au)_i \geq (Au)_{i+1} \\
&\quad \Leftrightarrow \rho\,(Y^t u)\,X_i + \lambda_1\,(Eu)_i \geq \rho\,(Y^t u)\,X_{i+1} + \lambda_1\,(Eu)_{i+1} \\
&\quad \Leftrightarrow \rho\,(Y^t u)\,[X_i - X_{i+1}] \geq \lambda_1\left[(Eu)_{i+1} - (Eu)_i\right] \\
&\quad \Leftrightarrow \rho\,(1-\rho)^{-1}\,(Y^t u)\,[X_i - X_{i+1}] \geq \lambda_1\,(1-\rho)^{-1}\left[(Eu)_{i+1} - (Eu)_i\right].
\end{aligned}$$

Since $X \in D$, the left-hand side of the last inequality is nonnegative; hence the inequality remains valid when $(1-\rho)^{-1}$ is replaced with $(1-\lambda_1)^{-1}$ on the right-hand side. Thus it holds that

$$\begin{aligned}
&(Au)_i \geq (Au)_{i+1} \\
&\quad \Rightarrow \rho\,(1-\rho)^{-1}\,(Y^t u)\,[X_i - X_{i+1}] \geq \lambda_1\,(1-\lambda_1)^{-1}\left[(Eu)_{i+1} - (Eu)_i\right] \\
&\quad \Leftrightarrow \rho\,(1-\rho)^{-1}\,(Y^t u)\,X_i + \lambda_1\,(1-\lambda_1)^{-1}\,(Eu)_i \\
&\qquad \geq \rho\,(1-\rho)^{-1}\,(Y^t u)\,X_{i+1} + \lambda_1\,(1-\lambda_1)^{-1}\,(Eu)_{i+1}.
\end{aligned}$$

That is, $Au \in D$ and $X \in D$ together imply $\rho(1-\rho)^{-1}(Y^t u)X + \lambda_1(1-\lambda_1)^{-1}Eu \in D$. That is, $(I-A)^{-1}u - u \in D$. Since $u \in D$, $(I-A)^{-1}u \in D$. Furthermore, if $Au \in \operatorname{int}(D)$, then for all i, each inequality in the argument above can be replaced with a strict inequality, hence $(I-A)^{-1}u \in \operatorname{int}(D)$. □

Remark. Let A be an $n \times n$, nonnegative, irreducible matrix with $\rho(A) < 1$. For $n \leq 3$, there are only two cases for A that are not covered in the results above: when

A is 3×3, nonsingular, and either A is not diagonalizable or A has three distinct eigenvalues. All cases for $n = 3$ are contained in Theorem 4.6.

Let A be a nonnegative, irreducible matrix with $\rho(A) < 1$. In view of the preceding results, several natural open questions arise. Suppose that the minimum polynomial for A has degree k. If each of $X_A, Au, A^2u, \ldots, A^{k-1}u$ are in D, does that imply that $(I - A)^{-1}u$ is in D? Does it imply that $A^r u$ is in D for all positive r? If not, what additional restrictions might be sufficient on A or on the minimum polynomial?

4. A second analytic approach.

THEOREM 4.1. *Let B be an $n \times n$ complex matrix with $\rho(B) = 1$. Then there exists a unique positive integer k with $k < n$, and there exist $n \times n$ complex matrices $B_1, \ldots, B_k$ with $B_k \neq 0$ such that* $\operatorname{adj}(I - xB) = I + xB_1 + x^2B_2 + \cdots + x^kB_k$. *Furthermore,*

(i) $k = n - 1$ *if and only if* $\operatorname{rank}(B) \geq n - 1$;

(ii) *if $k < n - 1$, then $k = m + t - 1$, where m is the number of nonzero eigenvalues of B (counting multiplicities), and where t is the size of the largest Jordan block corresponding to the eigenvalue 0 for B.*

Proof. Since each entry of $\operatorname{adj}(I - xB)$ is either zero or (± 1) times an $(n - 1) \times (n - 1)$ minor of $(I - xB)$, it follows that $k \leq n - 1$. Thus $\operatorname{adj}(I - xB) = B_0 + xB_1 + x^2B_2 + \cdots + x^{n-1}B_{n-1}$. Setting $x = 0, B_0 = \operatorname{adj}(I - 0B) = I$. Note that the coefficient matrix for x^{n-1} is generated only by terms from $-xB$. That is, $B_{n-1} = \operatorname{adj}(-B)$. Note that $\operatorname{adj}(-B) = 0$ if and only if every $(n - 1) \times (n - 1)$ minor of B is zero. That is, if and only if $\operatorname{rank}(B) < n - 1$. Thus (i) is proven.

If S is an invertible matrix, then $S = [\det(S)]\operatorname{adj}(S^{-1})$. Thus $S \operatorname{adj}(I - xB)S^{-1} = \operatorname{adj}(S^{-1})\operatorname{adj}(I - xB)\operatorname{adj}(S) = \operatorname{adj}(S(I - xB)S^{-1}) = \operatorname{adj}(I - xSBS^{-1})$. Thus

$$\operatorname{adj}(I - xSBS^{-1}) = I + xSB_1S^{-1} + x^2SB_2S^{-1} + \cdots + x^{n-1}SB_{n-1}S^{-1}.$$

Choose S so that SBS^{-1} is the Jordan canonical form of B. That is, $SBS^{-1} = J_1 \oplus \cdots \oplus J_r \oplus J_{r+1} \oplus \cdots \oplus J_s$, where the J_α for $1 \leq \alpha \leq r$ are the Jordan blocks corresponding to nonzero eigenvalues, and the J_α for $r < \alpha \leq s$ are the Jordan blocks corresponding to the eigenvalue zero. Then

$$\operatorname{adj}(I - xSBS^{-1}) = \bigoplus_{\alpha=1}^{s} \left[\operatorname{adj}(I - xJ_\alpha) \prod_{\beta \neq \alpha} \det(I - xJ_\beta) \right].$$

Consider the adjoint for a single Jordan block: $J = \lambda I_h + N_h$, where N_h is the $h \times h$ matrix whose only nonzero entries are ones down the superdiagonal. $\operatorname{Adj}(I - xJ)$ has diagonal entries $(1 - x\lambda)^{h-1}$, and the nonzero off-diagonal terms are of the form $(-x)^j(1 - x\lambda)^{h-j-1}$ for $1 \leq j \leq h - 1$. When $\lambda \neq 0$, the maximum degree of x in $\operatorname{adj}(I - xJ)$ is $h - 1$. When $\lambda = 0, (-x)^{h-1}$ is the only type of nonzero term. Thus $\operatorname{adj}(I - xJ)$ is always of degree $h - 1$ in x. Note that $\det(I - xJ) = (1 - x\lambda)^h$. When $\lambda \neq 0, \det(I - xJ)$ is of degree h in x. When $\lambda = 0, \det(I - xJ)$ is of degree zero in x. Consequently, the maximum degree of x in $\Pi_{\alpha=1}^{s} \det(I - xJ_\alpha)$ is precisely the sum of the sizes of $J_1, J_2, \ldots, J_r$. That is, $\Pi_{\alpha=1}^{s} \det(I - xJ_\alpha)$ is of degree m in x. Hence for $1 \leq \alpha \leq r, \operatorname{adj}(I - xJ_\alpha)\ \Pi_{\beta \neq \alpha} \det(I - xJ_\beta)$ is of degree $(h_\alpha - 1) + (m - h_\alpha)$, where h_α is the size of J_α. For $r < \alpha \leq s, \operatorname{adj}(I - xJ_\alpha)\ \Pi_{\beta \neq \alpha} \det(I - xJ_\beta)$ is of degree $(h_\alpha - 1) + (m - 0)$, where h_α is the size of J_α. Thus the maximum degree of x in $\operatorname{adj}(I - xSBS^{-1})$ and, hence, in $\operatorname{adj}(I - xB)$ is $m + t - 1$, where t is the size of the largest Jordan block for the eigenvalue zero. □

COROLLARY 4.2. *Let A be a nonnegative, irreducible matrix. The following are equivalent*:

(i) $\operatorname{adj}(I - xA) = I + xA_1 + x^2A_2$,

(ii) *at least one of the following holds*;

(a) $n \leq 3$,

(b) $n > 3$ *and* $\operatorname{rank}(A) \leq 2$,

(c) $n > 3$ *and* $\operatorname{rank}(A^2) = 1$.

Note that conditions (b) *and* (c) *imply that the size of the largest possible Jordan block for the eigenvalue zero is two.*

Proof (i) $\rightarrow$ (ii). If $A_2 \neq 0$ then either $2 = n - 1$, hence $n = 3$, or else $2 < n - 1$ and $2 = m + t - 1$. That is, $n > 3$ and $m + t = 3$. Since A is irreducible, $m \geq 1$. Since $t = 0$ implies $m = n, t \geq 1$. Thus either $m = 2$ and $t = 1$, implying $\operatorname{rank}(A) = 2$ or $m = 1$ and $t = 2$, implying $\operatorname{rank}(A^2) = 1$.

If $A_2 = 0$, but $A_1 \neq 0$, then either $1 = n - 1$, hence $n = 2$, or $1 < n - 1$ and $1 = m + t - 1$. In the latter case, $n > 2$ and $m + t = 2$, implying $m = t = 1$. That is, $\operatorname{rank}(A) = 1$.

If $A_2 = A_1 = 0$, then $A = 0$, which contradicts the irreducibility of A. □

Proof (ii) $\rightarrow$ (i) If $n \leq 3$, (i) is immediate. If $n > 3$ and $\operatorname{rank}(A) \leq 2$, then since $m + (t - 1) \leq \operatorname{rank}(A)$ always, $k = m + t - 1 \leq 2$. Now apply Theorem 4.1. Finally, if $n > 3$ and $\operatorname{rank}(A^2) = 1$, then $\rho(A)$ is the unique nonzero eigenvalue and $m = 1$. Clearly, $\operatorname{rank}(A^2) \leq 1$ implies $t \leq 2$. Again, $k = m + t - 1 \leq 2$. Apply Theorem 4.1. □

LEMMA 4.3. *Let B be a nonnegative, irreducible matrix with $\rho(B) = 1$. If $Bu \in \operatorname{int}(D)$, then there exists a maximal $\omega = \omega(B)$ such that $0 < \omega \leq 1$, and such that $(I - xB)^{-1}u \in \operatorname{int}(D)$ for $0 < x < \omega$.*

Proof. Since $\rho(B) = 1, 0 < \|B\|_2 \leq 1$. Then $\|B\|_2^k \leq 1$ for all $k \geq 0$. Assume that $0 < x < 1$. Then for $1 \leq i \leq n$,

$$\left|\left[x^2B^2\left[\sum_{k=0}^{\infty} x^kB^k\right]u\right]_i\right| \leq \left\|\left[x^2B^2\sum_{k=0}^{\infty} x^kB^k\right]u\right\|_2 \leq x^2\|B\|_2^2\left[\sum_{k=0}^{\infty} x^k\|B\|_2^k\right]\|u\|_2$$

$$\leq x^2\|B\|_2^2\left[\sum_{k=0}^{\infty} x^k\right]\|u\|_2 = x^2\|B\|_2^2(1-x)^{-1}\|u\|_2.$$

Since $0 \leq x < 1$, Lemma 1.2 yields

$$(I - xB)^{-1}u = u + xBu + x^2B^2\left[\sum_{k=0}^{\infty} x^kB^k\right]u.$$

For $1 \leq j < n$,

$$[(I - xB)^{-1}u]_j \geq 1 + x[Bu]_j$$

and

$$[(I - xB)^{-1}u]_{j+1} = 1 + x[Bu]_{j+1} + \left[\left[x^2B^2\sum_{k=0}^{\infty} x^kB^k\right]u\right]_{j+1}$$

$$\leq 1 + x[Bu]_{j+1} + x^2\|B\|_2^2(1-x)^{-1}\|u\|_2.$$

It follows that

$$[(I - xB)^{-1}u]_j - [(I - xB)^{-1}u]_{j+1}$$

$$\geq x\left[[Bu]_j - [Bu]_{j+1} - x\|B\|_2^2(1-x)^{-1}\|u\|_2\right].$$

Since $Bu \in \text{int}(D)$, the difference $[Bu]_j - [Bu]_{j+1}$ is strictly positive for $1 \leq j < n$. Thus for sufficiently small, positive x, the terms $[(I-xB)^{-1}u]_j$ are strictly decreasing. Since $I - xB$ is a nonsingular M–matrix for $0 < x < 1, (I - xB)^{-1}u \geq 0$ [BP, Thm. 6.2.3]. Thus $(I - xB)^{-1}u \in \text{int}(D)$ for sufficiently small positive x. Thus ω exists. □

LEMMA 4.4. *Let B be a nonnegative, irreducible matrix with $\rho(B) = 1$. If $X_B \in \text{int}(D)$, then there exists a minimal $\tau = \tau(B)$ such that $0 \leq \tau < 1$, and such that $(I - xB)^{-1}u \in \text{int}(D)$ for $\tau < x < 1$.*

Proof. Let $f(x)$ be the matrix valued function $f(x) = \text{adj}(I - xB)$. Note that $f(x)$ is continuous for all real x. For $0 < x < 1, I - xB$ is an irreducible, nonsingular M–matrix, hence $\det(I-xB) > 0$ and $(I-xB)^{-1}$ is strictly positive [BP, Thm. 6.2.3]. Since $(I - xB)^{-1} = \det(I - xB)\text{adj}(I - xB)$, it follows that $f(x)$ is strictly positive for $0 < x < 1$. Note also, for each x in $0 < x < 1, f(x)u \in \text{int}(D)$ if and only if $(I - xB)^{-1}u \in \text{int}(D)$.

Since B is nonnegative and irreducible, $\rho(B) = 1$ is a simple eigenvalue for B. Thus $\text{rank}(I - B) = n - 1$. Thus $\text{adj}(I - B) \neq 0$. Since $I - B$ has nullity one, its column null space has basis $\{X_B\}$ and its row null space has basis $\{[X_{B^t}]^t\}$. Since $\det(I - B) = 0$, for $x = 1$,

$$(I - xB)\text{adj}(I - xB) = [\text{adj}(I - B)]\,(I - B) = 0.$$

Thus $f(1) = \text{adj}(I - B) = cX_B[X_{B^t}]^t$ for some nonzero scalar c. Since $f(x)$ is continuous at $x = 1$, and since $f(x)$ is strictly positive for $0 < x < 1, f(1)$ is nonnegative. That is, $c > 0$. Then $f(1)u = [c[X_{B^t}]^t u]X_B$ is a positive multiple of X_B, hence $f(1)u \in \text{int}(D)$. Again using continuity at $x = 1$, it follows that τ exists such that $\tau < 1$ and $f(x)u \in \text{int}(D)$ for $\tau < x < 1$. Observe that $\tau \geq 0$ since $f(0)u = Iu \notin \text{int}(D)$. □

The following theorem is an immediate consequence of Lemmas 4.3 and 4.4.

THEOREM 4.5. *Let B be a nonnegative, irreducible matrix with $\rho(B) = 1$. Suppose that Bu and X_B are in $\text{int}(D)$. Let $\omega(B)$ be defined as in Lemma 4.3, and let $\tau(B)$ be defined as in Lemma 4.4. If $\tau(B) < \omega(B)$, then: $\tau(B) = 0, \omega(B) = 1, (I - xB)^{-1} \in \text{int}(D)$ for $0 < x < 1$, and $\text{adj}(I - xB)u \in \text{int}(D)$ for $0 \leq x \leq 1$.*

We currently have no useful general characterization of which matrices B satisfy the condition $\tau(B) < \omega(B)$. The numerical evidence presented in the second section, however, suggests that a substantial portion of the matrices B, such that Bu and X_B share a common grading, do satisfy this condition.

THEOREM 4.6. *Let A be a nonnegative, irreducible matrix with $\rho(A) < 1$. Suppose that A satisfies either* (i) *or* (ii) *in Corollary 4.2. If both Au and X_A are in $\text{int}(D)$, then $(I - A)^{-1}u$ is in $\text{int}(D)$.*

Proof. Let $\rho = \rho(A)$. Let $B = \rho^{-1}A$. Then B is a nonnegative, irreducible matrix with $\rho(B) = 1$. Clearly, $X_B = X_A$. Use (i): $\text{adj}(I - xB) = I + xB_1 + x^2B_2 = I + x(B_1 + xB_2)$. For $x > 0, \text{adj}(I - xB)u$ and $(B_1 + xB_2)u$ have the same gradings. In particular, $\text{adj}(I - xB)u \in \text{int}(D)$ if and only if $(B_1 + xB_2)u \in \text{int}(D)$. Note that for each $i, [(B_1 + xB_2)u]_i$ is a linear function in x. Pick i with $1 \leq i < n$. Let α and β be real numbers such that $\alpha < \beta$. If $[(B_1 + \alpha B_2)u]_i \geq [(B_1 + \alpha B_2)u]_{i+1}$ and if $[(B_1 + \beta B_2)u]_i \geq [(B_1 + \beta B_2)u]_{i+1}$ hold, then by linearity in $x, [(B_1 + xB_2)u]_i \geq [(B_1 + xB_2)u]_{i+1}$ holds for $\alpha \leq x \leq \beta$. Furthermore, if the inequality is strict at either endpoint, then it is strict for $\alpha < x < \beta$.

By hypothesis, $Au \in \text{int}(D)$, hence $Bu \in \text{int}(D)$. Applying Lemma 4.3, there exists ω with $0 < \omega < 1$ such that $(I - xB)^{-1}u \in \text{int}(D)$ for $0 < x < \omega$. For

$0 < x < 1, I - xB$ is a nonsingular M–matrix, and as argued in the proof of Lemma 4.4, $(I - xB)^{-1}u$ is a positive scalar multiple of $\text{adj}(I - xB)u$. Thus, $\text{adj}(I - xB)u \in \text{int}(D)$ for $0 < x < \omega$. Also by hypothesis, $X_B \in \text{int}(D)$, hence from Lemma 4.4 and its proof, there exists a positive τ with $\tau < 1$ such that $\text{adj}(I - xB)u \in \text{int}(D)$ for $\tau < x \leq 1$.

The argument in the preceding paragraph implies that when α is chosen as an arbitrarily small, positive number and when $\beta = 1$, the inequalities for successive entries of $(B_1 + xB_2)u$ are valid and strict for $\alpha < x < \beta$. Thus $\text{adj}(I - xB)u \in \text{int}(D)$ for $0 < x \leq 1$. Hence $(I - xB)^{-1}u \in \text{int}(D)$ for $0 < x < 1$. Since $0 < \rho(A) < 1$, and since $\rho(A)B = A, (I - A)^{-1}u \in \text{int}(D)$. □

The following example shows that the conclusion of Theorem 4.6 can be false if the condition that $Au \in \text{int}(D)$ is dropped. Let $X = (3, 2, 1)^t$. Let B be the parameterized matrix

$$B = \begin{bmatrix} 1 - r & r & r \\ r & r & 2 - 5r \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}.$$

For all $r, BX = X$ and $Bu = (1 + r, 2 - 3r, \frac{1}{2})^t$. For $0 < x < 1$ and $0 < r < \frac{1}{4}, A = xB$ is a nonnegative, irreducible matrix with $\rho(A) = x < 1$. Since A is $3 \times 3, A$ satisfies condition (ii) of Corollary 4.2. Clearly $X_A \in \text{int}(D)$. Note, however, that $Au \notin D$ when $0 < r < \frac{1}{4}$. From Theorem 1.2, $(I - xB)^{-1}u \approx Iu + xBu$ for small, positive x. Thus $(I - A)^{-1}u \notin D$ when $0 < r < \frac{1}{4}$.

5. Exploiting permutation invariance. In the context of this paper, circulant matrices have three important properties. If A is a circulant matrix, then u is an eigenvector for both A and $(I - A)^{-1}, \rho = \rho(A)$ is the unique row sum of A, and $(1 - \rho)^{-1}$ is the unique row sum of $(I - A)^{-1}$. Noting that the circulant matrices are precisely those matrices invariant under permutation similarity by the matrix for the full cycle permutation, we now examine how any permutation invariance can be exploited.

Let A be an $n \times n$, nonnegative, irreducible matrix with spectral radius $\rho(A)$. Suppose that P is a permutation matrix such that $PAP^t = A$. Clearly, $Pu = u$. Furthermore, $P(Au) = Au$ and $P[(I - A)^{-1}u] = (I - A)^{-1}u$. Also, $PAP^t = A$ implies PX_A is a positive eigenvector for $\rho(A)$ with norm one, hence $PX_A = X_A$. Thus the cycle structure of P is reflected in a pattern of constant blocks in the vectors X_A, Au, and $(I - A)^{-1}u$.

Assume that the permutation corresponding to P decomposes into k disjoint cycles. Let $\mathcal{V}$ denote the eigenspace for P for the eigenvalue $\lambda = 1$. Then $\mathcal{V}$ is a k-dimensional subspace of $\mathbb{R}^n$ with a natural basis consisting of certain $\{0, 1\}$ vectors. See [SW, §3]. Furthermore, $u_n \in \mathcal{V}$. Since A and P commute, $\mathcal{V}$ is an A-invariant space. Let M be the $k \times k$ matrix representing the restriction of A to $\mathcal{V}$ with respect to the natural basis. The following can be proven:

(i) M is a nonnegative, irreducible matrix with $\rho(M) = \rho(A)$.

(ii) Each entry of Mu_k is the value of all of the entries in the corresponding block of Au_n.

(iii) Each entry of $(I_k - M)^{-1}u_k$ is the value of all of the entries in the corresponding block of $(I_n - A)^{-1}u_n$.

(iv) There is a normalizing scalar $c > 0$ such that each entry of cX_M is the value of all of the entries in the corresponding block of X_A.

It follows immediately that the gradings for X_M, Mu, and $(I - M)^{-1}u$ lift naturally to gradings for X_A, Au, and $(I - A)^{-1}u$. Finally, when P has the form given by (3.1) of [SW], the matrix A naturally block partitions into blocks $A_{\langle i,j \rangle}$ for $1 \leq i, j \leq k$,

and $M = [m_{ij}]$ is determined uniquely by $m_{ij} = (h_i)^{-1}[u_{h_i}]^t A_{\langle i,j \rangle} u_{h_j}$ for $1 \leq i, j \leq k$. See [SW, §§3 and 4.]

REFERENCES

[BP] A. BERMAN AND R. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[O] R. OLDENBURGER, *Infinite powers of matrices and characteristic roots*, Duke Math. J., 6 (1940), pp. 357–361.

[SW] J. STUART AND J. WEAVER, *Matrices that commute with a permutation matrix*, Linear Algebra Appl., 150 (1991), pp. 255–265.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1167–1181, October 1994

009

COLLINEARITY AND TOTAL LEAST SQUARES*

RICARDO D. FIERRO† AND JAMES R. BUNCH†

Abstract. The least squares (LS) and total least squares (TLS) methods are commonly used to solve the overdetermined system of equations $Ax \approx b$. The main objective of this paper is to examine TLS when A is nearly rank deficient by outlining its differences and similarities to the well-known truncated LS method. It is shown that TLS may be viewed as a regularization technique much like truncated LS, even though the rank reduction depends on b. The sensitivity of LS and TLS approximate nullspaces to perturbations in the data is also examined. Some numerical simulations are included.

Key words. least squares, total least squares, truncated least squares, rank deficient, perturbation theory, acute subspaces

AMS subject classification. 65F20

1. Introduction and preliminaries.

The task of solving the overdetermined set of linear equations

$$Ax \approx b, \quad A \in \Re^{m \times n}, \quad (m \geq n) \tag{1}$$

arises in many disciplines. In many applications (1) does not have an exact solution and the ordinary least squares (LS) approach or total least squares (TLS) approach is commonly used. It is well known (e.g., [1], [4], [7], [10, p. 77], [12]) that collinearities (i.e., near linear dependencies) in the columns of the matrix A can have damaging effects on the ordinary LS estimator, as small changes in A or b may result in disproportionately large changes in the solution. Better results are achieved when stabilization techniques are employed, such as truncated LS, Tikhonov regularization, or subset selection.

Let A have the dyadic singular value decomposition (SVD)

$$A = \sum_{i=1}^{n} u_i' \sigma_i' v_i'^T. \tag{2}$$

Let TOL > 0 be a problem-dependent parameter that identifies the $n-k$ collinearities in A:

$$\sigma_k' > \text{TOL} > \sigma_{k+1}' \geq \cdots \geq \sigma_n'.$$

Then the numerical rank of A is k (with respect to TOL). It is not unreasonable to assume a well-defined gap in the singular value spectrum of A; otherwise, rank determination is a difficult problem, even for the SVD. Here we consider truncated LS: given TOL, one essentially solves

$$\begin{aligned} &\text{minimize } ||b - b_k||_2 \\ &\text{such that } b_k \in \mathcal{R}(A_k), \end{aligned}$$

* Received by the editors January 3, 1992; accepted for publication (in revised form) June 11, 1993.

† Department of Mathematics, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093-0112 (fierro@thunder.csusm.edu and jbunch@ucsd.edu).

where $A_k = \sum_{i=1}^{k} u_i' \sigma_i' v_i'^T$ is the rank-k matrix nearest in 2-norm to $A = A_k + \triangle A_k$, and $\triangle A_k$ the correction in A that is independent of b. Let $x' = \sum_{i=1}^{k} v_i'((b^T u_i')/\sigma_i')$ denote the minimum 2-norm solution to

$$A_k x = b_k,$$

or equivalently, the minimum 2-norm LS solution to

$$\min_x \|A_k x - b\|_2. \tag{3}$$

The minimum 2-norm LS solution x' to (3) always exists and is unique. Also, $\operatorname{rank}(A_k) = \operatorname{rank}([A_k\ b_k]) = k$. See [8] and [9] for more on the properties of the solution.

As in the ordinary LS case, the classical TLS solution (see Golub and Van Loan [5], Van Huffel and Vandewalle [18]) may be affected by collinearities in the matrix A. For reasons of stability a "truncated" TLS solution should be deduced from the span of the singular vectors corresponding to the collinearities in $[A\ b]$, which amounts to a rank reduction in $[A\ b]$. Let $[A\ b]$ have the dyadic SVD

$$[A\ b] = \sum_{i=1}^{n+1} u_i \sigma_i v_i^T. \tag{4}$$

The (truncated) TLS approach chooses $[\hat{A}\ \hat{b}]$ as the minimizer of

$$\text{minimize } \|[A\ b] - [C\ d]\|_2, \tag{5}$$

$$\text{subject to } \operatorname{rank}(C) = \operatorname{rank}([C\ d]) = k. \tag{6}$$

The nearest rank-k matrix in 2-norm to $[A\ b]$ is the best candidate for this problem. If $[\hat{A}\ \hat{b}] \equiv \sum_{i=1}^{k} u_i \sigma_i v_i^T$ satisfies (5)–(6), let $\hat{x}$ denote the minimum 2-norm TLS solution to

$$\hat{A}x = \hat{b}. \tag{7}$$

Also, $\hat{x}$ is the minimum 2-norm LS solution to

$$\min_x \|\hat{A}x - b\|_2.$$

The main objective of this paper is to examine TLS when A is nearly rank deficient by outlining its differences and similarities to the well-known truncated LS method. In §2 we prove an existence condition for the TLS solution and show how it implies that $\hat{A}$ may be viewed as an acute perturbation of A_k. In §3 we use the modified normal equations to elucidate some similarities and differences in the solutions. Then we see how the acuteness condition is related to the differences and similarities in the LS and TLS solutions and residuals. As the ratio σ_{k+1}/σ_k' decreases, TLS may be viewed as a regularization technique much like truncated LS, even though the rank reduction depends on b. In §4 we show how $\sigma_k' - \sigma_{k+1}$ is related to the sensitivity of LS and TLS subspaces to perturbations in the data. Then we explore the role of the orientation of b with respect to the column space of A using subspace perturbation theory and eigenequation relationships. We include some numerical simulations.

At this point we introduce the notation used throughout the paper. Scalars are represented by lowercase Greek letters or English alphabet with double subscripts.

Matrices are represented by the uppercase English alphabet. Superscripts T and the symbol $\dagger$ denote the transpose and the Moore–Penrose pseudoinverse of a matrix, respectively. Let $\|\cdot\| = \|\cdot\|_2$, the Euclidean norm. Let $\mathcal{R}(D)$, $\mathcal{N}(D)$, and $\kappa(D) = \|D\|\,\|D^\dagger\|$ denote the range, nullspace, and the condition number of the matrix D, respectively. $P_D = DD^\dagger$ denotes the orthogonal projection matrix onto $\mathcal{R}(D)$. We also denote the SVD of A by

$$A = U'\Sigma'V'^T \tag{8}$$

with

$$\begin{aligned}
&U' = [U'_1\ U'_2], \quad U'_1 = [u'_1, \ldots, u'_k], \quad U'_2 = [u'_{k+1}, \ldots, u'_n], \quad u'_i \in \Re^m, \quad U'^T U' = I_n,\\
&\Sigma' = \operatorname{diag}(\sigma'_1, \ldots, \sigma'_n) \in \Re^{n\times n}, \quad \sigma'_1 \geq \cdots \geq \sigma'_n \geq 0,\\
&V' = [V'_1\ V'_2], \quad V'_1 = [v'_1, \ldots, v'_k], \quad V'_2 = [v'_{k+1}, \ldots, v'_n], \quad v'_i \in \Re^n, \quad V'^T V' = I_n.
\end{aligned}$$

Partition the matrix Σ' as

$$\Sigma' = \begin{pmatrix} \Sigma'_1 & 0 \\ 0 & \Sigma'_2 \end{pmatrix},$$

so that $\Sigma'_1 \in \Re^{k\times k}$ and $\Sigma'_2 \in \Re^{(n-k)\times(n-k)}$.

We also denote the SVD of $[A\ b]$ by

$$[A\ b] = U\Sigma V^T \tag{9}$$

with

$$\begin{aligned}
&U = [U_1\ U_2], \quad U_1 = [u_1, \ldots, u_k], \quad U_2 = [u_{k+1}, \ldots, u_{n+1}], \quad u_i \in \Re^m, \quad U^T U = I_{n+1},\\
&\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n+1}) \in \Re^{(n+1)\times(n+1)}, \quad \sigma_1 \geq \cdots \geq \sigma_{n+1} \geq 0.\\
&V = [V_1\ V_2], \quad V_1 = [v_1, \ldots, v_k], \quad V_2 = [v_{k+1}, \ldots, v_{n+1}], \quad v_i \in \Re^{n+1}, \quad V^T V = I_{n+1}.
\end{aligned}$$

Partition the matrices V and Σ as

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix},$$

with $V_{11} \in \Re^{n\times k}, V_{12} \in \Re^{n\times(n-k+1)}, V_{21} \in \Re^{1\times k}, V_{22} \in \Re^{1\times(n-k+1)}, \Sigma_1 \in \Re^{k\times k}$, and $\Sigma_2 \in \Re^{(n-k+1)\times(n-k+1)}$.

2. TLS existence and acuteness. In accordance with Golub and Van Loan [5], we bring (1) into the form

$$[A\ b]\begin{bmatrix} x \\ -1 \end{bmatrix} \approx 0. \tag{10}$$

To compute a stabilized TLS solution, we find a reduced-rank approximation of $[A\ b]$. Since A has numerical rank k, it makes sense to approximate $[A\ b] = [\hat{A}\ \hat{b}] + [\triangle\hat{A}\ \triangle\hat{b}]$ by its nearest rank k approximation: $[\hat{A}\ \hat{b}] = U_1\Sigma_1 V_1^T$. $\triangle\hat{A} = A - \hat{A}$ is the correction in A that is dependent on b. Using a nullspace argument [5], [17], [23], the minimum norm TLS solution $\hat{x}$ is given by

$$\hat{x} = -V_{12}V_{22}^\dagger. \tag{11}$$

Note that the solution $\hat{x} \in \Re^n$ exists provided $V_{22} \neq 0$. As in [5], let Q be an $(n-k+1) \times (n-k+1)$ orthogonal matrix such that

$$\begin{bmatrix} V_{12} \\ V_{22} \end{bmatrix} Q = [v_{k+1}, \ldots, v_{n+1}]Q = \begin{bmatrix} W & z \\ 0 & \gamma \end{bmatrix}. \tag{12}$$

By this change of basis, an equivalent representation of the orthogonally invariant minimum 2-norm TLS solution is

$$\hat{x} = -z/\gamma \tag{13}$$

provided $V_{22} \neq 0$. The parameter $|\gamma|$ should be viewed as the distance of the rank-k approximation $[\hat{A}\ \hat{b}]$ from degeneracy, i.e., to incompatibility. That $[\hat{A}\ \hat{b}] = U_1 \Sigma_1 V_1^T$ indeed satisfies the compatibility requirement under a mild sufficiency condition is the content of the following theorem.

THEOREM 2.1. *Let A and $[A\ b]$ have the usual* SVD *and γ as in* (12). *If $\sigma_{k+1} < \sigma'_k$, then $\gamma \neq 0$ and $\hat{x} = -z/\gamma$ exists.*

Proof. Assume $\sigma_{k+1} < \sigma'_k$ and $\gamma = 0$. Then (12) implies $v_{n+1,j} = 0$ for $j = k+1, \ldots, n+1$. By [17, Thm. 3.1], $\sigma_j = \sigma'_{t(j)}$ for $t(j) = j$ or $j-1$, and $v_j = [v'_{t(j)}{}^T, 0]^T$, where $v'_{t(j)} \in \Re^n$ is a right singular vector of A associated with the singular value $\sigma'_{t(j)}$. Once these singular values have been assigned for $j = k+2, \ldots, n+1$, with $t(j) = j$ or $j-1$, it implies $\sigma_{k+1} = \sigma'_k$, a contradiction. Thus, $\gamma \neq 0$.

This theorem was proven more generally for the multidimensional TLS problem in [22] and extended to general orthogonal projection methods in [2]. The condition $\sigma_{k+1} < \sigma'_k$ implies that the nearest rank-k matrix approximation of $[A\ b]$ is unique; also, the condition is numerically mild since $\sigma_{k+1} \leq \sigma'_k$ by the interlacing property of singular values.

The formulas for the solutions $x' = A_k^\dagger b$ and $\hat{x} = -V_{12}V_{22}^\dagger$ in themselves do not provide much of a basis for comparing the two techniques. But as proven in [15], the set S_{TLS} of solutions to (7) is the same as the set of LS solutions to $\hat{A}x \approx b$. Thus,

$$S_{TLS} = \{x | x = \hat{A}^\dagger b + (I - \hat{A}^\dagger \hat{A})z \quad \forall z \in \Re^n\},$$

which is similar to the well-known set S_{LS} of LS solutions

$$S_{LS} = \{x | x = A_k^\dagger b + (I - A_k^\dagger A_k)z \quad \forall z \in \Re^n\}.$$

The minimum 2-norm solutions x' and $\hat{x}$ can be expressed formally as $x' = A_k^\dagger b$ and $\hat{x} = \hat{A}^\dagger b$. Both approaches search for solutions in k-dimensional subspaces; generally, a method that searches in a subspace of lower dimension utilizes less information in A and b. Because we did not distinguish the collinearities in $[A\ b]$ (deemed "predictive" if $|v_{n+1,j}|$ is moderate and "nonpredictive" if $|v_{n+1,j}|$ is small), it is very possible to find solutions x' and $\hat{x}$ in subspaces of higher dimension. However, this is not recommended for reasons of stability.

At this point it seems natural to inquire about how the row and column spaces of $\hat{A}$ differ from A_k when $\hat{A}$ is viewed as a perturbation of A_k, that is, E is defined so that $\hat{A} = A_k + E$. We consider the set of problems (1) where the row and column spaces of $\hat{A}$ and A_k do not differ *drastically* in the sense that the canonical angles between the column spaces, as well as the row spaces, are less than $\pi/2$ (otherwise, we can expect the pseudoinverses to differ significantly as the lower bound on $\|\hat{A}^\dagger - A_k^\dagger\|$

may be arbitrarily large [21] and hence the two methods may generate vastly different solutions). This is the essence of the following definition by Wedin [21].

DEFINITION. *The matrix C is an acute perturbation of D if $\|P_C - P_D\| < 1$ and $\|P_{C^T} - P_{D^T}\| < 1$. We also say that C and D are* acute.

To show that the matrices $\hat{A}$ and A_k are acute, we examine a *reduced form* of the problem:

$$U'^T A_k V' = \begin{pmatrix} \Sigma_1' & 0 \\ 0 & 0 \end{pmatrix}, \tag{14}$$

$$U'^T E V' = \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix}, \tag{15}$$

and

$$U'^T \hat{A} V' = \begin{pmatrix} \tilde{\Sigma}_1 & E_{12} \\ E_{21} & E_{22} \end{pmatrix}, \tag{16}$$

where $\tilde{\Sigma}_1 = \Sigma_1' + E_{11}$. Applying Theorem 3.3 [13, p. 139], A_k and $\hat{A}$ are acute if and only if in the reduced form $\tilde{\Sigma}_1$ is nonsingular and $E_{22} = E_{21}\tilde{\Sigma}_1^{-1}E_{12}$. From these facts we have the following result.

THEOREM 2.2. *Let A and $[A\ b]$ have the* SVD *as in* (8) *and* (9), *and let A_k and $\hat{A}$ have the reduced form as in* (14) *and* (16). *If $\sigma_{k+1} < \sigma_k'$, then $\hat{A}$ and A_k are acute matrices.*

Proof. As mentioned above, we need to show that $\tilde{\Sigma}_1$ is nonsingular and $E_{22} = E_{21}\tilde{\Sigma}_1^{-1}E_{12}$. To show that $\tilde{\Sigma}_1$ is nonsingular, it suffices to show $\|E_{11}\| < \sigma_k'$. Now, E can be rewritten as

$$E = \triangle A_k - \triangle \hat{A}. \tag{17}$$

Transforming both sides of (17) as in (15) yields

$$\begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & \Sigma_2' \end{pmatrix} - \begin{pmatrix} \triangle\hat{A}_{11} & \triangle\hat{A}_{12} \\ \triangle\hat{A}_{21} & \triangle\hat{A}_{22} \end{pmatrix}, \tag{18}$$

where

$$U'^T \triangle \hat{A} V' = \begin{pmatrix} \triangle\hat{A}_{11} & \triangle\hat{A}_{12} \\ \triangle\hat{A}_{21} & \triangle\hat{A}_{22} \end{pmatrix}.$$

Since $\triangle\hat{A}_{11}$ is a submatrix of $U'^T \triangle \hat{A} V'$, we have

$$\|\triangle\hat{A}_{11}\| \le \|U'^T \triangle \hat{A} V'\| = \|\triangle\hat{A}\| \le \|[\triangle\hat{A}\ \triangle\hat{b}]\| = \sigma_{k+1}.$$

From (18) and the assumption $\sigma_{k+1} < \sigma_k'$, this implies $\|E_{11}\| < \sigma_k'$, and hence $\tilde{\Sigma}_1$ is nonsingular. To show $E_{22} = E_{21}\tilde{\Sigma}_1^{-1}E_{12}$, define $S \equiv E_{22} - E_{21}\tilde{\Sigma}_1^{-1}E_{12}$. Performing a step of block Gaussian elimination on $U'^T \hat{A} V'$ gives

$$U'^T \hat{A} V' = \begin{bmatrix} I_k & 0 \\ E_{21}\tilde{\Sigma}_1^{-1} & I_{n-k+1} \end{bmatrix} S_1,$$

where

$$S_1 \equiv \begin{bmatrix} \tilde{\Sigma}_1 & E_{12} \\ 0 & S \end{bmatrix}.$$

If $S \neq 0$, then $\operatorname{rank}(S_1) \geq k+1$, implying $k = \operatorname{rank}(\hat{A}) = \operatorname{rank}(U'^T \hat{A} V') = \operatorname{rank}(S_1) \geq k+1$, a contradiction. Therefore, $S \equiv 0$. This completes the proof of the theorem.

This shows that $\sigma_{k+1} < \sigma'_k$ implies A_k and $\hat{A}$ are acute; that is, the canonical angles between their column spaces, as well as their row spaces, are less than $\pi/2$. This agrees with the intuition that if the fitting of the data applied by the TLS approach is "close" in the sense of acuteness to the fitting of the data applied by the truncated LS approach, then the former approach should have a solution since the truncated LS solution always exists. So the class of problems (1) that we consider in comparing the TLS and LS solutions and residuals consists of the problems for which $\sigma_{k+1} < \sigma'_k$. When this condition is satisfied, $\hat{A}^\dagger$ is a continuous function of the perturbation of the elements of A_k.

3. Bounds on the minimum norm LS and TLS solutions and residuals. Since the minimum norm LS and TLS solutions can be expressed $x' = A_k^\dagger b$ and $\hat{x} = \hat{A}^\dagger b$, they satisfy the modified normal equations

$$(A^T A - \triangle A_k{}^T \triangle A_k) x' = A^T b - \triangle A_k{}^T b, \tag{19}$$

$$(A^T A - \triangle \hat{A}^T \triangle \hat{A}) \hat{x} = A^T b - \triangle \hat{A}^T b. \tag{20}$$

These alternative relationships are useful in deducing the equivalences and differences between the two techniques. As Σ'_2 goes to zero and the LS problem becomes more compatible, then $\hat{x}$ approaches x'. In fact, for $\Sigma'_2 = 0$ and $b \in \mathcal{R}(A)$ then $\Sigma_2 = 0$ and $\hat{x}$ equals x' (also, see [16, Thm. 2.1] by Van Huffel and Vandewalle). To elucidate the differences in the solutions, we will use a first order expansion of SVD components. Let $S'_1 = \Sigma_1'^T \Sigma'_1$, $S_1 = \Sigma_1^T \Sigma_1$, and $S_2 = \Sigma_2^T \Sigma_2$. Equations (19) and (20) can be rewritten as

$$(V'_1 S'_1 V_1'^T) x' = A^T b - V'_2 \Sigma_2'^T U_2'^T b, \tag{21}$$

$$(V_{11} S_1 V_{11}^T) \hat{x} = A^T b - V_{12} S_2 V_{22}^T. \tag{22}$$

Substituting (22) into (21) yields

$$(V'_1 S'_1 V_1'^T) x' = (V_{11} S_1 V_{11}^T) \hat{x} + V_{12} S_2 V_{22}^T - V'_2 \Sigma_2'^T U_2'^T b.$$

Define the difference matrices E_S and E_V by $E_S = S_1 - S'_1$ and $E_V = V_{11} - V'_1$. Substituting $V_{11} = V'_1 + E_V$ and $S_1 = S'_1 + E_S$ yields to the first order (ignore terms involving the product of two or more difference matrices):

$$(V'_1 S'_1 V_1'^T)(x' - \hat{x}) \approx (V'_1 S'_1 E_V^T + E_V S'_1 V_1'^T + V'_1 E_S V_1'^T) \hat{x} + V_{12} S_2 V_{22}^T - V'_2 \Sigma_2'^T U_2'^T b.$$

Using the fact $\sigma_k'^2 \|x' - \hat{x}\| = \sigma_{\min}(V'_1 S'_1 V_1'^T) \|x' - \hat{x}\| \leq \|(V'_1 S'_1 V_1'^T)(x' - \hat{x})\|$, we get

$$\|x' - \hat{x}\| \leq \frac{1}{\sigma_k'^2} [(2\|E_V\|\sigma_1^2 + \|E_S\|)\|\hat{x}\| + \sigma_{k+1}^2 + \sigma'_{k+1} \|U_2'^T b\|]. \tag{23}$$

Equation (23) is not a tight bound for the norm $\|x' - \hat{x}\|$, but might suggest that we can expect the solutions x' and $\hat{x}$ to be very different when any of the following hold:

- σ'_k is small, implying A_k is ill conditioned.
- σ_{k+1} is large because either $Ax \approx b$ is highly incompatible or there is not a good gap in the singular value distribution of A.
- The largest singular values of A and $[A\ b]$ differ greatly, so that $\|E_S\|$ is large. As shown later this happens when b is oriented toward the larger left singular vectors of A.

Next, we determine a bound on the difference $A_k^\dagger - \hat{A}^\dagger$ relative to $A_k^\dagger$ as well as a bound on the difference $x' - \hat{x}$ relative to x'. The acuteness condition appears in the bounds.

LEMMA 3.1. *Assume* $\sigma_{k+1} < \sigma'_k$ *and define* $\phi_k \equiv \sigma_{k+1}/\sigma'_k$, $\mu = (1+\sqrt{5})/2$. *Then the difference* $A_k^\dagger - \hat{A}^\dagger$ *relative to* $A_k^\dagger$ *is bounded above by*

$$\frac{\|A_k^\dagger - \hat{A}^\dagger\|}{\|A_k^\dagger\|} \leq 2\mu \frac{\phi_k}{1-\phi_k}.$$

Proof. Since $\operatorname{rank}(A_k) = \operatorname{rank}(\hat{A})$, applying [15, Thm. 3.14], we have

$$\|A_k^\dagger - \hat{A}^\dagger\| \leq \mu \|A_k^\dagger\| \|\hat{A}^\dagger\| \|A_k - \hat{A}\|. \tag{24}$$

Using $A - \hat{A} = \triangle\hat{A}$ and $\|\triangle\hat{A}\| \leq \sigma_{k+1}$, it follows from the perturbation of singular values [6, p. 287] that $\|\hat{A}^\dagger\| \leq (\sigma'_k - \sigma_{k+1})^{-1}$. Furthermore, using the triangle inequality,

$$\|A_k - \hat{A}\| \leq \|A_k - A\| + \|A - \hat{A}\| \leq \sigma'_{k+1} + \sigma_{k+1} \leq 2\,\sigma_{k+1}. \tag{25}$$

Thus

$$\frac{\|A_k^\dagger - \hat{A}^\dagger\|}{\|A_k^\dagger\|} \leq 2\mu \frac{\sigma_{k+1}}{\sigma'_k - \sigma_{k+1}},$$

which is equivalent to the desired result.

This lemma implies that the solutions x' and $\hat{x}$ should be relatively close if $\phi_k \ll 1$. In other words, if there is a well-defined gap in the singular value spectrum of A ($\sigma'_{k+1} \ll \sigma'_k$) and $Ax \approx b$ is not too incompatible, then we expect the solutions to be relatively close.

Defining $\phi_k \equiv \sigma_{k+1}/\sigma'_k$ as above and $r' \equiv b - Ax'$, we have the following bound for the relative difference in the solutions x' and $\hat{x}$.

THEOREM 3.2. *Let* A *and* $[A\ b]$ *have the usual* SVD. *Let* x' *be the minimum norm* LS *solution to* (3) *and* $\hat{x}$ *the minimum norm* TLS *solution to* (7). *If* $\sigma_{k+1} < \frac{1}{2}\sigma'_k$ *then*

$$\frac{\|x' - \hat{x}\|}{\|x'\|} \leq \frac{\phi_k}{1-\phi_k}\left[3 + \kappa(A_k)\frac{\|r'\|}{\|b\|}\right].$$

Proof. The result follows from [8, (27a)] by setting $e = 0$, by viewing $\hat{A}$ as an acute perturbation of the rank-k matrix A_k, $\|\hat{A} - A_k\| \leq 2\sigma_{k+1}$, and by using the fact $\frac{2\,\phi_k}{1-\phi_k} + \phi_k \leq \frac{3\,\phi_k}{1-\phi_k}$.

Bounds on the norm $\|x' - \hat{x}\|$ for the two cases $k = n$ and $\sigma_{k+1} = \cdots = \sigma_{n+1}$ are derived in [18] and [22], respectively. While the bound in Theorem 3.2 holds for the more general case, it is overly pessimistic. However, it still gives a correct indication of the influence of ϕ_k.

The purpose of regularization techniques is to remove the potential instability in the problem due to finite precision and data errors. The regularization technique of Tikhonov [14] is widely known to be an effective method for solving the ill-conditioned LS problem. Hansen [8] has shown that when A has a well-defined gap in its singular value spectrum, then truncated LS is similar to Tikhonov regularization. The truncated LS method removes the potential instability by solving a nearby, alternative LS problem $A_k x \approx b$, where, roughly speaking, one strives to balance the condition number of A_k and the size of $\|A - A_k\|$. From this point of view and the fact that $\hat{x}$ solves the ordinary LS problem $\hat{A}x \approx b$, where $\hat{A}$ is an acute perturbation of A_k that depends on b, the TLS approach to (1) is a regularization method similar to truncated LS, but the condition number of $\hat{A}$ is larger than that of A_k.

Next, we provide a bound on the residuals. Let $r' = b - Ax' = b - A_k x'$, $\hat{r} = b - A\hat{x}$, and $\tilde{r} = b - \hat{A}\hat{x}$. Let $\phi_k \equiv \sigma_{k+1}/\sigma'_k$.

THEOREM 3.3. *Let A and $[A\ b]$ have the usual* SVD *and assume $\sigma_{k+1} < \sigma'_k$. Let x' be the minimum norm* LS *solution to* (3) *and $\hat{x}$ be the minimum norm solution to* (7)*, and let r' and $\hat{r}$ be defined as above. Then the following hold:*

$$\frac{\|r' - \hat{r}\|}{\|b\|} \leq 2\frac{\phi_k}{1 - \phi_k} \quad \text{and} \quad \|\hat{r}\| \leq \sigma_{k+1}\sqrt{1 + \|\hat{x}\|^2}.$$

Proof. To bound the relative error, we use the triangle inequality $\|r' - \hat{r}\| \leq \|r' - \tilde{r}\| + \|\tilde{r} - \hat{r}\|$. To bound $\|r' - \tilde{r}\|$, rewriting the residuals as $r' = (I_m - P_{U'_1})b$ and $\tilde{r} = (I_m - P_{U_1})b$ means $r' - \tilde{r} = (P_{U_1} - P_{U'_1})b$, with

$$\|P_{U_1} - P_{U'_1}\| = \|U_1 U_1^T - U'_1 {U'_1}^T\| = \|{U'_1}^T U_2\|.$$

Using $A = U_1 \Sigma_1 V_{11}^T + U_2 \Sigma_2 V_{12}^T$, and ${U'_1}^T = ({\Sigma'_1}^+){V'_1}^T A^T$, it follows that

$${U'_1}^T U_2 = ({\Sigma'_1}^+)^T {V'_1}^T V_{12} \Sigma_2^T.$$

Thus,

$$\|{U'_1}^T U_2\| \leq \|{\Sigma'_1}^+\| \, \|\Sigma_2\| = \phi_k$$

and hence

$$\frac{\|r' - \tilde{r}\|}{\|b\|} \leq \|P_{U_1} - P_{U'_1}\| \leq \phi_k. \tag{26}$$

To bound $\|\tilde{r} - \hat{r}\|$, from $\hat{r} - \tilde{r} = (\hat{A} - A)\hat{x}$, we get

$$\|\tilde{r} - \hat{r}\| \leq \|\hat{A} - A\| \, \|\hat{x}\| \leq \sigma_{k+1}\|\hat{x}\|.$$

But $\hat{x} = \hat{A}^\dagger b$ implies $\|\hat{x}\| \leq \|\hat{A}^\dagger\| \, \|b\| \leq \|b\|/(\sigma'_k - \sigma_{k+1})$. Thus,

$$\frac{\|\tilde{r} - \hat{r}\|}{\|b\|} \leq \frac{\sigma_{k+1}}{\sigma'_k - \sigma_{k+1}} = \frac{\phi_k}{1 - \phi_k}. \tag{27}$$

The sum of the two inequalities in (26) and (27) is bounded by the desired result. To bound the residual, $\hat{r} = b - A\hat{x} = -[\triangle\hat{A} \ \triangle\hat{b}][\hat{x}^T \ -1]^T$, and taking norms in the obvious way yields the desired result. This completes the proof of the theorem.

The bound in Theorem 3.3 for the difference in the residuals is a monotonic function of ϕ_k. As ϕ_k increases, so does the expected difference in the residuals. The residuals approach each other as ϕ_k decreases.

4. Perturbation of LS and TLS approximate nullspaces. In this section we examine the sensitivity of the LS and TLS approximate nullspaces. This is important because, as we see in Theorem 4.1, perturbations can affect the truncated LS and TLS solutions. Let $[\tilde{A}\ \tilde{b}] = [A\ b] + [\triangle A\ \triangle b]$ denote a perturbation of $[A\ b]$ with the dyadic SVD

$$[\tilde{A}\ \tilde{b}] = \sum_{i=1}^{n+1} \tilde{\sigma}_i \tilde{u}_i \tilde{v}_i^T. \tag{28}$$

Let $[\check{A}\ \check{b}]$ represent the nearest rank k matrix to $[\tilde{A}\ \tilde{b}]$ and $\tilde{A}_k$ the nearest rank k matrix to $\tilde{A}$, and let $\tilde{b}_k$ denote the projection of $\tilde{b}$ onto $\mathcal{R}(\tilde{A}_k)$. As usual, we assume the data A is nearly rank-$(n-k)$ deficient. In direct analogy with TLS in (12)–(13), the minimum norm LS solutions to

$$\min_x \|A_k x - b\| \quad \text{and} \quad \min_x \|\tilde{A}_k x - \tilde{b}\|$$

can be determined by solving the homogeneous equations

$$[A_k\ b_k] \begin{bmatrix} x \\ -1 \end{bmatrix} = 0 \quad \text{and} \quad [\tilde{A}_k\ \tilde{b}_k] \begin{bmatrix} x \\ -1 \end{bmatrix} = 0 \tag{29}$$

using a nullspace argument. Thus it makes sense to compare the distance between the LS approximate nullspaces $\mathcal{N}([A_k\ b_k])$ and $\mathcal{N}([\tilde{A}_k\ \tilde{b}_k])$ and the distance between the TLS approximate nullspaces $\mathcal{N}([\hat{A}\ \hat{b}])$ and $\mathcal{N}([\check{A}\ \check{b}])$ (see [6, p. 76] for definition of distance between subspaces).

When the size of the perturbation $\|[\triangle A\ \triangle b]\|$ satisfies

$$\|[\triangle A\ \triangle b]\| < \sigma'_k - \sigma'_{k+1} \tag{30}$$

then $\mathcal{R}(A_k)$ and $\mathcal{R}(\tilde{A}_k)$ are acute subspaces. Since the LS solutions always exist and can be deduced from the the basis vectors of $\mathcal{N}([A_k\ b_k])$ and $\mathcal{N}([\tilde{A}_k\ \tilde{b}_k])$, it makes sense to compare the distance between these LS approximate nullspaces. Similarly,

$$\|[\triangle A\ \triangle b]\| < \sigma_k - \sigma_{k+1} \tag{31}$$

ensures that $\mathcal{N}([\hat{A}\ \hat{b}])$ and $\mathcal{N}([\check{A}\ \check{b}])$ are acute subspaces. Equations (30)–(31) are tantamount to requiring that a singular value in the unperturbed cluster of small singular values remains in the corresponding perturbed cluster. Note that these conditions hold whenever

$$\|[\triangle A\ \triangle b]\| < \sigma'_k - \sigma_{k+1}. \tag{32}$$

Hence the existence condition for TLS can be directly related to the sensitivity of the subspaces to acute perturbations. When (32) is satisfied, the perturbed TLS problem cannot be *nongeneric* (nongeneric implies $\tilde{\sigma}'_k = \tilde{\sigma}_{k+1}$) in the sense of Van Huffel and Vandewalle [17].

Now, let us assume we are given an orthonormal basis for the nullspace of $[A_k\ b_k]$ ($[\hat{A}\ \hat{b}]$) and an orthonormal basis for the nullspace of its perturbed version $[\tilde{A}_k\ \tilde{b}_k]$ ($[\check{A}\ \check{b}]$). Then we can perform a change of basis as in (12) and compute the LS (TLS) solutions as in (13); partition the unit vectors

$$v = \begin{pmatrix} z \\ \gamma \end{pmatrix} \quad \text{and} \quad \tilde{v} = \begin{pmatrix} \tilde{z} \\ \tilde{\gamma} \end{pmatrix}, \tag{33}$$

where $\tilde{v}$ is considered the perturbed version of v. If $\|[\triangle A\ \triangle b]\| < \sigma'_k - \sigma_{k+1}$, then $\min(\gamma, \tilde{\gamma}) \neq 0$. Define $x \equiv -z/\gamma$ and $\tilde{x} \equiv -\tilde{z}/\tilde{\gamma}$. Then it follows $|\gamma|^{-1} = \sqrt{1 + \|x\|^2}$ and $|\tilde{\gamma}|^{-1} = \sqrt{1 + \|\tilde{x}\|^2}$. The effect of the perturbation $[\triangle A\ \triangle b]$ on v is measured by θ, the angle between v and $\tilde{v}$. Next we show how this angle influences $\|x - \tilde{x}\|$.

THEOREM 4.1. *Define the vectors v and $\tilde{v}$ as in* (33), *and define $x = -z/\gamma$ and $\tilde{x} = -\tilde{z}/\tilde{\gamma}$. Assume* $\operatorname{sign}(\gamma) = \operatorname{sign}(\tilde{\gamma})$. *If $\|[\triangle A\ \triangle b]\| < \sigma'_k - \sigma_{k+1}$, then*

$$\|x - \tilde{x}\| \leq \frac{2\sqrt{2}\,\sin\theta}{|\gamma\,\tilde{\gamma}|},$$

where θ is the angle between v and $\tilde{v}$.

Proof. Since v and $\tilde{v}$ are unit vectors, we have $\|v - \tilde{v}\| = \sqrt{2(1 - \cos\theta)} \leq \sqrt{2}\,\sin\theta$, $\|z\| \leq 1$, and

$$\|v - \tilde{v}\|^2 = \|z - \tilde{z}\|^2 + (\gamma - \tilde{\gamma})^2. \tag{34}$$

Thus, we get $\|z - \tilde{z}\| \leq \sqrt{2}\,\sin\theta$ and $|\gamma - \tilde{\gamma}| \leq \sqrt{2}\,\sin\theta$. It follows that

$$\begin{aligned}
\|x - \tilde{x}\| &= \|\tilde{z}\tilde{\gamma}^{-1} - z\gamma^{-1}\| \\
&= \|(\tilde{z} - z)\tilde{\gamma}^{-1} - z(\gamma^{-1} - \tilde{\gamma}^{-1})\| \\
&\leq \|(\tilde{z} - z)\tilde{\gamma}^{-1}\| + \|z(\gamma^{-1} - \tilde{\gamma}^{-1})\| \\
&\leq \sqrt{2}\,\sin\theta\,|\tilde{\gamma}^{-1}| + |\gamma^{-1} - \tilde{\gamma}^{-1}| \\
&= \sqrt{2}\,\sin\theta\,|\tilde{\gamma}^{-1}| + \frac{|\gamma - \tilde{\gamma}|}{|\gamma\,\tilde{\gamma}|} \\
&\leq \frac{2\sqrt{2}\,\sin\theta}{|\gamma\,\tilde{\gamma}|}.
\end{aligned}$$

This concludes the proof of the theorem.

The following analysis is motivated by Theorem 4.1 and by the analysis in [19, pp. 215–217] by Van Huffel and Vandewalle; however, it differs in that we now compare the distance between corresponding approximate nullspaces (since the LS problems in (29) are reformulated as TLS-like problems). However, as we see, we draw the same conclusions as in [19, pp. 220–225].

THEOREM 4.2. *Let $[\tilde{A}\ \tilde{b}] = [A\ b] + [\triangle A\ \triangle b]$ with the above* SVD*s and assume $\|[\triangle A\ \triangle b]\| < \sigma'_k - \sigma_{k+1}$. Then the following relations hold:*

$$\text{(i)} \qquad \operatorname{dist}\left(\mathcal{N}([A_k\ b_k]), \mathcal{N}([\tilde{A}_k\ \tilde{b}_k])\right) \leq \frac{\|[\triangle A\ \triangle b]\|}{\sigma'_k} + \alpha_{LS},$$

$$\text{(ii)} \qquad \operatorname{dist}\left(\mathcal{N}([\hat{A}\ \hat{b}]), \mathcal{N}([\check{A}\ \check{b}])\right) \leq \frac{\|[\triangle A\ \triangle b]\|}{\sigma_k - \sigma_{k+1} - \|[\triangle A\ \triangle b]\|},$$

where

$$\alpha_{LS} = \mu \frac{\left(\operatorname{dist}(\mathcal{R}(A_k), \mathcal{R}(\tilde{A}_k))\,\|[A\ b]\| + \|[\triangle A\ \triangle b]\|\right)\,\|[\triangle \tilde{A}_k\ \tilde{r}]\|}{\sigma'_k\,(\sigma'_k - \|\triangle A\|)},$$

$\tilde{r} = (I - \tilde{A}_k \tilde{A}_k^{\dagger})\tilde{b}$, $\operatorname{dist}(S_1, S_2) = \|P_{S_1} - P_{S_2}\|$, *and P_{S_i} is the orthogonal projector onto subspace S_i.*

Proof. Using the augmented matrices and some algebraic manipulation (see [3] for details), part (i) follows from extending a result (see [3, Thm. 3]) of Van der Sluis and Veltkamp [15, p. 269] to bound the LS nullspaces:

$$\|[A_k\ b_k]^\dagger[A_k\ b_k] - [\tilde{A}_k\ \tilde{b}_k]^\dagger[\tilde{A}_k\ \tilde{b}_k]\| \le \|[A_k\ b_k]^\dagger\|\,\|[\triangle A\ \triangle b]\| + t_{LS}, \tag{35}$$

where

$$\begin{aligned} t_{LS} &= \mu\|[A_k\ b_k]^\dagger\|\,\|[\tilde{A}_k\ \tilde{b}_k]^\dagger\|\,\|[A_k\ b_k] - [\tilde{A}_k\ \tilde{b}_k]\|\,\|[\triangle\tilde{A}_k\ \tilde{r}]\| \\ &\le \mu\frac{(\operatorname{dist}(\mathcal{R}(A_k),\mathcal{R}(\tilde{A}_k))\,\|[A\ b]\| + \|[\triangle A\ \triangle b]\|)\,\|[\triangle\tilde{A}_k\ \tilde{r}]\|}{\sigma'_k\,(\sigma'_k - \|\triangle A\|)}. \end{aligned}$$

Part (ii) follows from [19, pp. 220–225] using Wedin's perturbation bounds in connection with the SVD and the perturbation property of singular values.

The relation between these perturbation bounds and the solutions is addressed in [3]. Usually,

$$\alpha_{LS} << \frac{\|[\triangle A\ \triangle b]\|}{\sigma'_k},$$

and we expect the bounds on the distance between the TLS approximate nullspaces to be less than the LS approximate nullspaces whenever

$$\sigma_{k+1} + \|[\triangle A\ \triangle b]\| < \sigma_k - \sigma'_k. \tag{36}$$

If $O(10^{-t}) = \max(O(\sigma_{k+1}), O(\|[\triangle A\ \triangle b]\|))$, then for inequality (36) to fail, σ_k and σ'_k must agree in at least the first t decimal digits! It is interesting to see how the existence condition for the TLS solution resurfaces—this time in the inequality that determines when the TLS bound is lower than the LS bound. Provided $Ax \approx b$ is not too incompatible, A has a well-defined gap in its singular values and $\|[\triangle A\ \triangle b]\|$ is not too large, this inequality will almost always be satisfied.

Now, we argue that $\sigma_k - \sigma'_k$ may increase as the component of b along u'_k increases. Premultiplying the eigenequations

$$[A\ b][A\ b]^T u_k = \sigma_k^2 u_k \tag{37}$$

by ${u'_k}^T$ yields

$${\sigma'_k}^2 {u'_k}^T u_k + ({u'_k}^T b)(b^T u_k) = \sigma_k^2 {u'_k}^T u_k.$$

Therefore, it follows

$$({\sigma_k}^2 - {\sigma'_k}^2){u'_k}^T u_k = ({u'_k}^T b)(b^T u_k)$$

and hence

$${\sigma_k}^2 - {\sigma'_k}^2 \ge ({\sigma_k}^2 - {\sigma'_k}^2)|{u'_k}^T u_k| = |{u'_k}^T b|\,|b^T u_k|. \tag{38}$$

From the dyadic SVD form of $[A\ b]$ in (4) we know $|b^T u_k| = \sigma_k |v_{n+1,k}|$, where $v_{n+1,k}$ is the last component of the vector v_k. If $|{u'_k}^T b| \ge |b^T u_k|$, then it follows

$$\frac{{\sigma_k}^2 - {\sigma'_k}^2}{\sigma_k^2} \ge |v_{n+1,k}|^2.$$

Note that if $v_{n+1,k} = 0$ then b is orthogonal to u_k and moreover from [17, Thm. 3.1] $v_{n+1,k} = 0$ implies $b \perp u'_k$. Thus $b \not\perp u'_k$ implies $v_{n+1,k} \neq 0$. Now, let us assume $|b^T u_k| \geq |{u'_k}^T b|$. Then from (38) we have

$$\sigma_k{}^2 - {\sigma'_k}^2 \geq |{u'_k}^T b|^2.$$

From this we conclude that $\sigma_k - \sigma'_k$ may increase as the component of b along u'_k increases (i.e., as $|{u'_k}^T b|$ increases), and verily so if $|b^T u_k| \geq |{u'_k}^T b|$. We note that this conclusion, as well as the following numerical experiments, confirm the theoretical analysis given in [19, pp. 215–217, 220–225]. Finally, a similar argument shows that the larger singular values of A and $[A\ b]$ may increasingly differ as b is oriented along the larger left singular vectors of A, hence $\|E_S\|$ in §3 may be large.

That the orientation of b and the bounds of the LS and TLS numerical nullspaces are related are exemplified in the following numerical simulations using components of the the mean squared error (MSE) (see [7]). The simulation is designed as follows. We perturbed an exact system $Ax = b$ with an error matrix $[\triangle A\ \triangle b]$ whose entries are Gaussian with 0 mean and δ standard deviation. δ^2 ranged from 10^{-4} to 10^{-2}. The MSE of the estimator x, where x is either x' or $\hat{x}$, is defined by

$$\text{MSE}(x) = E[(x - E(x))^T (x - E(x))] + (E(x) - x_0)^T (E(x) - x_0), \tag{39}$$

where E is the expected value operator, x_0 is the exact solution, the first term is the TOTAL VARIANCE ("wobbliness") and the second is the SQUARED BIAS ("bias"). In these simulations, given the vector $x_0 \in \Re^{10}$, we set $b \equiv Ax_0$. All simulations were carried out using Pro-MATLAB [11] on a DECstation 3100.

- We plotted the TOTAL VARIANCE versus the SQUARED BIAS in a log–log diagram.
- $A \in \Re^{50\times 10}$ has singular values

$$\sigma(A) = \{10, 7, 7, 3, 2, 1, 0.1, 0.0005, 0.0001, 0.00001\}.$$

In 2-norm, A is within 0.0005 of a well-conditioned matrix of rank $k = 7$.

- For a fixed variance δ^2, the truncated LS and TLS solutions of 100 independent sets of equations $(A + \triangle A)x \approx (b + \triangle b)$ were computed. The mean of the trials was used to compute the expected values. The simulation procedure was repeated for increasing values of δ^2.

We examine three cases.

- *Case* 1. The exact solution is $x_0 = v'_1 + v'_2$, which means b is oriented along the larger (left) singular vectors of A.
- *Case* 2. The exact solution is $x_0 = v'_6 + v'_7$, which means b is oriented along the $(k-1)$st and kth (left) singular vectors of A.
- *Case* 3. The exact solution x_0 is a random vector in $\Re^{10}$.

In Fig. 1, b is oriented along the larger left singular vectors of A and neither method is favored in the experiment (or theory). In Fig. 2, b is oriented along the smaller left singular vectors, which increases $\sigma_k - \sigma'_k$, and TLS performs much better as expected. In Fig. 3, b is randomly oriented in $\mathcal{R}(A)$ and the results show that TLS can indeed be viewed as a regularization technique much like truncated LS. In all these examples TLS tends to have a higher TOTAL VARIANCE than LS and consequently the TLS curves extend further to the right.

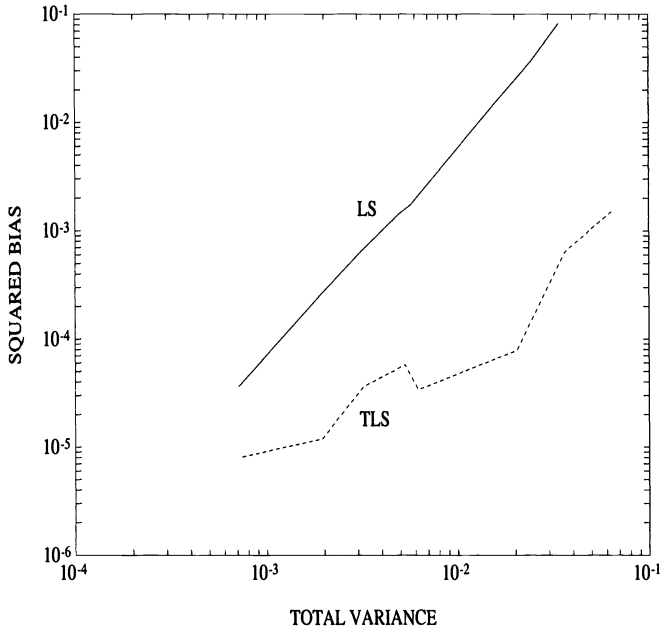


FIG. 1. *The squared bias versus total variance for truncated* LS *and* TLS *with the observation vector b oriented along the larger singular vectors of A. The squared bias is nearly equal for both methods under small perturbations of the exact system.*

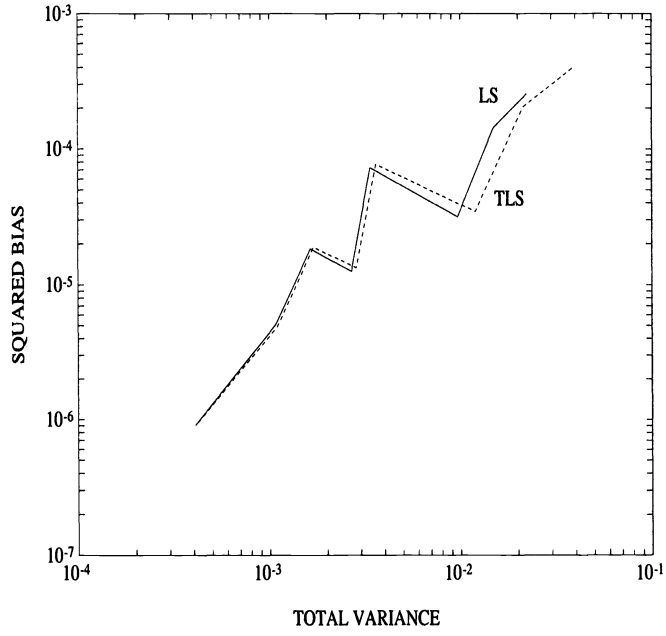


FIG. 2. *The squared bias versus total variance for truncated* LS *and* TLS *with the observation vector b oriented along the* $(k-1)$th *and* kth *left singular vectors of A. In the same domain of total variance, the squared bias for truncated* LS *is higher than* TLS. *The disparity increases as the total variance increases.*

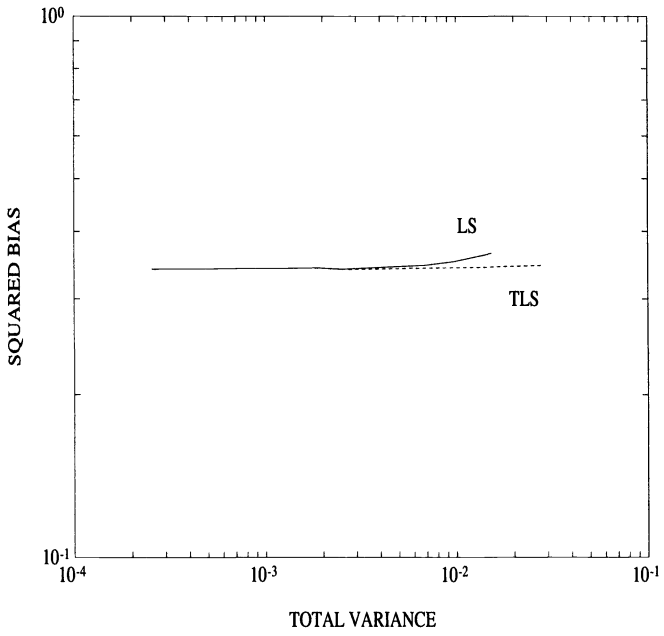


FIG. 3. *The squared bias versus total variance for truncated* LS *and* TLS *with the observation vector b randomly oriented in the range of A.* TLS *performance is nearly the same as* LS.

5. Conclusions. In this paper we examined TLS when A is nearly rank deficient by outlining differences and similarities to the well-known truncated LS method. It was shown how $\hat{A}$ may be viewed as an acute perturbation of A_k, how differences and similarities depend on both $\sigma'_k - \sigma_{k+1}$ and σ_{k+1}/σ'_k, and how $\sigma'_k - \sigma_{k+1}$ is related to the sensitivity of LS and TLS subspaces to perturbations in the data. Also, using the modified normal equations, we elucidated some similarities and differences in the solutions. Perturbation theory for the approximate nullspaces and eigenequation relationships elucidated the role of the orientation of b with respect to the column space of A.

Acknowledgments. We wish to thank Sabine Van Huffel, Per Christian Hansen, and the referees for their valuable suggestions in improving the paper.

REFERENCES

[1] L. ELDÉN, *Algorithms for the regularization of ill-conditioned least squares problems*, BIT, 17 (1977), pp. 134–145.

[2] R. D. FIERRO AND J. R. BUNCH, *Orthogonal projection and total least squares*, Numer. Linear Algebra Appl., to appear.

[3] ———, *Perturbation Theory for Orthogonal Projection Methods with Applications to Least Squares and Total Least Squares*, Linear Algebra Appl., to appear.

[4] G. H. GOLUB, V. KLEMA, AND G. W. STEWART, *Rank Degeneracy and Least Squares Problems*, Tech. Report TR-456, Dept. of Computer Science, University of Maryland, College Park, 1976.

[5] 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.

[6] ———, *Matrix Computations*, 2nd ed., John Hopkins University Press, Baltimore, MD, 1989.

[7] R. F. GUNST AND R. L. MASON, *Biased estimation in regression: An evaluation using mean squared error*, J. Amer. Statist. Assoc., 72 (1977), pp. 616–628.
[8] P. C. HANSEN, *The truncated* SVD *as a method for regularization*, BIT, 27 (1987), pp. 534–553.
[9] ———, *Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 503–518.
[10] C. L. LAWSON AND R. J. HANSEN, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[11] C. B. MOLER, J. LITTLE, AND S. BANGERT, *Pro-MATLAB User's Guide*, The Math Works Inc., Sherborn, MA, 1987.
[12] G. W. STEWART, *Rank degeneracy*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 403–413.
[13] G. W. STEWART AND JI-GUANG SUN, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[14] A. N. TIKHONOV, *Solution of incorrectly formulated problems and the regularization method*, Soviet Math. Dokl., 4 (1963), pp. 1035–1038.
[15] A. VAN DER SLUIS AND G. W. VELTKAMP, *Restoring rank and consistency by orthogonal projection*, Linear Algebra Appl., 28 (1979), pp. 257–278.
[16] S. VAN HUFFEL AND J. VANDEWALLE, *Algebraic relationships between classical regression and total least squares estimation*, Linear Algebra Appl., 93 (1987), pp. 149–160.
[17] ———, *Analysis and solution of the nongeneric total least squares problem*, SIAM J. Matrix Anal. Appl., 3 (1988), pp. 360–372.
[18] ———, *Algebraic connections between the least squares and total least squares problems*, Numer. Math., 55 (1989), pp. 4431–4449.
[19] ———, *The Total Least Squares Problem: Computational Aspects and Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.
[20] P. Å. WEDIN, *Perturbation bounds in connection with the singular value decomposition*, BIT, 12 (1972), pp. 99–111.
[21] ———, *Perturbation theory for pseudoinverses*, BIT, 13 (1973), pp. 217–232.
[22] M. WEI, *Total least squares and least squares problems with more than one solution*, Numer. Math., 61 (1992), pp. 123–148.
[23] M. D. ZOLTOWSKI, *Generalized minimum norm and constrained total least squares with applications to signal processing*, in Advanced Algorithms and Architectures for Signal Processing III, SPIE, 975 (1988), pp. 78–85.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1182–1207, October 1994

010

RANK ROBUSTNESS OF COMPLEX MATRICES WITH RESPECT TO REAL PERTURBATIONS*

M. A. WICKS† AND R. A. DECARLO‡

Abstract. This paper examines the problem of computing a minimum norm real matrix perturbation that causes a general complex (system) matrix to drop rank. Given the state model describing a linear time-invariant system, the norm of this matrix perturbation helps to determine the robustness of several system properties with respect to real parameter variations. The norm of this perturbation, or the real-restricted singular value of the complex matrix, is known to be a discontinuous function of the complex matrix. The paper presents a simple condition on the complex matrix that eliminates this discontinuity. Specifically, the paper shows that the size of the smallest real rank-reducing perturbation is a continuous function of the complex matrix as long as the imaginary part of the complex matrix has full rank. The paper examines other aspects of the continuity of the problem. It also presents an algorithm that converges to a point satisfying a necessary condition for obtaining the smallest real rank-reducing matrix perturbation. A Lyapunov function approach is used to establish convergence of the algorithm. Some numerical examples are included illustrating the accuracy of the approach.

Key words. matrix perturbation theory, stability robustness, controllability robustness, rank, rounding errors, singular value decomposition

AMS subject classifications. 15A03, 65F35, 65G05, 93B35, 93B20, 93D99

1. Introduction.

The motivation for this paper arises from the study of the robustness of certain properties of finite dimensional linear time-invariant systems with respect to structured parameter variations. Many system properties are characterized by the rank of a family of matrices derived from the property being investigated and the system matrices identified with the state model. For example, let the system matrices, A, B, C, and D determine a linear time-invariant state model,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times r}$. A system is uncontrollable if and only if for some $\lambda \in \mathbb{C}$, $\operatorname{rank}[A - \lambda I, \quad B] < n$. Similarly, a system fails to be asymptotically stable if and only if for some $\lambda \in \mathbb{C}$ having $\operatorname{Re}[\lambda] \geq 0$, $\operatorname{rank}[A - \lambda I] < n$ (see, for example, [11]). For convenience, "unstable" means "not asymptotically stable."

As opposed to the problem of simply classifying whether a particular system property holds or fails to hold for some nominal system, robustness problems attempt to measure the magnitude of parameter variations which may result in some desirable system property failing to hold. The motivation for such problems is twofold: first, ensuring that a particular property will continue to hold in the face of anticipated

*Received by the editors January 21, 1991; accepted for publication (in revised form) June 24, 1993.

†Electrical and Computer Engineering Department, GMI Engineering and Management Institute, 1700 West Third Avenue, Flint, Michigan 48504-4898 (mwicks@nova.gmi.edu). This research was supported in part by David Ross grant 690-1285-1378 while the author was affiliated with Purdue University.

‡School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907-1285.

parameter variations and, second, ensuring that some property holds for a system for which only noisy or truncated parameter values are available (e.g., those obtained from measurement or numerical calculation). As an example, consider the controllability robustness problem: the magnitude of the minimum norm matrix perturbation $[\delta A, \quad \delta B]$, for which the perturbed pair $[A - \delta A, \quad B - \delta B]$ is uncontrollable, is given by

$$\min_{\lambda \in \mathbb{C}} \sigma_n [A - \lambda I, \quad B] \tag{1.2}$$

in [4]. This problem is equivalent to finding the minimum norm matrix perturbation $[\delta A, \quad \delta B]$ which causes a system matrix $[A - \delta A - \lambda I, \quad B - \delta B]$ to be rank deficient for some $\lambda \in \mathbb{C}$. The actual rank-reducing perturbation matrices δA and δB associated with the minimum of (1.2) are contained in $\mathbb{C}^{n \times n}$ and $\mathbb{C}^{n \times r}$, respectively, i.e., these minimum norm perturbations may have nonzero imaginary parts. The problem of performing the minimization of (1.2) is discussed in [3], [4], and [10]. Similarly the magnitude of the matrix perturbation δA having smallest norm for which the matrix $A - \delta A$ is unstable (given that A is asymptotically stable) is given by

$$\min_{\omega \in \mathbb{R}} \sigma_n [A - j\omega I] \tag{1.3}$$

in [9]. This problem is also equivalent to finding the minimum norm perturbation matrix $[\delta A]$ that causes a system matrix, in this case $[A - \delta A - j\omega I]$, to lose rank for some $\omega \in \mathbb{R}$. Again the destabilizing perturbation having smallest norm δA will lie in $\mathbb{C}^{n \times n}$ and may have a nonzero imaginary part. A bisection method for performing the minimization of (1.3) was developed in [2]. A unified treatment of several similar problems is presented in [6].

A criticism of the above measures is that they are pessimistic; see for example, [3], [6], and [9]. The smallest rank-reducing perturbation obtained from the implied minimizations may be complex, although the physical parameter variations can be only real. More generally, arbitrary parameter variations may not be possible; parameter variations may be known to lie in an even more restrictive set due to physical constraints. Performing minimizations over these more restrictive sets results in larger measures of robustness [8].

Toward the goal of computing these "restricted" robustness measures, this paper investigates the problem of determining real-restricted singular values of complex matrices. This problem is distinct from the problem of finding a minimum norm real perturbation that makes a system uncontrollable or unstable. The smallest singular value of a matrix M provides the norm of the smallest matrix δM for which $M - \delta M$ fails to have full rank. In a similar fashion, one can define an $\mathcal{S}$-restricted singular value of M as the norm of the smallest matrix $\delta M \in \mathcal{S}$ (if there is one) for which $M - \delta M$ fails to have full rank. In particular, when $M \in \mathbb{C}^{n \times m}$, the real-restricted singular value of M, $\sigma_{\min}(M; \mathbb{R}^{n \times m})$ is the size of the smallest real matrix δM for which $M - \delta M$ fails to have full rank, i.e., $\mathcal{S} = \mathbb{R}^{n \times m}$. This paper considers the computation of $\sigma_{\min}(M; \mathbb{R}^{n \times m})$.

2. Problem formulation. The following notation proves to be useful throughout this paper:

$\|M\|$, the Frobenius norm of the matrix M;
$\|v\|$, the usual Euclidean norm of the vector v;
$\bar{a}$, the complex conjugate of the scalar or vector a;

M^H, the conjugate of the transpose of the matrix M;
M^T, the transpose of the matrix M (not conjugated);
$\sigma_n[M]$, the nth singular value of the matrix, M;
$\mathrm{Re}[x]$, the real part of x;
$\mathrm{Im}[x]$, the imaginary part of x;
$\ker[M]$, the kernel of the map from $\mathbb{C}^m \to \mathbb{C}^n$, determined by the matrix M;
$\mathrm{range}[M]$, the range of the map from $\mathbb{C}^m \to C^n$, determined by the matrix M;
M^+, the Moore–Penrose pseudoinverse of the matrix M.

Throughout this paper, M is always used to denote a matrix $M \in \mathbb{C}^{n\times m}$ with $m \geq n$ implied. The spaces $\mathbb{R}^{n\times m}$ and $\mathbb{C}^{n\times m}$ may be viewed as vector spaces with the usual topology. Often it is helpful to view $\mathbb{C}^{n\times m}$ as a $2nm$ dimensional linear space over the field of real numbers. This is indicated where necessary.

The measure $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ is given by

DEFINITION 2.1. *Let $M \in \mathbb{C}^{n\times m}$ with $n \leq m$ and let*

$$\mathcal{S}(M; \mathbb{R}^{n\times m}) = \{\delta M \in \mathbb{R}^{n\times m} : \mathrm{rank}[M - \delta M] < n\}, \tag{2.1}$$

i.e., if M has full rank, $\mathcal{S}(M;\mathbb{R}^{n\times m})$ is the set of rank-reducing perturbations for M. Define

$$\sigma_{\min}(M; \mathbb{R}^{n\times m}) = \min_{\delta M \in \mathcal{S}(M;\, \mathbb{R}^{n\times m})} \|\delta M\|. \tag{2.2}$$

Some elementary properties of the set $\mathcal{S}(M; \mathbb{R}^{n\times m})$ demonstrate that the problem of finding a minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ is well posed. The following lemma characterizes the first of these properties.

LEMMA 2.2. *Let u be a nonzero element of $\mathbb{C}^n$, $n \geq 2$ such that and $\mathrm{Re}[u]$ and $\mathrm{Im}[u]$ are linearly independent. For each $M \in \mathbb{C}^{n\times m}$, there exists a perturbation matrix $\delta M \in \mathbb{R}^{n\times m}$ such that $u^H(M - \delta M) = 0$, i.e., δM is an element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. Moreover, the smallest such δM is given by*

$$\delta M = \begin{bmatrix} \mathrm{Re}[u^H] \\ \mathrm{Im}[u^H] \end{bmatrix}^+ \begin{bmatrix} \mathrm{Re}[u^H M] \\ \mathrm{Im}[u^H M] \end{bmatrix}. \tag{2.3}$$

Proof. With δM as in (2.3), we show that $u^H \delta M = u^H M$. The pseudoinverse on the right side of (2.3) is actually a right inverse since $\mathrm{Re}[u]$ and $\mathrm{Im}[u]$ are linearly independent. Consider then,

$$\begin{aligned} u^H \delta M &= (\mathrm{Re}[u^H] + j\,\mathrm{Im}[u^H]) \begin{bmatrix} \mathrm{Re}[u^H] \\ \mathrm{Im}[u^H] \end{bmatrix}^+ \begin{bmatrix} \mathrm{Re}[u^H M] \\ \mathrm{Im}[u^H M] \end{bmatrix} \\ &= [\,1 \quad j\,] \begin{bmatrix} \mathrm{Re}[u^H M] \\ \mathrm{Im}[u^H M] \end{bmatrix} = u^H M \end{aligned}$$

as was to be shown. Being the solution to a least squares problem, this is the smallest such δM. □

Equation (2.3) of Lemma 2.2 provides a partial parametrization of some elements of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. These elements of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ are parametrized by the complex vector u. Proposition 3.3, which appears later in the paper, demonstrates that the parameterization of (2.3) includes a minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ if the imaginary part of M has full rank. Hence, it is possible to compute a minimum norm real element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ by minimizing the norm of δM given by (2.3)

over $u \in \mathbb{C}^n$. However, a direct minimization of the Frobenius matrix norm using a standard minimization program poses a numerical stability problem halving the accuracy of the minimizing matrix. To see the difficulty, consider the minimization

$$\min_a \left\| \begin{pmatrix} 1 \\ a \end{pmatrix} \right\|.$$

The norm of the global minimum of this problem is numerically indistinguishable from the norm of the vector $[1, \sqrt{\epsilon}]$ since float $\|[1, \sqrt{\epsilon}]\| =$ float$\|[1, 0]\|$, where ϵ is any positive number less than the machine constant. Numerically speaking, these two vectors have the same norm. However, the two vectors are numerically distinguishable. This difficulty is avoided by requiring a solution to satisfy an orthogonality condition rather than directly minimizing a norm. In the context of the previous example, the minimizing vector must be numerically orthogonal (meaning that a numerical evaluation of the inner product yields approximately zero) to the vector $[0, 1]$. The vector $[1, \epsilon]$, which is a better solution than $[1, \sqrt{\epsilon}]$, is numerically orthogonal to $[0, 1]$.

LEMMA 2.3. *For each $M \in \mathbb{C}^{n\times m}$, the set $\mathcal{S}(M; \mathbb{R}^{n\times m})$ is closed.*

Proof. Let $\delta M_k \in \mathcal{S}(M; \mathbb{R}^{n\times m})$ be a sequence converging to δM_*. The proof is completed by showing that $\delta M_* \in \mathcal{S}(M; \mathbb{R}^{n\times m})$. For each δM_k, there exists a u_k satisfying $\|u_k^H\| = 1$ and

$$u_k^H(M - \delta M_k) = 0. \tag{2.4}$$

Because each u_k has unit length, there is a convergent subsequence $\{u_{k_n}\} \to u_*$ with the property that $\|u_*\| = 1$ and $u_*^H(M - \delta M_*) = 0$, by the continuity of (2.4). Hence δM_* must lie in $\mathcal{S}(M; \mathbb{R}^{n\times m})$. Again, if M has full rank, then δM_* is a real rank-reducing perturbation for M. □

The quantity $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ is well defined as per the following.

PROPOSITION 2.4. *For each $M \in \mathbb{C}^{n\times m}$ with $n \geq 2$, there exists a $\delta M_* \in \mathcal{S}(M; \mathbb{R}^{n\times m})$ such that*

$$\sigma_{\min}(M; \mathbb{R}^{n\times m}) = \|\delta M_*\|.$$

Proof. From Lemmas 2.2 and 2.3, the set $\mathcal{S}(M; \mathbb{R}^{n\times m})$ is closed and nonempty. It is nonempty because the assumption $n \geq 2$ allows an arbitrary choice for u having real and imaginary parts as in Lemma 2.2. Let $\delta M_0 \in \mathcal{S}(M; \mathbb{R}^{n\times m})$ and consider the set

$$\mathcal{T} = \mathcal{S}(M; \mathbb{R}^{n\times m}) \cap \{\delta M \in \mathbb{R}^{n\times m}: \|\delta M\| \leq \|\delta M_0\|\}.$$

This set $\mathcal{T}$ is nonempty and compact. Hence, there is a $\delta M_* \in \mathcal{T}$ such that $\|\delta M_*\| = \min_{\delta M\in\mathcal{T}} \|\delta M\|$. Also, $\delta M_* \in \mathcal{T} \subset \mathcal{S}(M; \mathbb{R}^{n\times m})$. Clearly δM_* is a minimum norm matrix of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. □

Since the problem of finding a real minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ is well posed for $n \geq 2$, it is possible to take up the question of computing δM_*.

Before developing an algorithm for computing the quantity $\sigma_{\min}(M; \mathbb{R}^{n\times m})$, it is useful to study the continuity of this measure. The next section examines the continuity of $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ with respect to M.

3. The question of continuity. The behavior of the measure $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ provides insight into its computation. For example, problems arise when examining the continuity of $\sigma_{\min}(M; \mathbb{R}^{n\times m})$. While studying the real-restricted controllability

robustness problem, Eising [3] encountered a discontinuity when performing a minimization equivalent to

$$\min_{\lambda \in \mathbb{C}} \sigma_n([A - \lambda I, \quad B]; \mathbb{R}^{n \times m}). \tag{3.1}$$

The root of this problem is indicated by the following proposition.

PROPOSITION 3.1. *The function* $\sigma_{\min}(M; \mathbb{R}^{n \times m})$ *is not continuous on* $\mathbb{C}^{n \times m}$.

The following example verifies this proposition.

Example 3.1. Let $\epsilon > 0$ and

$$M(\epsilon) = \begin{bmatrix} 1 & 0 \\ 0 & j\epsilon \end{bmatrix}.$$

The determinant of $M(\epsilon) - \delta M$ is given by

$$\det[M(\epsilon) - \delta M] = (\delta m_{11}\, \delta m_{22} - \delta m_{22} + \delta m_{21}\, \delta m_{12}) + j\epsilon(1 - \delta m_{11}).$$

Clearly, $\det[M(\epsilon) - \delta M] = 0$ implies that $\delta m_{11} = 1$ when $\epsilon > 0$. Any real rank-reducing perturbation δM satisfies $\|\delta M\| \geq 1$. Moreover, $\sigma_{\min}(M(\epsilon); \mathbb{R}^{n \times m}) = 1$ for every $\epsilon > 0$, but $\sigma_{\min}(M(0); \mathbb{R}^{n \times m}) = 0$. Thus

$$\lim_{\epsilon \to 0^+} \sigma_{\min}(M(\epsilon); \mathbb{R}^{n \times m}) \neq \sigma_{\min}(M(0); \mathbb{R}^{n \times m}),$$

which illustrates the discontinuity. This example is easily extended to larger dimensions.

The discontinuity implies that the problem of determining $\sigma_{\min}(M; \mathbb{R}^{n \times m})$ is potentially ill conditioned. More specifically, there is no guarantee that given any $\epsilon > 0$, there is some $\delta > 0$ and a "small" K such that

$$|\sigma_{\min}(M + \delta M; \mathbb{R}^{n \times m}) - \sigma_{\min}(M; \mathbb{R}^{n \times m})| < K\|\delta M\|$$

whenever $\|\delta M\| < \delta$.

Remarkably, however, the minimization of (3.1) is continuous and well conditioned for real matrices A and B even though the embedded problem of computing $\sigma_{\min}(M; \mathbb{R}^{n \times m})$ is neither continuous nor well conditioned. For example, the minimization of (3.1) can be effectively computed without explicitly determining $\sigma_{\min}(M; \mathbb{R}^{n \times m})$. See [10] for details.

The first goal of this section is to characterize the nature of these discontinuities. Continuity depends on the rank of the perturbation matrix δM_* that minimizes (2.2), or equivalently on the space spanned by the real and imaginary parts of the left null-vector of $M - \delta M_*$. The next two propositions characterize properties needed before studying continuity.

PROPOSITION 3.2. *Let* $M \in \mathbb{C}^{n \times m}$. *If* δM_* *is a minimum norm element of* $\mathcal{S}(M; \mathbb{R}^{n \times m})$, *then* $\operatorname{rank}[\delta M_*] \leq 2$.

Proof. Clearly there exists a unit length $u \in \mathbb{C}^n$ such that $u^H(M - \delta M_*) = 0$. Consider the minimization problem,

$$\min_{\delta M \in \mathbb{R}^{n \times m}} \left\| \begin{bmatrix} \operatorname{Re}[u^H] \\ \operatorname{Im}[u^H] \end{bmatrix} \delta M - \begin{bmatrix} \operatorname{Re}[u^H M] \\ \operatorname{Im}[u^H M] \end{bmatrix} \right\| = 0. \tag{3.2}$$

The minimum must be zero since letting $\delta M = \delta M_*$ in (3.2) yields zero. This is a linear least squares problem, having as its unique minimum norm solution the matrix

$$\delta M_0 = \begin{bmatrix} \mathrm{Re}[u^H] \\ \mathrm{Im}[u^H] \end{bmatrix}^+ \begin{bmatrix} \mathrm{Re}[u^H M] \\ \mathrm{Im}[u^H M] \end{bmatrix}.$$

The matrix δM_0 is an element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ and must have norm greater than or equal to that of δM_*. However, δM_0 is also the minimum norm solution to problem (3.2) and must have norm less than or equal to that of δM_*, since δM_* is also a solution to (3.2). Thus, $\|\delta M_0\| = \|\delta M_*\|$. From the uniqueness of the minimum norm solution to (3.2), $\delta M_0 = \delta M_*$. The matrix δM_* clearly has rank zero, one, or two. □

PROPOSITION 3.3. *Let $M \in \mathbb{C}^{n\times m}$ for which $\mathrm{Im}[M]$ has full rank and $n \geq 2$. Let u be any nonzero vector such that $u^H(M - \delta M_0) = 0$, where δM_0 is any element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. The vectors, $\mathrm{Re}[u]$ and $\mathrm{Im}[u]$ are linearly independent.*

Proof. Let δM_0 be any element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. Suppose there is a nonzero vector u satisfying $u^H(M - \delta M_0) = 0$, but for which $\mathrm{Re}[u]$ and $\mathrm{Im}[u]$ are linearly dependent. As such, there exists a constant, $c \in \mathbb{C}$ of modulus one, such that $\mathrm{Im}[cu] = 0$. Let $\hat{u} = cu$. The vector $\hat{u}$ satisfies $\hat{u}^H(M - \delta M_0) = 0$ implying

$$\mathrm{Im}[\hat{u}^H(M - \delta M_0)] = \hat{u}^H \mathrm{Im}[M] = 0,$$

which would imply $\mathrm{Im}[M]$ does not have full rank. □

Upper and lower bounds for $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ are useful. Proposition 3.5 that follows, provides these bounds. The development of this upper bound requires the following lemma.

LEMMA 3.4. *Let $u_1, u_2 \in \mathbb{C}^n$ be linearly independent. There exists a nonzero vector, $x \in \mathrm{span}\{u_1, u_2\}$, such that $\|\mathrm{Re}[x]\| = \|\mathrm{Im}[x]\|$ and $(\mathrm{Re}[x])^T(\mathrm{Im}[x]) = 0$.*

Proof. Note that for $x \in \mathbb{C}^n$, $\|\mathrm{Re}[x]\| = \|\mathrm{Im}[x]\|$ and $(\mathrm{Re}[x])^T(\mathrm{Im}[x]) = 0$ if and only if $x^T x = 0$. Given linearly independent $u_1, u_2 \in \mathbb{C}^{n\times m}$, suppose $u_1^T u_1 \neq 0$; otherwise the result follows trivially. For some as yet unspecified $c \in \mathbb{C}$, consider the quantity,

$$(cu_1 + u_2)^T(cu_1 + u_2) = (u_1^T u_1)c^2 + 2(u_1^T u_2)c + (u_2^T u_2).$$

This quadratic has a zero for some $c \in \mathbb{C}$ since $u_1^T u_1$ is nonzero. Let $x = cu_1 + u_2$. This x has the stated properties, which completes the proof. □

PROPOSITION 3.5. *It holds that $\sigma_n[M] \leq \sigma_{\min}(M; \mathbb{R}^{n\times m}) \leq \sqrt{2}\sigma_{n-1}[M]$.*

Proof. The lower bound given is obvious. To establish the upper bound, let u_{n-1} be a unit length left singular vector associated with $\sigma_{n-1}[M]$ and let u_n be a unit length left singular vector associated with $\sigma_n[M]$. From Lemma 3.4 there exist constants $c_1, c_2 \in \mathbb{C}$ for which $x = c_1 u_{n-1} + c_2 u_n$ satisfies $\|\mathrm{Re}[x]\| = \|\mathrm{Im}[x]\| = 1$ and $(\mathrm{Re}[x])^T(\mathrm{Im}[x]) = 0$. Consider the matrix,

$$\delta M \triangleq [\,\mathrm{Re}[x] \quad -\mathrm{Im}[x]\,] \begin{bmatrix} \mathrm{Re}[x^H M] \\ \mathrm{Im}[x^H M] \end{bmatrix}.$$

This matrix δM is an element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ since $x^H \delta M = x^H M$. Also

$$\|\delta M\| = \left\| \begin{bmatrix} \mathrm{Re}[x^H M] \\ \mathrm{Im}[x^H M] \end{bmatrix} \right\| = \|x^H M\|$$

from the orthogonality of the real and imaginary parts of x. It follows that

$$\|x^H M\|^2 = |c_1|^2\sigma_{n-1}^2[M] + |c_2|^2\sigma_n^2[M] \leq 2\sigma_{n-1}^2[M]$$

since $|c_1|^2 + |c_2|^2 = 2$. It follows that $\sigma_{\min}(M; \mathbb{R}^{n\times m}) \leq \sqrt{2}\sigma_{n-1}[M]$. This completes the proof of Proposition 3.5. □

The next two results, Lemma 3.6 and Proposition 3.7 establish that a nonzero minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$, say δM_* satisfies $\text{rank}[M - \delta M_*] = n - 1$, i.e., the $n-$ 1st singular value of $M - \delta M_*$ is strictly positive, when $\text{Im}[M]$ has full rank.

LEMMA 3.6. *Let $M \in \mathbb{C}^{n\times m}$ for which $\text{Im}[M]$ has full rank and $n \geq 2$. Let δM be any element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. For any two nonzero vectors in the left kernel of $M - \delta M$, say u_1 and u_2,*

$$\text{span}\,[\,\text{Re}[u_1],\quad \text{Im}[u_1]\,] \cap \text{span}\,[\,\text{Re}[u_2],\quad \text{Im}[u_2]\,] \neq \{0\}$$

if and only if $u_1 = cu_2$ for some $c \in \mathbb{C}$.

Proof. Let δM be an element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ and let u_1 and u_2 be nonzero vectors in the left kernel of $M - \delta M$. Suppose there are nonzero real vectors x and y such that

$$[\,\text{Re}[u_1],\quad \text{Im}[u_1]\,]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [\,\text{Re}[u_2],\quad \text{Im}[u_2]\,]\begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

i.e., $\text{Re}[u_1(x_1 - jx_2) - u_2(y_1 - jy_2)] = 0$. Proposition 3.3 implies that $u_1(x_1 - jx_2) - u_2(y_1 - jy_2)$ must be zero since its real part is zero and it is obviously in the left kernel of $M - \delta M$. Hence $u_1 = cu_2$ for some c, completing the lemma (the converse is obvious). □

PROPOSITION 3.7. *Let $M \in \mathbb{C}^{n\times m}$ for which $\text{Im}[M]$ has full rank and $n \geq 2$ and let δM_* be a nonzero minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$, then* $\text{rank}[M - \delta M_*] = n - 1$.

Proof. Let u_1 and u_2 be any two nonzero vectors in the left kernel of $M - \delta M_*$. By the construction used in Lemma 2.2, specifically (2.3), it follows that

$$\text{range}[\delta M_*] \subset \text{span}\,[\,\text{Re}[u_1],\quad \text{Im}[u_1]\,]$$

and

$$\text{range}[\delta M_*] \subset \text{span}\,[\,\text{Re}[u_2],\quad \text{Im}[u_2]\,].$$

By Lemma 3.6, $u_1 = cu_2$. The vectors u_1 and u_2 cannot be linearly independent. These vectors were arbitrary in the kernel of $M - \delta M_*$; hence, the result follows. □

The points of discontinuity of the restricted singular values have a special structure; given $M \in \mathbb{C}^{n\times m}$ and any sequence, $\{M_k\} \to M$, the value of $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ is always less than or equal to the limit of the sequence $\{\sigma_{\min}(M_k; \mathbb{R}^{n\times m})\}$. More specifically, the nature of these points is characterized by the following lemma.

LEMMA 3.8. *If $\{M_k\} \in \mathbb{C}^{n\times m}, n \geq 2$ is any sequence converging to M, then*

$$\sigma_{\min}(M; \mathbb{R}^{n\times m}) \leq \liminf\{\sigma_{\min}(M_k; \mathbb{R}^{n\times m})\}.$$

Proof. For each M_k there is a unit length vector u_k and minimum norm element of $\mathcal{S}(M_k; \mathbb{R}^{n\times m})$, say δM_k, satisfying

$$u_k^H(M_k - \delta M_k) = 0 \tag{3.3}$$

by Proposition 2.4. Also,

$$\sigma_{\min}(M_k; \mathbb{R}^{n\times m}) = \|\delta M_k\|. \tag{3.4}$$

Let $l = \liminf\{\sigma_{\min}(M_k; \mathbb{R}^{n\times m})\}$. The sequence, $\{\sigma_{\min}(M_k; \mathbb{R}^{n\times m})\}$ is bounded, by Proposition 3.5 and the convergence of $\{M_k\}$. Therefore, the value l is finite. The sequence $\{u_k\}$ is bounded as is the sequence $\{\delta M_k\}$. Thus a subsequence indexing k_n exists such that $\{\sigma_{\min}(M_{k_n}; \mathbb{R}^{n\times m})\} \to l$, $\{u_{k_n}\} \to u$, and $\{\delta M_{k_n}\} \to \delta M$ for some u and δM. The limits u and δM are assumed to have no special relationship yet. From the continuity of (3.3), however,

$$u^H(M - \delta M) = 0.$$

This shows that δM is an element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ implying

$$\sigma_{\min}(M; \mathbb{R}^{n\times m}) \le \|\delta M\|.$$

Taking the limit of (3.4) (with respect to the subsequence indexing), however, implies $\|\delta M\| = l$, as was to be shown. □

PROPOSITION 3.9. *Let $M \in \mathbb{C}^{n\times m}, n \ge 2$ be such that $\mathrm{Im}[M]$ has full rank, then $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ is continuous at M in $\mathbb{C}^{n\times m}$.*

Proof. Let $\{M_k\}$ be any sequence that converges to M. There exist a nonzero vector u and a minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ say δM_* that satisfy $u^H(M - \delta M_*) = 0$. From Proposition 3.3, the vectors $\mathrm{Re}[u]$ and $\mathrm{Im}[u]$ are linearly independent. The matrix

$$\delta M_k \triangleq \begin{bmatrix} \mathrm{Re}[u^H] \\ \mathrm{Im}[u^H] \end{bmatrix}^+ \begin{bmatrix} \mathrm{Re}[u^H M_k] \\ \mathrm{Im}[u^H M_k] \end{bmatrix}$$

is a continuous function of M_k and is an element of $\mathcal{S}(M_k; \mathbb{R}^{n\times m})$ (Lemma 2.2). Also, as per the proof of Proposition 3.2,

$$\delta M_* = \begin{bmatrix} \mathrm{Re}[u^H] \\ \mathrm{Im}[u^H] \end{bmatrix}^+ \begin{bmatrix} \mathrm{Re}[u^H M] \\ \mathrm{Im}[u^H M] \end{bmatrix}.$$

Hence it follows that

$$\lim\{\delta M_k\} = \delta M_*$$

and

$$\lim\{\|\delta M_k\|\} = \sigma_{\min}(M; \mathbb{R}^{n\times m}).$$

Clearly, however,

$$\|\delta M_k\| \ge \sigma_{\min}(M_k; \mathbb{R}^{n\times m}),$$

since δM_k is an element of $\mathcal{S}(M_k; \mathbb{R}^{n\times m})$ but not necessarily having minimum norm. Therefore,

$$\lim\{\|\delta M_k\|\} \ge \limsup\{\sigma_{\min}(M_k; \mathbb{R}^{n\times m})\}.$$

Combining the above yields

$$\sigma_{\min}(M; \mathbb{R}^{n\times m}) \ge \limsup\{\sigma_{\min}(M_k; \mathbb{R}^{n\times m})\}.$$

Combining this result with that of Lemma 3.8 yields

$$\lim\{\sigma_{\min}(M_k; \mathbb{R}^{n\times m})\} = \sigma_{\min}(M; \mathbb{R}^{n\times m}).$$

This establishes the continuity of $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ at M which was to be shown. □

Proposition 3.9 implies that points of discontinuity of $\sigma_{\min}(M; \mathbb{R}^{n\times m})$ will be avoided if $\mathrm{Im}[M]$ is known to have full rank. This full rank condition is generally the case for the minimizations of (1.2) and (1.3). As long as $\sigma_n[\mathrm{Im}[M]]$ is greater than the roundoff or measurement uncertainty of M, Proposition 3.9 ensures that M is a safe distance from a point of discontinuity of $\sigma_{\min}(M; \mathbb{R}^{n\times m})$.

The next goal of this paper is to develop necessary conditions that a minimizer of (2.2) must satisfy. The next section addresses this question.

4. Development of necessary condition. The following lemma provides a necessary condition for a particular perturbation matrix to be a minimizer of (2.2), when $\mathrm{Im}[M]$ has full rank.

LEMMA 4.1. *Let $M \in \mathbb{C}^{n\times m}$ be a matrix for which $\mathrm{Im}[M]$ has full rank and $n \geq 2$. Let δM_0 be any element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. If there is a real matrix $\delta\tilde{M}_0$ such that* (i) $\|\delta\tilde{M}_0\| < \|\delta M_0\|$ *and* (ii) *some nontrivial $u_0 \in \ker[(M - \delta M_0)^H]$ and every $v \in \ker[M - \delta M_0]$ satisfy $u_0^H(M - \delta\tilde{M}_0)v = 0$ (or equivalently $u_0^H(\delta M_0 - \delta\tilde{M}_0)v = 0$), then there exists a δM_1 in $\mathcal{S}(M; \mathbb{R}^{n\times m})$ such that $\|\delta M_1\| < \|\delta M_0\|$.*

Proof. Let $\delta M_0 \in \mathcal{S}(M; \mathbb{R}^{n\times m})$ and let $\delta\tilde{M}_0$ be any matrix satisfying (i) and (ii) above. Assume $M - \delta\tilde{M}_0$ has full rank, otherwise $\delta\tilde{M}_0$ is an element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$ having norm smaller than the norm of δM_0, in which case the result follows immediately. Let $u_0 \in \ker[(M - \delta M_0)^H]$ be arbitrary and let $x_0^H = u_0^H(\delta\tilde{M}_0 - M)$ or equivalently $x_0^H = u_0^H(\delta\tilde{M}_0 - \delta M_0)$. From condition (ii) x_0 is orthogonal to every $v \in \ker(M - \delta M_0)$. It follows that $x_0 \in \mathrm{range}[(M - \delta M_0)^H]$. Hence, there exists a $\tilde{u}_0 \in \mathrm{range}[M - \delta M_0]$ such that $\tilde{u}_0^H(M - \delta M_0) = x_0^H$. The remainder of the proof constructs a real rank-reducing perturbation smaller than δM_0 having a linear combination of u_0 and $\tilde{u}_0$, as a left null vector.

Consider the sequence obtained from the product,

$$[u_0 + \gamma\tilde{u}_0]^H[(M - \delta M_0) - \gamma(\delta\tilde{M}_0 - \delta M_0)] = -\gamma^2\tilde{u}_0^H(\delta\tilde{M}_0 - \delta M_0). \tag{4.1}$$

For γ sufficiently small, a real matrix perturbation can be used to cancel this vector. Consider the matrix,

$$\delta M_\gamma \triangleq \gamma^2 \begin{bmatrix} \mathrm{Re}[u_0^H + \gamma\tilde{u}_0^H] \\ \mathrm{Im}[u_0^H + \gamma\tilde{u}_0^H] \end{bmatrix}^+ \begin{bmatrix} \mathrm{Re}[\tilde{u}_0^H(\delta\tilde{M}_0 - \delta M_0)] \\ \mathrm{Im}[\tilde{u}_0^H(\delta\tilde{M}_0 - \delta M_0)] \end{bmatrix}. \tag{4.2}$$

The pseudoinverse on the right side of (4.2) is actually a right-inverse for sufficiently small γ, say $\gamma < \gamma_0$, because $\mathrm{Re}[u_0]$ and $\mathrm{Im}[u_0]$ are linearly independent by Proposition 3.3. Hence, for $\gamma < \gamma_0$,

$$[u_0 + \gamma\tilde{u}_0]^H\delta M_\gamma = \gamma^2\tilde{u}_0^H(\delta\tilde{M}_0 - \delta M_0). \tag{4.3}$$

Thus, adding (4.3) to (4.1) yields

$$[u_0 + \gamma\tilde{u}_0]^H\left[M - [\gamma\delta\tilde{M}_0 + (1 - \gamma)\delta M_0 - \delta M_\gamma]\right] = 0,$$

i.e., $[\gamma\delta\tilde{M}_0 + (1 - \gamma)\delta M_0 - \delta M_\gamma]$ is a rank-reducing real perturbation for M. Clearly $\lim_{\gamma\to 0}(1/\gamma^2)\delta M_\gamma < \infty$ (examine (4.2)). For an appropriate constant b and $\gamma < \gamma_0$, it follows that $\|\delta M_\gamma\| \leq \gamma^2 b$. The norm of the matrix $[\gamma\delta\tilde{M}_0 + (1 - \gamma)\delta M_0 - \delta M_\gamma]$ can be made less than $\|\delta M_0\|$ by an appropriate choice for γ, since the norm of $\delta\tilde{M}_0$

is less than the norm of δM_0 and the norm of δM_γ vanishes quadratically. (One such choice is $\gamma < \max\{\frac{1}{b}[\|\delta M_0\| - \|\delta \tilde{M}_0\|], \gamma_0\}$.) This shows that for any k_0 satisfying

$$\gamma < \max\left\{\frac{1}{b}[\|\delta M_0\| - \|\delta \tilde{M}_0\|], \gamma_0\right\},$$

the matrix

$$\delta M_1 = [\gamma \delta \tilde{M}_0 + (1-\gamma)\delta M_0 - \delta M_\gamma]$$

is a real rank-reducing perturbation having norm less than $\|\delta M_0\|$. This completes the proof. □

The following necessary condition follows immediately from Lemma 4.1.

THEOREM 4.2. *Let $M \in \mathbb{C}^{n\times m}$ be a matrix for which* Im$[M]$ *has full rank and $n \geq 2$; let δM_* be a minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$; and let the matrix $\delta \hat{M}$ be the unique real matrix having smallest norm that satisfies*

$$u^H(\delta \hat{M})v = u^H M v \tag{4.4}$$

for some nonzero $u \in \ker[(M - \delta M_*)^H]$ and every $v \in \ker[(M - \delta M_*)]$. The matrices δM_* and $\delta \hat{M}$ must be equal.

Proof. Let δM_* be a minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. It is apparent that for some nonzero $u \in \ker[(M - \delta M_*)^H]$ and every $v \in \ker[M - \delta M_*]$, the matrix δM_* satisfies $u^H(M - \delta M_*)v = 0$. If it were not the smallest such matrix, then Lemma 4.1 would provide a smaller rank-reducing real perturbation matrix. But that is impossible since δM_* is a minimum norm real rank-reducing perturbation. □

Note that the dimension of the left kernel of $M - \delta M_*$ in Theorem 4.2 is one due to Proposition 3.7.

Theorem 4.2 provides a necessary condition for obtaining a candidate for a minimizer of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. The next section proposes algorithms that converge to a matrix satisfying this necessary condition.

5. Algorithm development. This section examines an algorithm for finding an element $\delta M \in \mathcal{S}(M; \mathbb{R}^{n\times m})$ that satisfies the necessary condition of Theorem 4.2. Given a rank-reducing perturbation δM, Theorem 4.2 provides a test to check whether δM satisfies a necessary condition for being a minimum norm element of $\mathcal{S}(M; \mathbb{R}^{n\times m})$. Moreover, given an initial approximation for the minimum norm, real matrix perturbation δM_k the solution of a least squares problem similar to (4.4) in Theorem 4.2 yields a matrix $\delta \hat{M}_k$, which can be used to find a better approximation to the desired solution. Furthermore, it can be shown that an improved approximation lies on the line connecting δM_k and $\delta \hat{M}_k$. The approximation is improved in the sense that the successive approximation decreases the value of a certain Lyapunov function at each iteration. Specifically, the Lyapunov function used to prove convergence of the algorithm is $P_k = [\sigma_n]_k + g_k\|\delta M_k\|$.

The Lyapunov function consists of two terms: the first term $[\sigma_n]_k$ measures closeness to $\mathcal{S}(M; \mathbb{R}^{n\times m})$ while the second measures the norm $\|\delta M_k\|$. The constant g_k is an adaptive weight parameter that is adjusted based on the condition of the problem. Establishing convergence of the Lyapunov function values to a constant and convergence of $[\sigma_n]_k$ to zero are the keys to proving convergence of the forthcoming algorithm.

Two assumptions will simplify the description of the algorithm and the discussion of its convergence. The assumptions are unnecessary for constructing a convergent algorithm; however, the clarity gained in the discussion warrants their introduction.

Assumption 1. The sequence $\{[\sigma_{n-1}]_k\}$ computed by the algorithm below is bounded away from zero.

Assumption 2. The sequence $\{g_k\}$ computed by the algorithm below is bounded away from zero, where g_k essentially measures the linear independence of the real and imaginary parts of the nth left singular vector of $M - \delta M_k$.

Proposition 3.7 motivates Assumption 1 because it demonstrates that a solution matrix δM_* satisfying the necessary condition cannot satisfy $\text{rank}[M - \delta M_*] < n - 1$. The algorithm below generates a sequence of real matrices δM_k having a subsequence converging to a matrix δM_*. So long as the sequence of singular values $[\sigma_{n-1}]_k$ of $M - \delta M_k$ is bounded away from zero the algorithm computes a matrix satisfying the necessary condition for a matrix in a neighborhood of the original M. It is possible to construct an algorithm that allows $[\sigma_{n-1}]_k$ to become zero at a finite number of points and converges to a matrix satisfying the necessary condition. However, dealing with this rare situation complicates the specification and discussion of the algorithm. For simplicity the algorithm is presented and analyzed under the assumption that $[\sigma_{n-1}]_k$ is bounded away from zero.

Proposition 3.3 motivates Assumption 2. Lemma A.2 in Appendix A shows that g_k is bounded below by the inverse of the spectral condition number of the matrix $[\,\text{Re}[u_n]_k,\ \text{Im}[u_n]_k\,]$. Proposition 3.3 ensures that this condition number remains bounded in a neighborhood of a solution point. Assumption 2 can be removed by changing the initial value for g_0 and the initial choice for δM_0 given in the algorithm. Proposition A.3 of Appendix A discusses removal of Assumption 2.

The quantities to be computed by the algorithm are illustrated graphically in Fig. 1. It is useful to consider the algorithm with reference to this figure.

ALGORITHM STATEMENT. Given a prespecified constant ϵ:

1. Let $k = 0$, let δM_0 be any real matrix satisfying $\text{rank}[M - \delta M_0] = n - 1$ and let $g_0 = 1$.[1]

2. Compute a singular value decomposition of $M - \delta M_k$, i.e., find orthonormal sets $\{[u_i]_k\}$, $i = 1, \ldots, n$ and $\{[v_i]_k\}$, $i = 1, \ldots, m$ so that

$$M - \delta M_k = \sum_{i=1}^{n} [u_i]_k [\sigma_i]_k [v_i]_k^H.$$

3. Define $V_k = [\,[v_n]_k \quad [v_{n+1}]_k \quad \ldots \quad [v_m]_k\,]$.

4. Compute $\delta \bar{M}_k$, a matrix direction designed to decrease the nth singular value. Let $\delta \bar{M}_k$ be the unique real matrix having smallest norm for which

$$[u_n]_k{}^H \delta \bar{M}_k V_k = [\,[\sigma_n]_k \quad 0 \quad \ldots \quad 0\,].$$

5. Compute $\delta \tilde{M}_k$, a matrix direction designed to decrease the norm of the real perturbation without changing the nth singular value. Let $\delta \tilde{M}_k$ be the unique real matrix having smallest norm for which

$$[u_n]_k{}^H \delta \tilde{M}_k V_k = [u_n]_k{}^H \delta M_k V_k.$$

---

[1]The less restrictive choice $\delta M_0 = [0]$ will work provided Assumptions 1 and 2 remain valid and provided that a left singular vector associated with the nth left singular value of M has linearly independent real and imaginary parts.

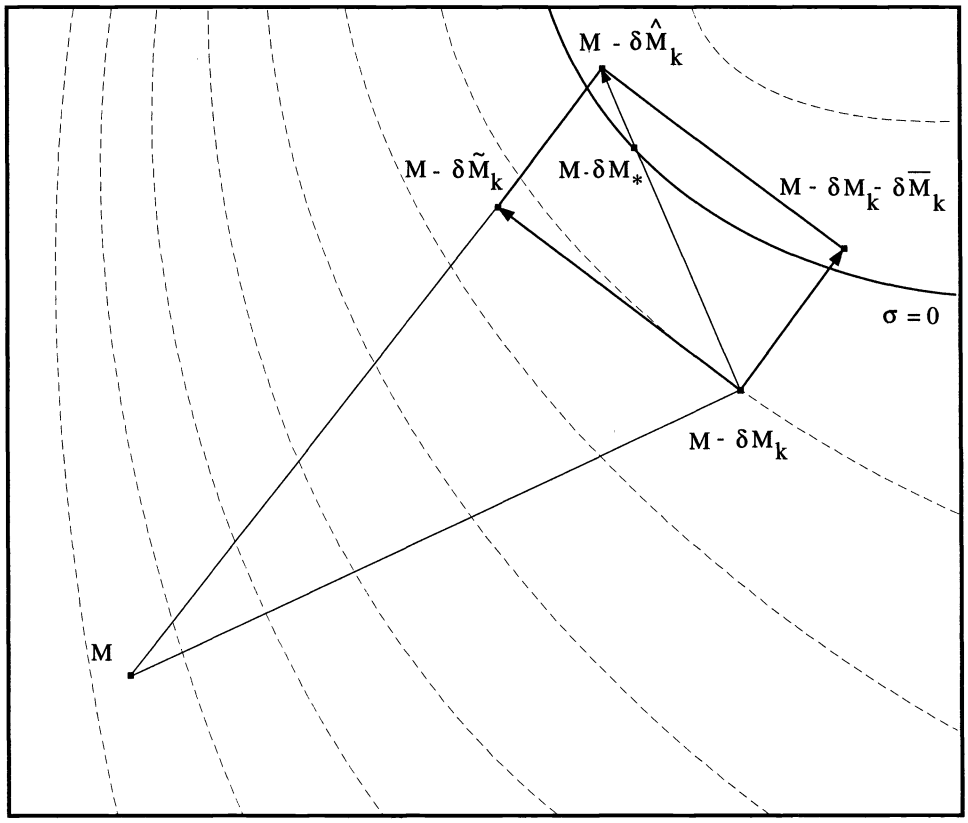


FIG. 1. *Geometric relationships among computed quantities and singular value level sets.*

(Lemma 2.2 guarantees the consistency of the equations and the existence of a solution in steps 4 and 5. The equation in step 5 comes from equating $[u_n]_k{}^H(M - \delta\tilde{M}_k)V_k$ to $[u_n]_k{}^H(M - \delta M_k)V_k$.

6. If $[\sigma_n]_k < \epsilon\|M - \delta M_k\|$ and $g_k\|\delta\tilde{M}_k - \delta M_k\| < \epsilon\|M - \delta M_k\|$ then stop; otherwise continue.

7. Define

$$g_k = \min\left(g_{k-1}, \frac{[\sigma_n]_k}{2\|\delta\bar{M}_k\|}\right)$$

when $[\sigma_n]_k \neq 0$; define $g_k = g_{k-1}$ otherwise.

8. Define

$$[\hat{u}]_k{}^H = \left[[u_n]_k^H(\delta\hat{M}_k - \delta M_k) - [\sigma_n]_k[v_n]_k^H\right](M - \delta M_k)^+,$$

where $\delta\hat{M}_k = \delta\tilde{M}_k + \delta\bar{M}_k$.

9. Define $\gamma_k = \min(\frac{1}{4}\frac{1}{q_k}([\sigma_n]_k + g_k b_k\|\delta\tilde{M}_k - \delta M_k\|), 1)$, where

$$q_k = \|\hat{u}_k^H(\delta\hat{M}_k - \delta M_k)\|$$

and

$$b_k = \begin{cases} \frac{\|\delta\tilde{M}_k - \delta M_k\|}{\|\delta M_k\|} & \text{if } \|\delta M_k\| \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

10. Set $\delta M_{k+1} = \gamma_k \delta M_k + (1 - \gamma_k)\delta\hat{M}_k$.
11. Let $k = k + 1$ and go to step 2.

A graphical illustration of the basic geometry involved in the algorithm appears in Fig. 1. In this figure, M is the given complex matrix. The dashed lines represent level sets of $\sigma_n(M - \delta M)$ in the space of real matrices δM. The solid curved line represents the level set corresponding to $\text{rank}[M - \delta M] = 0$. The matrix $M - \delta M_*$ is the desired solution point. The current approximation to $M - \delta M_*$, i.e., $M - \delta M_k$, appears at the lower right corner of the rectangle in the figure. Traveling along the right edge of the rectangle toward $M - \delta M_k - \delta\bar{M}_k$ decreases σ_n. This direction is normal to the level set at $M - \delta M_k$. The upper right corner lies near the zero level set. On the other hand, traveling from $M - \delta M_k$ toward the lower left corner decreases the norm of the perturbation with a small change in σ_n. This side of the rectangle is (i) tangent to the level set at $M - \delta M_k$, (ii) orthogonal to the line connecting M and $M - \delta\tilde{M}_k$, and (iii) orthogonal to the line connecting $M - \delta M_k$ and $M - \delta M_k - \delta\bar{M}_k$. Traveling along the diagonal of the rectangle, the approach taken here attempts to meet both objectives. Some point along the diagonal decreases the value of the Lyapunov function that represents a weighted combination of both objectives. This geometry proves useful in interpreting the theorems and proofs below.

The proof of the convergence of this algorithm requires four lemmas. It is helpful to view $\mathbb{R}^{n\times m}$ as a vector space equipped with the inner product, $< \cdot, \cdot >$, defined as follows for $M, N \in \mathbb{R}^{n\times m}$:

$$< M, N >= \sum_{i=1}^{n}\sum_{j=1}^{m} m_{ij} n_{ij}.$$

The norm induced by this inner product is the Frobenius norm.

In the algorithm, the matrix $\delta\hat{M}_k$ is the sum of two matrix components, $\delta\bar{M}_k$ and $\delta\tilde{M}_k$. Given δM_k, traveling in the matrix direction $\delta\bar{M}_k$ serves to decrease the size of $[\sigma_n]_k$ while the direction $\delta\tilde{M}_k - \delta M_k$ decreases the norm of δM_k. Traveling along the diagonal of the rectangle shown in Fig. 1 decreases the value of the Lyapunov function. Recall that $\delta\tilde{M}_k$ and $\delta\bar{M}_k$ are obtained from steps 4 and 5 of the algorithm as follows: $\delta\tilde{M}_k$ is a minimum norm real matrix satisfying

$$[u_n]_k{}^H \delta\tilde{M}_k V_k = [u_n]_k{}^H \delta M_k V_k,$$

while $\delta\bar{M}_k$ is a minimum norm real matrix satisfying

$$[u_n]_k{}^H \delta\bar{M}_k V_k = [\,[\sigma_n]_k \quad 0 \quad \dots \quad 0\,].$$

Moreover, $\delta\hat{M}_k = \delta\bar{M}_k + \delta\tilde{M}_k$.

Lemma 5.1, which follows, demonstrates the orthogonality of $\delta\tilde{M}_k - \delta M_k$ and $\delta\tilde{M}_k$ with respect to the inner product defined above.

LEMMA 5.1. *For any $\alpha, \beta \in \mathbb{R}$ the matrices $\delta\tilde{M}_k$ and $\delta\tilde{M}_k - \delta M_k$ are orthogonal. Hence, $\delta\tilde{M}_k$ and δM_k satisfy*

$$\|\alpha\delta\tilde{M}_k + \beta[\delta\tilde{M}_k - \delta M_k]\|^2 = |\alpha|^2\|\delta\tilde{M}_k\|^2 + |\beta|^2\|\delta\tilde{M}_k - \delta M_k\|^2.$$

Proof. The matrix $\delta\tilde{M}_k - \delta M_k$ satisfies

$$[u_n]_k{}^H(\delta\tilde{M}_k - \delta M_k)V_k = 0. \tag{5.1}$$

Since $\delta\tilde{M}_k$ is chosen to be a minimum norm solution in step 5 and since

$$[u_n]_k{}^H\left[\delta\tilde{M}_k + c(\delta\tilde{M}_k - \delta M_k)\right]V_k = [u_n]_k{}^H\delta M_k V_k$$

for all real c, it follows that $\|\delta\tilde{M}_k + c(\delta\tilde{M}_k - \delta M_k)\| > \|\delta\tilde{M}_k\|$ for all $c \neq 0$. This can only occur if $\delta\tilde{M}_k$ and $\delta\tilde{M}_k - \delta M_k$ are orthogonal with respect to the inner product on $\mathbb{R}^{n\times m}$. The result follows from the stated orthogonality. □

Lemma 5.2, which follows, provides an upper bound for the size of the solution of a particular matrix equation.

LEMMA 5.2. *Given $[v_i]_k$ for $i = 1,\ldots,m$, and $M - \delta M_k$ as defined in the algorithm, for any integer $r < n$ and any matrix, X, if $X^H[v_i]_k = 0$ for $i = r+1,\ldots,m$ and $[\sigma_r]_k \neq 0$, then there exists a matrix Y for which $Y^H(M - \delta M_k) = X^H$. Moreover, $\|Y\| \leq [\sigma_r]_k^{-1}\|X\|$.*

Proof. The result follows immediately from some elementary properties of singular spaces. □

Lemma 5.3 below provides a bound for the decrease in the norm of δM_k when moving along the path from δM_k toward $\delta\hat{M}_k$, which is the diagonal of the rectangle shown in Fig. 1.

LEMMA 5.3. *At each iteration,*

$$\|(1-\gamma)\delta M_k + \gamma\delta\hat{M}_k\| - \|\delta M_k\| \leq \gamma(\|\delta\bar{M}_k\| - \frac{1}{2}b_k\|\delta\tilde{M}_k - \delta M_k\|)$$

for $\gamma \in (0,1)$.

Proof. Assume $\|\delta M_k\| \neq 0$ otherwise the result follows trivially, since $\delta\hat{M}_k = \delta\bar{M}_k$ and $\delta\tilde{M}_k = 0$ in this case. Notice that

$$\begin{aligned}
\|(1-\gamma)\delta M_k + \gamma\delta\tilde{M}_k\|^2 &= \|(\gamma-1)(\delta\tilde{M}_k - \delta M_k) + \delta\tilde{M}_k\|^2 \\
&= (\gamma-1)^2\|\delta\tilde{M}_k - \delta M_k\|^2 + \|\delta\tilde{M}_k\|^2 \\
&= (\gamma)(\gamma-2)\|\delta\tilde{M}_k - \delta M_k\|^2 + (\|\delta\tilde{M}_k - \delta M_k\|^2 + \|\delta\tilde{M}_k\|^2) \\
&= (\gamma)(\gamma-2)\|\delta\tilde{M}_k - \delta M_k\|^2 + \|\delta M_k\|^2
\end{aligned}$$

follows from two applications of Lemma 5.1. In consequence,

$$\|(1-\gamma)\delta M_k + \gamma\delta\tilde{M}_k\|^2 - \|\delta M_k\|^2 = (\gamma)(\gamma-2)\|\delta\tilde{M}_k - \delta M_k\|^2$$

implying that

$$\|(1-\gamma)\delta M_k + \gamma\delta\tilde{M}_k\| - \|\delta M_k\| \leq -\frac{\gamma}{2}\frac{\|\delta\tilde{M}_k - \delta M_k\|^2}{\|\delta M_k\|} \tag{5.2}$$

for $\gamma \in (0,1)$. The previous inequality was obtained by dividing both sides by $\|(1-\gamma)\delta M_k + \gamma\delta\tilde{M}_k\| + \|\delta M_k\|$ and restricting $\gamma \in (0,1)$. A simple application of the triangle inequality results in

$$\|(1-\gamma)\delta M_k + \gamma\delta\hat{M}_k\| - \|\delta M_k\| \leq -\frac{\gamma}{2}\frac{\|\delta\tilde{M}_k - \delta M_k\|^2}{\|\delta M_k\|} + \gamma\|\delta\bar{M}_k\|.$$

This differs from (5.2) since the tilde (˜) has been changed to a hat (ˆ) on the left side. (Recall from step 8 of the algorithm that $\delta\hat{M}_k = \delta\tilde{M}_k + \delta\bar{M}_k$.) This completes the proof of the lemma. □

The fourth lemma provides a bound for the change in the nth singular value of $M - \delta M$ when moving along the path from δM_k toward $\delta \hat{M}_k$.

LEMMA 5.4. *At each iteration the nth singular value of the matrix* $M - [(1 - \gamma)\delta M_k + \gamma \delta \hat{M}_k]$ *is less than or equal to*

$$(1 - \gamma)[\sigma_n]_k + \gamma^2 q_k,$$

where

$$q_k = \|\hat{u}_k^H (\delta \hat{M}_k - \delta M_k)\|.$$

Proof. From Lemma 5.2 there exists $[\hat{u}]_k$ orthogonal to $[u_k]$ such that

$$[\hat{u}]_k{}^H (M - \delta M_k) = [u_n]_k^H (\delta \hat{M}_k - \delta M_k) - [\sigma_n]_k [v_n]_k^H.$$

To complete the proof of the lemma, consider that the norm of the product,

$$\begin{aligned}
&\left([u_n]_k + \gamma[\hat{u}]_k\right)^H \left[M - (1-\gamma)\delta M_k - (\gamma)\delta \hat{M}_k\right] \\
&\quad = \left([u_n]_k + \gamma[\hat{u}]_k\right)^H \left[(M - \delta M_k) - \gamma(\delta \hat{M}_k - \delta M_k)\right] \\
&\quad = (1-\gamma)[\sigma_n]_k [v_n]_k^H + \gamma\left[[\hat{u}]_k^H (M - \delta M_k) - [u_n]_k^H (\delta \hat{M}_k - \delta M_k) + [\sigma_n]_k [v_n]_k^H\right] \\
&\qquad - \gamma^2 \hat{u}_k^H (\delta \hat{M}_k - \delta M_k)
\end{aligned}$$

is less than or equal to the stated bound,

$$(1 - \gamma)[\sigma_n]_k + \gamma^2 q_k,$$

implying that the nth singular value is less than the stated bound. This completes the proof of the lemma. □

Proof of convergence of the algorithm is achieved by appealing to a Lyapunov function approach: The function

$$P_k = [\sigma_n]_k + g_k \|\delta M_k\|$$

decreases at each iteration by a positive definite function of $[\sigma_n]_k$ and $\|\delta \tilde{M}_k - \delta M_k\|$. Proof of convergence involves two steps. The first step is to show that the choice

$$\gamma_k = \min\left\{\frac{1}{4} q_k^{-1} \left([\sigma_n]_k + g_k \frac{(\|\delta \tilde{M}_k - \delta M_k\|)^2}{\|\delta M_k\|}\right), 1\right\}$$

and the choice $g_k = \min(\frac{1}{2}[\sigma_n]_k \|\delta \bar{M}_k\|^{-1}, g_{k-1})$ (when $[\sigma_n]_k \neq 0$; $g_k = g_{k-1}$ otherwise) imply that

$$\begin{aligned}
P_{k+1} - P_k \leq \max\Big[&- \frac{1}{16} q_k^{-1} ([\sigma_n]_k + g_k b_k \|\delta \tilde{M}_k - \delta M_k\|)^2, \\
&- \frac{1}{4} ([\sigma_n]_k + g_k b_k \|\delta \tilde{M}_k - \delta M_k\|)\Big],
\end{aligned}$$

which is less than zero unless δM_k satisfies the necessary condition. The second step is to show that the quantities q_k^{-1} and g_k are both bounded away from zero. It follows that (i) $\{[\sigma_n]_k\} \to 0$, (ii) $\{\|\delta \tilde{M}_k - \delta M_k\|\} \to 0$, (iii) the sequence $\{\delta M_k\}$ has a

subsequence converging to a matrix δM_*, and (iv) for k sufficiently large δM_k satisfies the necessary condition for being a minimum norm rank-reducing perturbation for some $\tilde{M}$ in a neighborhood of M. The first step is achieved in the following proposition.

PROPOSITION 5.5. *The quantities computed by the algorithm satisfy*

$$P_{k+1} - P_k \leq \max\left[-\frac{1}{16}q_k^{-1}([\sigma_n]_k + g_k b_k \|\delta\tilde{M}_k - \delta M_k\|)^2, \right.$$
$$\left. -\frac{1}{4}([\sigma_n]_k + g_k b_k \|\delta\tilde{M}_k - \delta M_k\|)\right],$$

where $P_k = [\sigma_n]_k + g_k\|\delta M_k\|$.

Proof. Since $\gamma_k \in (0,1)$ it follows from Lemmas 5.3 and 5.4, that

$$\begin{aligned} P_{k+1} - P_k &= \left[\sigma_{k+1} + g_{k+1}\|\delta M_{k+1}\|\right] - \left[\sigma_k + g_k\|\delta M_k\|\right] \\ &\leq (-\gamma_k[\sigma_n]_k + \gamma_k^2 q_k) + \gamma_k g_k \left(\|\delta\bar{M}_k\| - \frac{1}{2}b_k\|\delta\tilde{M}_k - \delta M_k\|\right) \\ &\leq -\frac{1}{2}\gamma_k\left\{[\sigma_n]_k + g_k b_k\|\delta\tilde{M}_k - \delta M_k\|\right\} + \gamma_k^2 q_k. \end{aligned}$$

The two inequalities above come from the definition of g_k, which implies that g_k is nonincreasing and that $g_k\|\delta\bar{M}_k\| \leq \frac{1}{2}[\sigma_n]_k$. The choice for γ_k given in step 9 of the algorithm minimizes the quadratic bound given above over $\gamma_k \in [0,1]$. Substituting this value for γ_k into the above bound and using the fact that $\gamma_k = 1$ only if $q_k \leq \frac{1}{4}([\sigma_n]_k + g_k b_k\|\delta\tilde{M}_k - \delta M_k\|)$, one obtains

$$P_{k+1} - P_k \leq \max\left[-\frac{1}{16}q_k^{-1}([\sigma_n]_k + g_k b_k\|\delta\tilde{M}_k - \delta M_k\|)^2 \right.$$
$$\left. -\frac{1}{4}([\sigma_n]_k + g_k b_k\|\delta\tilde{M}_k - \delta M_k\|)\right]. \qquad \square$$

To complete the proof of convergence, it remains to establish that the sequence $\{q_k\}$ is bounded. If $\{q_k\}$ is bounded and if $\{g_k\}$ does not have zero as a limit point, it follows from Proposition 5.5 that $\{[\sigma_n]_k\} \to 0$ and $\{\|\delta\tilde{M}_k - \delta M_k\|\} \to 0$. The following lemma demonstrates that $\{q_k\}$ is bounded.

LEMMA 5.6. *Given the earlier assumptions and the algorithm as stated above, the sequence* $\{q_k\}$ *is bounded.*

Proof. From the definition of q_k, Lemma 5.2 implies that

$$q_k \leq \frac{1}{[\sigma_{n-1}]_k}\|\delta\hat{M}_k - \delta M_k\|\left[\|\delta\hat{M}_k - \delta M_k\| + [\sigma_n]_k\right].$$

Since $\{P_k\}$ is decreasing, it follows that $\|\delta\tilde{M}_k - \delta M_k\| \leq \|\delta M_k\| \leq (P_0/g_k)$ and $[\sigma_n]_k \leq P_0$, implying that $\|\delta\hat{M}_k - \delta M_k\| \leq (3/2)(P_0/g_k)$. Boundedness of $[\sigma_{n-1}]_k$ and $\{g_k\}$ away from zero in Assumptions 1 and 2 imply that the sequence $\{q_k\}$ is bounded. $\square$

It is now possible to establish convergence of the algorithm.

THEOREM 5.7. *Given the algorithm as stated above and Assumptions* 1 *and* 2 *it follows that the algorithm terminates.*

Proof. Using the boundedness of q_k and Assumption 2, Proposition 5.5 implies that $\|\delta\tilde{M}_k - \delta M_k\|$ and $[\sigma_n]_k$ become arbitrarily small since the sequence $\{P_k\}$ remains positive. Since g_k is bounded below by a positive number the algorithm must terminate. $\square$

Some additional properties of the matrix $\delta\tilde{M}_k$ are useful. At each iteration, the matrix $\delta\tilde{M}_k$ satisfies several properties.

(i) It is an element of $\mathcal{S}(\tilde{M}_k; \mathbb{R}^{n\times m})$, where

$$\tilde{M}_k \triangleq M - (\delta M_k - \delta\tilde{M}_k) - [u_n]_k[\sigma_n]_k[v_n]_k^H$$

since $\tilde{M}_k - \delta\tilde{M}_k$ is equal to the rank deficient matrix, $M - \delta M_k - [u_n]_k[\sigma_n]_k[v_n]_k^H$.

(ii) The vector $[u_n]_k$ is contained in the left null-space of the rank deficient matrix $\tilde{M}_k - \delta\tilde{M}_k$ and the columns of V_k span the right null-space.

(iii) $\delta\tilde{M}_k$ is the real matrix having minimum norm over all $\delta M \in \mathbb{R}^{n\times m}$ satisfying

$$[u_n]_k^H \delta M V_k = [u_n]_k^H \tilde{M}_k V_k.$$

The previous equation is equivalent to the constraint in step 5 of the algorithm, since $[u_n]_k^H \tilde{M}_k V_k = [u_n]_k^H \delta\tilde{M}_k V_k = [u_n]_k^H \delta M_k V_k$.

(iv) The sequence $\{\tilde{M}_k\} \to M$.

The previous properties show that $\delta\tilde{M}_k \in \mathcal{S}(\tilde{M}_k; \mathbb{R}^{n\times m})$. Moreover, $\delta\tilde{M}_k$ satisfies the necessary condition of Theorem 4.2 for being a minimum norm element of the set $\mathcal{S}(\tilde{M}_k; \mathbb{R}^{n\times m})$. For k sufficiently large, $\{\tilde{M}_k\}$ approaches M, i.e., δM_k satisfies the necessary condition for the neighboring matrix $\tilde{M}_k$. Hence $\|\delta\tilde{M}_k - \delta M_k\| + [\sigma_n]_k$ provides an estimate of the accuracy of the solution in the sense that it measures the distance from a nearby problem for which $\delta\tilde{M}_k$ satisfies the necessary condition exactly. This discussion is summarized in the following theorem.

THEOREM 5.8. *When the algorithm terminates, $\delta\tilde{M}_k$ satisfies the necessary condition for being a minimum norm element of $\mathcal{S}(\tilde{M}_k; \mathbb{R}^{n\times m})$, where $\tilde{M}_k = M - (\delta M_k - \delta\tilde{M}_k) - [u_n]_k[\sigma_n]_k[v_n]_k^H$ and $\|M - \tilde{M}_k\| \le \|\delta\tilde{M}_k - \delta M_k\| + [\sigma_n]_k \le \epsilon(1 + 1/g_k)\|M - \delta M_k\|$.*

Proof. Matrix $\delta\tilde{M}_k$ satisfies the necessary condition for being a minimum norm real matrix that satisfies $\operatorname{rank}[\tilde{M}_k - \delta\tilde{M}_k] = n - 1$. See the previous discussion. □

Theorem 5.7 demonstrates only that the algorithm would terminate if the algorithm were carried out with infinite precision. However, the stopping criterion given in the algorithm statement is designed to be achievable in the presence of numerical roundoff. The reasons for the criterion given in the algorithm are discussed in the next section.

6. Implementation and numerical results. A listing of a MATLAB [7] implementation of the algorithm is included in Appendix B. This implementation has been tested on a SPARC workstation which has a 53-bit effective floating point mantissa and a machine epsilon of 2.22×10^{-16} as reported by MATLAB.

Steps 4 and 5 of the implementation require some explanation. The least squares solutions in steps 4 and 5 are formulated using Kronecker products [1] by applying the "vec" operator to the equation. The relevant equation in step 4 becomes

$$V_k^T \otimes [u_n]_k^H \operatorname{vec}[\delta\bar{M}] = \operatorname{vec}\begin{bmatrix}[\sigma_n]_k & 0 & \dots & 0\end{bmatrix}. \tag{6.1}$$

The real minimum norm solution is obtained from the equation

$$\operatorname{vec}[\delta\bar{M}] = \begin{bmatrix}\operatorname{Re}[V_k^T \otimes [u_n]_k^H] \\ \operatorname{Im}[V_k^T \otimes [u_n]_k^H]\end{bmatrix}^+ \operatorname{vec}\begin{bmatrix}[\sigma_n]_k & 0 & \dots & 0\end{bmatrix}. \tag{6.2}$$

A sufficient number of zeros are added to the row vector in (6.2) to make the dimensions conform. The pseudoinverse required in (6.2) is obtained from a singular value decomposition (SVD).

A similar formulation yields $\delta\tilde{M}_k$. Step 5 requires solution of the equation

$$\begin{bmatrix} \mathrm{Re}[V_k^T \otimes [u_n]_k^H] \\ \mathrm{Im}[V_k^T \otimes [u_n]_k^H] \end{bmatrix} \mathrm{vec}[\delta\tilde{M}] = \begin{bmatrix} \mathrm{Re}[V_k^T \otimes [u_n]_k^H] \\ \mathrm{Im}[V_k^T \otimes [u_n]_k^H] \end{bmatrix} \mathrm{vec}[\delta M_k].$$

The MATLAB implementation obtains the solution in step 5 by projecting $\mathrm{vec}[\delta M_k]$ along the kernel of the matrix

$$\begin{bmatrix} \mathrm{Re}[V_k^T \otimes [u_n]_k^H] \\ \mathrm{Im}[V_k^T \otimes [u_n]_k^H] \end{bmatrix}. \tag{6.3}$$

The kernel of the matrix is obtained from the result of the SVD computed in step 4.

Step 8 also requires some explanation. The norm of the pseudoinverse in step 8 increases as $[\sigma_n]_k$ decreases. However, by construction, the vector on the left of $(M - \delta M_k)^+$ is orthogonal to the singular space associated with $[\sigma_n]_k$. Hence, only the first $n-1$ singular values and vectors are used for the inversion.

The stopping criterion in step 6 has been selected to account for the conditioning of the problem. It is well known that $[\sigma_n]_k$ can only be computed within a few multiples of $\epsilon[\sigma_1]_k$, where ϵ is the machine constant. As discussed earlier, the quantity g_k is an indirect measure of the conditioning of the inversion required to find $\delta\tilde{M}_k$. Hence, roundoff errors are magnified by $1/g_k$ in the computation of $\delta\tilde{M}_k$. Moreover, it can be shown that the projector used to compute $\delta\tilde{M}_k$ has a relative accuracy proportional to $1/g_k$. Because g_k is a decreasing sequence, it represents the worst conditioning encountered during execution of the algorithm. This discussion suggests that the relative error associated with the product $g_k\|\delta\tilde{M}_k - \delta M_k\|$ is approximately equal to the machine constant epsilon, and motivates the stated stopping criterion. While this is not a rigorous justification of the stopping criterion, this stopping criterion has been successful in numerous numerical experiments. Moreover, it is suspected that $1/g_k$ measures the sensitivity of the problem to changes in M.

The implementation given in Appendix B has been written to expose the main steps in the algorithm to guarantee convergence and to compute an accurate solution. A few comments on the storage requirements and the time requirements are in order. Estimating time requirements for SVD algorithms is difficult because the time required per iteration is a complex function of the number of rows and columns of the matrix. The time also depends on which portions of the decomposition are required. See, for example, [5]. Moreover, the number of iterations required depends upon the given matrix. Hence, estimates for SVD computation time must be partly theoretical and partly experimental.

For the following discussion, assume that the row and column dimensions are approximately equal $m \approx n$, and that the number of columns exceeds the number of rows by a small number, say $m - n + 1 << n^{1/2}$. The SVD of $M - \delta M_k$ computed in step 2 is the primary contributor to execution time in the implementation given in Appendix B. Step 2 requires a complete SVD of the complex matrix $M - \delta M_k$. The time required for this operation is proportional to n^3 and numerical experiments estimate that MATLAB requires approximately $50n^3$ floating point operations (flops) to compute this SVD. Step 4 requires a real SVD of a matrix whose dimensions are $2(m-n+1)$ by nm. However, the full decomposition is not required. Omitting computation of the null vectors saves computation time. The time required by the

economy version of the SVD in MATLAB, which omits the null vector computation, is proportional to $4(m-n+1)^2nm$ and experiments estimate that MATLAB requires approximately $20(m-n+1)^2n^2$ flops to compute this decomposition. Hence, the second SVD in step 4 requires comparatively little time due to its small number of rows provided p is small. The remaining operations are mostly matrix by vector multiplications, diagonal matrix inversions, matrix additions, and norm computations. The additional operations require approximately $18n^2p + 37n^2 + 6n + 8p^2 + 6p$ flops per iteration where $p = (m - n + 1)$.

Storing the matrices, M, δM_k, $\delta\hat{M}_k$, $\delta\tilde{M}_k$, and $\delta\bar{M}_k$ requires $6nm$ real storage locations, although this could be economized by using the same locations for several of these quantities. Storing the SVD of $M - \delta M_k$ requires $2n^2 + 2m^2 + n$ real storage locations. The matrix whose pseudoinverse is required in step 4 requires $2nmp$ storage locations. Storing the SVD of this matrix requires $4p^2+2nmp+2p$ locations. Together this is about the same as $4p$ real $n \times m$ complex matrices. Assuming $n \approx m$ storage of all matrices requires approximately $(10 + 4p)n^2$ real storage locations. Additional storage is required for intermediate quantities. The additional storage amounts to approximately $4n$ additional locations, assuming $m \approx n$.

Some simple numerical examples illustrate the accuracy of the technique. The first example has a known matrix that satisfies the necessary condition exactly.

Example 6.1. Consider the matrix,

$$M = \begin{bmatrix} -122-128j & -256+128j & -511-128j & -515-128j & -641-384j \\ -130-128j & -256-128j & -511+128j & -511-384j & -385-128j \\ 142-128j & 256-128j & 511-128j & 505-384j & 641+128j \\ 118-128j & 256+128j & 511+128j & 517-128j & 385+384j \end{bmatrix}.$$

The matrix

$$\delta M_* = \begin{bmatrix} 6 & 0 & 1 & -3 & -1 \\ -2 & 0 & 1 & 1 & -1 \\ 14 & 0 & -1 & -7 & 1 \\ 10 & 0 & -1 & 5 & 1 \end{bmatrix}$$

exactly satisfies the necessary condition. The algorithm given in Appendix B converges to a solution after 17 iterations requiring approximately 11,000 flops per iteration. The Frobenius norm of the error matrix obtained by subtracting the numerical solution from δM_* as given above is 3.9×10^{-13}. Note that the relative error obtained by dividing the absolute error by the Frobenius norm of $M - \delta M_*$ is approximately 0.5ϵ. The algorithm returned an error estimate of 3.4×10^{-13} where this is the distance away from the problem whose solution is obtained exactly. The final value for g was 0.199.

To further evaluate the accuracy of the approach, the algorithm was tested on several matrices where the solution was known approximately. By a solution, we mean a matrix satisfying the necessary condition given in Theorem 4.2. Several steps are taken to generate such matrices. First, a random complex matrix is generated whose real and imaginary parts have a specified size. The real portion of the matrix is generated by selecting entries randomly between zero and one from a uniform distribution. A normalization provides the real part with the specified size. The same procedure yields an imaginary part for the matrix having a specified size. An SVD of the complex matrix, call it R, yields an approximate left singular vector u_n and an approximate orthonormal basis for the direct sum of the right singular space associated with σ_n and the right null-space V. The quantities u_n and V are considered to be exact specifications for the desired null-space bases at the solution point. The subsequent steps

TABLE 1
Numerical results obtained by varying dimension.

Size	Iterations required	k-flops count	σ_{n-1}	Estimated error	Actual error	Relative error
5	8	10	0.1258	5.8×10^{-15}	1.1×10^{-16}	0.16
15	9	205	0.0306	1.3×10^{-15}	4.3×10^{-17}	0.14
30	12	1463	0.0094	1.8×10^{-15}	1.5×10^{-16}	0.46
50	19	6160	0.0031	4.6×10^{-15}	9.2×10^{-16}	2.92
100	21	47300	0.0027	4.8×10^{-15}	1.5×10^{-16}	4.71

attempt to generate a problem whose solution has u_n as a left null-vector and has V as a basis for the right null-space of $M - \delta M_*$. To generate δM_* in the appropriate space, a random vector in $\mathbb{R}^2$ is selected having entries uniformly distributed between zero and one and normalized to have a specified size. The transpose of the matrix given by (6.3) is multiplied by this random vector to generate $\text{vec}[\delta M_*]$. The matrix M is made rank-deficient via $M = R - u_n \sigma_n v_n^H$ to produce a matrix M having approximately the desired left and right null-spaces. This approach yields a complex matrix that is a perturbed version of a complex matrix having the specified null-space, since u_n, σ_n, and v_n^H exactly specify the singular space of a matrix near R. The real matrix δM_* will be a perturbed version of a matrix that is contained in the row space of (6.3). Both matrices will have a relative error approximately equal to the machine epsilon. Thus, the real matrix is a perturbed solution to a perturbed problem, both perturbations having a relative size approximately equal to the machine epsilon.

The technique described above was used to study the accuracy of the algorithm. Table 1 compiles results obtained by varying the size of the problem. The algorithm was tested on square matrices for which the norm of the real and imaginary parts were both approximately one. The norm of the solution was selected to be 0.001. The estimated error displayed in the table was computed by the algorithm. The actual error was obtained by computing the norm of the difference of the computed solution from the approximately known solution. The relative error is normalized to the size of $M - \delta M_*$, where δM_* is the approximately known solution and specified in multiples of the machine epsilon. The flop counts are average counts per iteration and are given in thousands of flops per iteration.

Similarly, Table 2 displays results obtained by varying the size of the imaginary part of M while holding the size of the real part at one and holding the size of the solution fixed at 0.001. The size of M was 10×12. The final value of g_* has been included in Table 2 as it provides some measure of the independence of the real and imaginary parts of the left singular vector at the solution point. The flop count has been omitted since each case had the same flop count of 135,000 flops per iteration.

7. Conclusions. This paper addresses two issues concerning the computation of the real-restricted singular value of a complex general matrix: the continuity of the singular value over the space of complex matrices and a necessary condition that a minimum norm rank-reducing real perturbation must satisfy. It presents an algorithm that computes a matrix satisfying a necessary condition for being a real, minimum norm rank-reducing perturbation for a problem in a neighborhood of the original problem. Upon termination the algorithm provides an estimate for the distance from the original problem. Results demonstrating the accuracy of the algorithm are included.

TABLE 2
Results obtained from varying size of imaginary part.

$\|\operatorname{Im} M\|$	Iterations required	g_*	σ_{n-1}	Estimated error	Actual error	Relative error
10^0	9	0.3129	0.0748	1.7×10^{-15}	1.7×10^{-16}	0.5
10^{-1}	19	0.1479	0.0559	2.6×10^{-16}	1.5×10^{-16}	0.7
10^{-2}	26	0.0103	0.0556	5.7×10^{-15}	9.8×10^{-16}	4.4
10^{-3}	49	0.00194	0.0556	1.1×10^{-13}	2.8×10^{-15}	12.5
10^{-4}	173	4.6×10^{-4}	0.0556	2.4×10^{-13}	4.2×10^{-15}	18.8
10^{-5}	1339	5.2×10^{-5}	0.0556	3.9×10^{-12}	5.3×10^{-14}	238

The algorithm is accurate when the real and imaginary parts of the left singular vector associated with the solution point are far from dependence. The rate of convergence is linear, and the algorithm performs well on many problems. However, convergence can be slow for some problems, particularly when the size of the imaginary part of the matrix is small compared to the real part or when the imaginary part is nearly singular. These conditions can cause the real and imaginary parts of the left singular vector at the solution point to be nearly dependent. Methods for accelerating convergence need further study. The number and nature of points that may satisfy the necessary condition given in Theorem 4.2 and techniques for selecting initial starting points also need further study.

Appendix A. The goal of this appendix is to provide supplemental proofs that allow removal of Assumption 2 given in §5 of the paper. Two preliminary lemmas are presented before the main result.

LEMMA A.1. *Let* $M \in \mathbb{C}^{n\times m}$, $m \geq n \geq 2$. *Suppose* $\operatorname{rank}[M] < n$ *and* $\operatorname{rank}[\operatorname{Im}[M]] = n$. *Let* u *be any nonzero vector satisfying* $u^H M = 0$. *The spectral condition number of the matrix* $[\operatorname{Re}[u],\ \operatorname{Im}[u]]$ *is less than or equal to* $\sigma_1[\operatorname{Re}[M]]/\sigma_n[\operatorname{Im}[M]]$.

Proof. Without loss of generality, assume $\operatorname{Re}[u]^T \operatorname{Im}[u] = 0$ and $\|\operatorname{Re}[u]\| \leq \|\operatorname{Im}[u]\|$. Clearly,

$$\sigma_1[\operatorname{Re}[M]]\,\|\operatorname{Re}[u]\| \geq \|\operatorname{Re}[u]^T \operatorname{Re}[M]\| = \|\operatorname{Im}[u]^T \operatorname{Im}[M]\| \geq \sigma_n[\operatorname{Im}[M]]\,\|\operatorname{Im}[u]\|,$$

implying that $\kappa[\operatorname{Re}[u],\ \operatorname{Im}[u]] = \|\operatorname{Im}[u]\|/\|\operatorname{Re}[u]\| \leq \sigma_1[\operatorname{Re}[M]]/\sigma_n[\operatorname{Im}[M]]$. □

LEMMA A.2. *The quantities computed by the algorithm described in the paper satisfy*

$$\|\delta\bar{M}_k\| \leq \sqrt{2}[\sigma_n]_k\, \kappa[\operatorname{Re}[u_n]_k,\ \operatorname{Im}[u_n]_k].$$

Proof. There exists a minimum norm real matrix δM_0 satisfying

$$[u_n]_k^H \delta M_0 = [u_n]_k^H([u_n]_k[\sigma_n]_k[v_n]_k^H)$$

as per Lemma 2.2. Hence δM_0 satisfies

$$[u_n]_k^H \delta M_0 V_k = [u_n]_k \delta\bar{M}_k V_k$$

implying that $\|\delta M_0\| \geq \|\delta\bar{M}_k\|$. It can be shown that

$$\|\delta\bar{M}_k\| \leq \|\delta M_0\| \leq \sqrt{2}[\sigma_n]_k \kappa[\operatorname{Re}[u_n]_k,\ \operatorname{Im}[u_n]_k]$$

implying the statement of the lemma. □

PROPOSITION A.3. *If the algorithm stated in §5 is modified so that* $g_0 = \frac{1}{4}\sigma[\mathrm{Im}[M]/\|\delta M_0\|$ *where the matrix* δM_0 *satisfies* $\mathrm{rank}[M - \delta M_0] = n - 1$ *then* $\lim\{g_k\} > 0$.

Proof. Since $[u_n]_k^H[M - \delta M_k - [u_n]_k[\sigma_n]_k[v_n]_k^H] = 0$, it follows from Lemma A.1 that

$$\kappa\,[\,\mathrm{Re}[u_n]_k,\quad \mathrm{Im}[u_n]_k\,] < \frac{\sigma_1[\mathrm{Re}[M]] + \sigma_1[\delta M_k] + [\sigma_n]_k}{\sigma_n[\mathrm{Im}[M]] - [\sigma_n]_k}$$

so long as $[\sigma_n]_k < \sigma_n[\mathrm{Im}[M]]$ (the initial choice for P_0 guarantees that $[\sigma_n]_k < \frac{1}{4}\sigma_n[\mathrm{Im}[M]]$ since $P_0 = g_0\|\delta M_0\| = \frac{1}{4}\sigma_n[\mathrm{Im}[M]]$ and $[\sigma_n]_k \le P_k \le P_0$.) Suppose that g_k is strictly less than g_{k-1}. From Lemma A.2 and the definition of g_k,

$$g_k = \left(\frac{1}{2}\right)\frac{[\sigma_n]_k}{\|\delta M_k\|} \ge \frac{1}{2\sqrt{2}}\,\frac{\sigma_n[\mathrm{Im}[M]] - [\sigma_n]_k}{\sigma_1[\mathrm{Re}[M]] + \|\delta M_k\| + [\sigma_n]_k}.$$

Since $P_k \le \frac{1}{4}\sigma_n[\mathrm{Im}[M]]$ for all k, it follows that

$$[\sigma_n]_k \le \tfrac{1}{4}\sigma_n[\mathrm{Im}[M]]$$

and $\|\delta M_k\| \le 1/(4g_k)\sigma_n[\mathrm{Im}[M]]$. Hence

$$g_k \ge \frac{3}{8\sqrt{2}}\,\frac{\sigma_n[\mathrm{Im}[M]]}{\sigma_1[\mathrm{Re}[M]] + (\frac{1}{4g_k} + \frac{1}{4})\sigma_n[\mathrm{Im}[M]]}$$

from which it follows that there is a constant C for which $g_k \ge C > 0$ for all k. We have shown that $g_k < g_{k-1}$ for any k implies that $g_k \ge C > 0$ and g_k cannot converge to zero. □

Appendix B. This appendix provides a listing of a MATLAB implementation of the algorithm described in the paper.

```
% function rrp = realrrp(M)
% M is an complex matrix (n by m) with n <= m
%
% Name:  realrrp
% Purpose:  This algorithm finds a matrix satisfying a necessary
%           condition for being a minimum norm, real, rank-reducing
%           perturbation for a given complex matrix
% Description: This algorithm uses a Lyapunov function to guarantee
% convergence to a solution point.
%
% Remarks:  This function must be called with the number of rows
% smaller than the number of columns
%
% Variable descriptions:
%   M:  User supplied n by m complex matrix
%   n:  Row dimension of M
%   m:  Column dimension of M
%   dm: Current approximation of solution
%   M0:  Current value of M - dm
%   u,s,v:  SVD of M0
%   Luv:  Large operator formed from u and v.
```

```
%   U,S,V:  SVD of Luv.
%   dmhat: See algorithm description
%   dmtilde: See algorithm description
function rrp = realrrp (M)
   tol = 8.0*eps;
   limit = 10000;
   printint = 10;
   flops(0);
%
%  PRELIMINARY CHECKS
%
%  Check for obvious problems...
%
% First check dimensions
%
   sz=size(M);
   n = sz(1); m=sz(2);
   if (m < n)
     disp  ('Error in minrealrrp.m:');
     error ('   Row dimension exceeds column dimension');
   end;
%
% Check size of imaginary part of M
%
   s = svd(imag(M));
   if (s(n) < tol*normm);
      disp  ('Error in Algorithm.m:');
      error ('    Size of imaginary part of M is too small');
   end;
%
% INITIALIZATION SECTION
%
%  Set up some useful quantities...
%
   normm = norm(M,'fro');
   done = 0; % Initialize flag for program termination.
   count = 0;
   g = 1;
%
% Step 1.  Get an initial real, rank-reducing perturbation matrix.
%
   dm = initrealrrp(M);
   normdm = norm(dm,'fro');
%
% ITERATION SECTION
%
  while (count < limit & done == 0)
    count = count + 1;
    if (round(count/printint)*printint == count)
      printthis = 1;
    else
      printthis = 0;
    end;
%
```

```
% Step 2.  Compute SVD of M - dm to get U, V, sigma_n and sigma_n_1
%
    M0 = M-dm;
    [u,s,v]=svd(M0);
    sigma_n = s(n,n);
    sigma_n_1 = s(n-1,n-1); % second from last
%
% Step 3.  Define Vk, and some other useful names
%
    V = v(:,n:m);
    U = u(:,n);
    S = [s(n,n:m)'; zeros(m-n+1,1)];
    Luv = kron (conj(V'), U');
    Luv = [real(Luv); imag(Luv)];
%
% Step 4.  Compute dmbar, which satisfies U'*dmbark*V = S, via a
%          least squares solution;
%
    [vl,sl,ul]=svd(Luv',0); % Economy style SVD
    vecdmbar = (vl*(diagpinv(sl, tol)*(ul'*S)));
    dmbar = unvec (vecdmbar, n, m);
    normdmbar = norm (dmbar, 'fro');
%
% Step 5.  Compute dmtilde (with a projector), which satisfies
%          U'*dmtilde*V = U'*dm*V
%
    rnk = diagrank (sl, tol);
%
% Project 'dmk' along kernel of Luv
%
    vecdmtilde = vl(:,1:rnk)*(vl(:,1:rnk)'*dm(:));
    dmtilde = unvec (vecdmtilde, n, m);
    normtilde = norm (dmtilde-dm, 'fro');
%
% Step 6.  Test for exit
%
    if (sigma_n + g*normtilde <= tol*s(1,1))
       done = 1;
    end;
%
% Step 7.  Adjust g if necessary
%
    if (sigma_n ~= 0)
       g = min ([g, 0.5*sigma_n/normdmbar]);
    end;
%
% Step 8.  Determine uhat
%
    dmhat  = dmtilde+dmbar;
    deldm = dmhat-dm;
    x = U'*(deldm) - s(n,n)*v(:,n)';
    uhat = ((x*v(:,1:n-1))*diagpinv(s(1:n-1,1:n-1),tol))*u(:,1:n-1)';
%
% Step 9.  Determine b, q, and finally gamma
```

```
%
%   First q
%
    q = norm(uhat*(deldm));
%
%   Now b
%
    if (normdm == 0)
       b = 0;
    else
       b = normtilde / normdm;
    end;
%
%   And gamma
%
    d = 0.25*(sigma_n + g * b * normtilde);
    if (d > q)
       gamma = 1;
    else
       gamma = (d / q);
    end;
%
% Step 10.  Compute the new delta M
%
    dm = (1-gamma)*dm + gamma*(dmhat);
    normdm = norm(dm, 'fro');
    if (printthis == 1)
       count, normdmbar, normtilde, gamma, g, q, sigma_n
    end;
  end;
disp ('Solution is '); disp (dm);
disp ('Final g value is '); disp(g);
disp ('Distance from desired problem is'); normtilde+sigma_n
disp ('Number of iterations is '); disp (count);
disp ('Flop count is '); flops
rrp = dm;
```

REFERENCES

[1] J. W. BREWER, *Kronecker products and matrix calculus in system theory*, IEEE Trans. Circuits Syst., CAS-25 (1978), pp. 772–781.

[2] R. BYERS, *Bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.

[3] R. EISING, *The distance between a system and the set of uncontrollable systems*, in Proc. MTNS (June 1983), Springer-Verlag, Beer-Sheva, 1984, pp. 303–314.

[4] ———, *Between controllable and uncontrollable*, Syst. Contr. Lett., 4 (1984), pp. 263–264.

[5] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.

[6] C. S. KENNEY AND A. J. LAUB, *Controllability and stability radii for companion form systems*, Math. Control Signals Systems, 1 (1988), pp. 239–256.

[7] THE MATHWORKS, INC., *MATLAB*, (c) Copyright 1984–1992, Sherborn, MA.

[8] G. W. STEWART, *Rank degeneracy*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 403–413.

[9] C. F. VAN LOAN, *How near is a stable matrix to an unstable matrix?*, Contemp. Math., 47 (1985), pp. 465–478.

[10] M. A. WICKS AND R. A. DECARLO, *Computing the distance to an uncontrollable system*, IEEE Trans. Automat. Control, AC-36 (1991), pp. 39–49.

[11] W. M. WONHAM, *Linear Multivariable Control: a Geometric Approach*, Springer-Verlag, New York, 1979.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1208–1225, October 1994

011

ROW ORDERING FOR A SPARSE QR DECOMPOSITION *

THOMAS H. ROBEY† AND DEBORAH L. SULSKY‡

Abstract. A new row ordering strategy based on pairing rows to minimize local fill-in is presented. The row ordering can be combined with most column ordering strategies to reduce computation, maintain sparsity, and solve rank deficient problems. Comparison of the new row pairing algorithm with Duff's fixed pivot row ordering on a collection of sparse matrix test problems shows a median 47–71% reduction, depending on the column ordering, in floating point operations (flops) required for the QR decomposition. On a finite element application using nested domain decomposition for the column ordering, the new row ordering is competitive with the row ordering from nested domain decomposition.

Key words. QR decomposition, sparse matrices, Givens rotations

AMS subject classification. 65

1. Introduction.

The QR decomposition is a relatively robust means for solving a variety of problems. The decomposition of an $m \times n$ matrix A ($m \geq n$) is usually written $QAP = \begin{bmatrix} R \\ 0 \end{bmatrix}$, where Q is an $m \times m$ orthogonal matrix composed of orthogonal transformations, P is a permutation matrix, and R is an upper triangular matrix. The column permutations are usually required when A does not have full rank and then R is upper trapezoidal. The use of orthogonal transformations is numerically stable and, in practice, the rank of the matrix can be determined accurately when column permutations are performed during factorization. The matrix Q can be determined in factored form using several methods; Givens rotations are used here. While the LU decomposition (Gaussian elimination) is widely used and researched, the QR decomposition is used less frequently in applications because it is generally more expensive to compute than the LU decomposition. However, there are many applications that produce rank deficient matrices [1], or that require least squares solutions, for which Gaussian elimination is unsuitable. Furthermore, many of these applications produce sparse matrices, a property that can be exploited to reduce the cost of computation.

To design a sparse QR decomposition, the overall goals must first be identified. The primary goal is to minimize the execution time for computing the decomposition. Since execution times are highly dependent on computer architecture and the algorithm's implementation, it is desirable to have other less variable measures of computational cost. Flops are the primary measure of computational work. Indeed, for sparse matrices, column and row permutations are motivated by an attempt to preserve sparsity and thereby reduce flops. Execution time can also be impacted by the overhead associated with choosing appropriate reorderings of rows and columns. It is important that this overhead does not offset gains achieved by reducing flops. Other measures of performance are the amount of intermediate fill-in, the number of

* Received by the editors August 6, 1990; accepted for publication (in revised form) June 10, 1993.

† Spectra Research Institute, 1613 University Boulevard NE, Albuquerque, New Mexico 87102 (trobey@math.unm.edu) This author's research was conducted at the Department of Mathematics and Statistics, University of New Mexico, Albuquerque, New Mexico 87131-1141.

‡ Department of Mathematics and Statistics, University of New Mexico, Albuquerque, New Mexico 87131-1141.

nonzeros in R, and the number of Givens rotations. The amount of intermediate fill-in and the number of nonzeros in R are measures of how well sparsity is maintained. It is desirable to keep R as sparse as possible since most applications use R. Some applications use Q, and thus the storage for Q can be an issue. Since Q is stored in factored form, the number of Givens rotations performed during the decomposition is a measure of the requisite storage. These measures will be used to gauge the relative success of sparse QR decomposition algorithms.

When A is sparse, both row and column permutations may be performed as a means of reducing computational expense and maintaining sparsity. While both row and column orderings impact the computational work, only the column ordering influences the final sparsity in R [2]. This observation follows from the fact that the Cholesky factor of the normal equations matrix A^TA mathematically has the same structure as R, and the structure of the Cholesky factor is determined solely by the column ordering. Column ordering strategies can be divided by whether the orderings are done a priori or at each step of the process (local orderings).

An a priori scheme for column ordering suggested by George and Heath [2] utilizes the relationship between R and the Cholesky factor of the normal equations matrix. Symbolically, the normal equations matrix is formed and then a minimum degree algorithm is performed on (A^TA) to determine a column ordering. The nonzero pattern of R can also be determined a priori, which facilitates memory allocation. This column ordering attempts to minimize the number of nonzeros in R. The amount of flops required to compute the decomposition is not necessarily reduced by this scheme, because different row orderings produce varying amounts of intermediate fill-in. To avoid forming A^TA, Ostrouchov [3] develops a symbolic minimum degree algorithm that operates on the structure of A that can apply local row and column orderings a priori. The local nature allows local tie-breaking criteria to be used during column ordering.

Duff [4] suggests a local column ordering that selects the column with the minimum number of nonzeros. This strategy seeks to locally minimize the number of Givens rotations and thus attempts to indirectly minimize other quantities.

Nested dissection is an a priori method that produces both column and row orderings [5] and is based on the underlying graph of the matrix. For problems where the underlying graph is a grid of nodes, these orderings have the advantage of bounds on both the number of flops and the number of nonzeros in R. An element-based variation is nested domain decomposition [1]. Both of these techniques successively split the problem into smaller problems. Then the decomposition proceeds in the reverse direction beginning with the small problems. The orderings produced are coarse, i.e., the rows and/or columns are ordered in groups with no ordering within each group. A disadvantage of nested dissection is that the quality of the ordering depends on the topology of the graph, and no reasonable bounds on flops or the number of nonzeros in R exist if an underlying grid does not exist.

Row ordering schemes can be divided into either fixed pivot row strategies or variable pivot row strategies. In each column, fixed pivot row strategies always use the same element, the pivot, to zero out all the other nonzeros in that column. Therefore the pivot eventually becomes the diagonal entry in R. Variable pivot row strategies allow the use of more than one element as a pivot in each column and the final pivot for the column becomes the diagonal entry in R. Note that fixed pivot row orderings are a subset of the variable pivot row orderings. Thus, in theory, there always exists a variable pivot row ordering that performs as well as the best fixed pivot row ordering although, in practice, fixed pivot strategies may outperform variable pivot strategies

[3]. Note that although the terminology row ordering strategy is used, these strategies are not necessarily equivalent to an a priori permutation of the rows [3].

The strategy of George and Heath [2] orders the rows in increasing order with respect to the column containing the last nonzero (after column permutations), an a priori fixed pivot scheme. George and Heath are mainly concerned with developing a column ordering that keeps R sparse, and their row ordering strategy outlined above makes only a minimal attempt to reduce intermediate fill-in and flops although they demonstrate a significant reduction in the cost of computation over the natural row ordering and the reverse ordering. Liu [6] introduces submatrix rotations to reduce the computational cost for the George and Heath strategy by imposing a coarse-grained row ordering. Liu's submatrix rotations can be interpreted as a variable pivot row strategy.

Duff [4] examines several local row ordering strategies and recommends a fixed pivot row strategy. The sparsest row is chosen as the pivot row and succeeding rows are processed based on minimizing fill-in of the pivot row. Gentleman [7] suggests a variable pivot row ordering based on processing the sparsest pair of rows first.

None of the row ordering schemes is completely satisfactory, since they often do not come close to minimizing intermediate fill-in. It seems impossible, short of an extensive search, to design an algorithm that minimizes intermediate fill-in in general. Our goal is a heuristic row ordering strategy that is easily implemented, efficient, and performs well on a large class of problems compared to other strategies.

The scheme presented in §2 introduces a row ordering strategy based on a variable pivot row. In §3, the strategy is compared on test matrices with Duff's fixed pivot row strategy and Gentleman's variable pivot row strategy using both Duff's column ordering strategy and a minimum degree column ordering. Section 4 considers a finite element application and compares the new row ordering strategy with Duff's fixed pivot row ordering strategy, Gentleman's scheme, and a row ordering from nested domain decomposition. Nested domain decomposition is used to obtain the column ordering.

On common test problems, the variable pivot row strategy performs at least as well as, and usually outperforms, the row ordering strategies cited above. The reduction in flops is as much as 90% for some test cases; furthermore, the improvement roughly appears to increase with the size of the matrix.

2. Implementation. For the present assume that A has full rank. The overall decomposition procedure involves successively applying Givens rotations to zero out elements below the diagonal in each column. At the start of the kth step, zeros have been placed below the diagonal in the first k-1 columns. Let the *active submatrix* be the lower right block of the matrix under current consideration. The general procedure involves two *stages* within each step. The first stage applies the column ordering strategy and the second stage applies the row ordering strategy and numeric operations. Begin with the active submatrix as the whole A matrix. Search for a column in the active submatrix using some local column ordering strategy or choose the next column if an a priori column ordering is being used. This column is called the *selected column* and is pivoted (if necessary) to the first column in the active submatrix. The first stage is then complete. The rows of the active submatrix with nonzeros in the selected column form the *selected set of rows.* In the second stage, the selected set of rows is operated on with Givens rotations to zero out all but one nonzero in the selected column. The remaining row with a nonzero is pivoted to the top of the active submatrix, and this row is then deleted along with the selected column to

form the next active submatrix. The process is repeated at the next step with the new active submatrix which now has one less column and row than the previous active submatrix.

If A does not have full rank, then the final pivot in each column must be monitored. If a zero (or small) final pivot is encountered, then the column is pivoted to the right and eliminated from the active submatrix. Note that the active submatrix is no longer in the lower right corner.

The algorithm presented above is a column oriented algorithm since each step completely processes one column. For the column ordering strategies cited above, a column oriented algorithm can handle large problems that do not fit into core memory by using an algorithm that reads in the selected set of rows at the beginning of the second stage. Note that it is not necessary to keep R in core, but it is necessary to access rows more than once. The number of times a row is accessed is small if sparsity is maintained. The application in §4 has been implemented using this out-of-core algorithm. As a comparison, the row oriented algorithm of George and Heath [2] reads each row exactly once and requires only a single working row in core, but also requires storage for R in core memory. The advantage of the George and Heath algorithm is not necessarily in handling large problems out of core, but in its use of a static storage structure that lends itself to programming languages lacking dynamic memory allocation. A detailed analysis of execution times and storage efficiency of the two out-of-core approaches is beyond the scope of this paper.

The way in which rows are processed in the second stage has a great effect on the amount of intermediate fill-in. Intermediate fill-in causes the sparsity to be lost and can result in more Givens rotations being required later in the decomposition. If it is assumed there is no chance cancellation, the following formula describes the relationship between the variables

$$\text{Nonzeros in A} + \text{intermediate fill-in} - \text{Givens} = \text{nonzeros in R}. \tag{1}$$

If the nonzeros in R are fixed by a given column ordering, then reducing intermediate fill-in must reduce the number of Givens rotations, thus making the scheme more efficient. For local column ordering strategies, intermediate fill-in can affect the column choice and therefore intermediate fill-in can indirectly affect the number of nonzeros in R. If the effect of intermediate fill-in on the number of nonzeros in R is small, then reducing intermediate fill-in still gives a comparable reduction in the number of Givens rotations.

Note that the process of applying a Givens rotation results in a pair of rows assuming the same zero-nonzero pattern except in the selected column (assuming no chance cancellation). Thus, it does not matter which row is considered to be the pivot row as far as the sparsity pattern is concerned; the resulting sparsity pattern is identical after the appropriate row permutation. These observations lead to the following strategy, which is called a *variable pair strategy.*

The variable pair strategy is based on minimizing local fill-in. If any rows are encountered that have the same zero-nonzero pattern, then the search for a pair of rows is halted and a Givens rotation is applied to the pair. Note that there is no resulting fill-in, and therefore such a strategy is both locally and globally optimal with respect to intermediate fill-in. If there are no pairs with the same zero-nonzero pattern, then the strategy is to choose the pair of rows that results in the least amount of local fill-in. Often two pairs of rows produce the same amount of fill-in. In this case, a tie-breaking strategy is based on a strategy of preserving the sparsity of future active submatrices by choosing the sparsest pair of rows (Gentleman's scheme [7]).

It is instructive to demonstrate the various row ordering strategies on some simple examples. Duff [4] suggests an example where the selected column has already been found and the Givens reduction stage is to begin. The active submatrix is

$$\begin{bmatrix} x & o & o & o & o & o \\ x & x & x & x & o & o \\ x & x & x & x & o & o \\ x & o & o & o & x & x \\ o & & & \vdots & & \\ \vdots & & & & & \\ o & & & & & \end{bmatrix}.$$

Processing the pairs in their natural order, i.e., (1, 2), (1, 3), and (1, 4) results in a fill-in of eight. Using Duff's fixed pivot scheme results in the pairings (1, 4), (1, 2), (1, 3), and a fill-in of nine. Gentleman's variable pivot scheme results in the same pairing as the fixed pivot scheme. The variable pair strategy results in the pairing (2, 3), (1, 4), and (1, 2) with a fill-in of seven.

Duff suggests an example for which Gentleman's variable pivot method fails to be optimal with respect to intermediate fill-in:

$$\begin{bmatrix} x & x & x & o & o & o & o \\ x & o & o & x & x & o & o \\ x & x & x & x & o & o & o \\ x & o & o & o & x & x & x \\ o & & & \vdots & & & \\ \vdots & & & & & & \\ o & & & & & & \end{bmatrix}.$$

Using the natural order, the fill-in is ten. The fixed pivot strategy produces a fill-in of nine. The variable pivot scheme results in the pairings (1, 2), (3, 4), and (1, 3) for a fill-in of twelve. The variable pair strategy results in the pairings (1, 3), (1, 2), (1, 4), and a fill-in of nine.

The variable pair strategy seems to produce an optimal ordering in the cases examined so far, but this is not always true. Consider the active submatrix

$$\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & o & o & x \\ x & o & o & o & x & x \\ o & & & \vdots & & \\ \vdots & & & & & \\ o & & & & & \end{bmatrix}.$$

Using the variable pair strategy the rows are processed in the order (1, 2) and (1, 3) with a fill-in of five. However, the order (2, 3) and (1, 2) produces a fill-in of four. Thus the variable pair strategy for minimizing fill-in at each step does not assure global minimization of intermediate fill-in.

The discussion above motivates the row ordering strategy by the desire to reduce flops; but, overhead associated with choosing a row ordering can also impact overall performance of the algorithm. Overhead is measured by the amount of integer arithmetic performed during the decomposition. To estimate the complexity of our row

ordering algorithm, consider one step of the decomposition. Let r be the number of rows in the selected set of rows at the beginning of the second stage. Let M be the number of nonzeros in the final pivot row at the end of the second stage. With no chance cancellations, the number of nonzeros in any row in the selected set at anytime during the second stage is bounded by M. Processing the active submatrix using the variable pair method involves the following setup.

1. Check for sparsity patterns that are exact matches and process those pairs of rows immediately. There are $O(r^2)$ pairs each requiring at most $O(M)$ comparisons; so this step has a bound on the number of comparisons that is $O(Mr^2)$.

2. Calculate fill-in for all pairs of rows and set up a list storing the members of the pairs and the corresponding fill-in. This step has a bound on the amount of integer arithmetic that is $O(Mr^2)$, but note that the number of rows might be less than r after step 1.

3. Sort the pairs in the list in order of the increasing amount of fill-in using Gentleman's scheme as a tie breaker ($O(r^2 \log r)$.

4. Perform a Givens rotation on the first pair in the list; the flops have an $O(M)$ bound.

If the above steps are repeated (at most r-1 times) to complete the second stage, the overhead is at worst $O(Mr^3)$ and the flops are bounded by $O(Mr)$. However, the overhead can be reduced by noting that after step 4 the list of pairs constructed in step 2 does not have to be entirely redone. A pair in the list that has the discarded row or the current pivot row as one of its members is no longer valid and is deleted from the list. Since the sparsity pattern in the current pivot row may have changed, fill-in must be recomputed for a pair that has the pivot row as a member. A temporary list is created, with at most r-2 entries, containing all pairs that involve the pivot row and its corresponding fill-in ($O(Mr)$). The temporary list is then sorted according to fill-in and merged with the original list ($O(r^2)$). This updating is done at most r-2 times. The overhead for the variable pair method is thus bounded by $O(Mr^2)+O(r^3)$ work. At any given step of the decomposition, the values of M and r depend on the row and column orderings. Of course, M is at most n and r is at most m. For a sparse matrix, r should generally be much less than m and M much less than n. If M and r are the same order, then both terms in the overhead bound are the same size, $O(Mr^2)$. In a sparse matrix with nonzero elements that in some sense are evenly distributed throughout the matrix so that the number of nonzeros in any column is roughly the same as the number of nonzeros in each row, it might be possible to keep $M = O(r)$ if the amount of intermediate fill-in is kept small.

In §3, the variable pair method is compared with Duff's fixed pivot method [4] and with Gentleman's scheme [7]. Given the same active submatrix as above, the fixed pivot method compares the sparsity structure of the pivot row to each remaining row in the selected set of rows to determine the order in which Givens rotations are applied. Overhead for this method is bounded by $O(Mr^2)$. On the other hand, Gentleman's variable pivot scheme does not use the sparsity structure. For the remaining rows in the selected set of rows, the total number of nonzeros for each pair of rows is used to pick the order in which rows are processed, making the overhead $O(r^2)$.

It is also important to remember that this analysis is for one active submatrix. The hope is that by using more information about the sparsity structure, the variable pair method and Duff's fixed pivot scheme will do better than Gentleman's scheme in the long run because the active submatrices will be sparser, i.e., the values of M and r will be kept small. This hope is not always realized, as shown in the examples in §3.

TABLE 1
Characteristics of the test matrices.

	Number of			Sparsity
	Rows	Columns	Nonzeros	(%)
CURTIS55[1]	55	54	292	9.8
WILL58	58	57	282	8.5
WILL200	200	199	702	1.8
ASH219	219	85	438	2.4
ABB313	313	176	1557	2.8
ASH331	331	104	662	1.9
ASH608	608	188	1216	1.1
LSQUAR15	784	225	3136	1.8
ASH958	958	292	1916	0.7
WELL1033	1033	320	4732	1.4
LSQUAR20	1444	400	5776	1.0
NETWORK4	1488	784	7040	0.6
NETWORK3	1512	402	7152	1.2
WELL1850[2]	1850	712	8758	0.7

[1] The matrix given in Curtis [9] is the transpose of that supplied in the Harwell–Boeing sparse matrix collection [8]. The matrix from the Harwell–Boeing sparse matrix collection is used here.

[2] The version used by Liu [6] has three fewer nonzero entries.

3. Examples. Ten test matrices from the Harwell–Boeing sparse matrix collection [8] as well as four matrices from George, Heath and Ng [9] (also used by Liu [6]), are used to compare the row ordering strategies. Three square matrices are altered by adding a row with a nonzero in the first column. These matrices, used by Duff [4], were developed by Curtis [10] and Willoughby [11]. WELL1033 and WELL1850 are surveying problems supplied by Michael Saunders. ASH219, ASH331, ASH608, and ASH958 are surveys of the United Kingdom and Holland supplied by V. Ashkenazi. ABB313 is a survey of Sudan supplied by M. Abbas. NETWORK3 and NETWORK4 are geodetic adjustment problems. LSQUAR15 and LSQUAR20 arise in the natural factor formulation of the finite element method. CURTIS55 and WILL58 both exhibit a dominant principal band with very sparse outliers. WILL200 has about thirty small bands scattered throughout the matrix. ASH219, ASH331, ASH608, and ASH958 all are banded about the principal diagonal. ABB313, LSQUAR15, and LSQUAR20 have two diagonal bands. WELL1033 and WELL1850 have several bands along with less structured patterns. NETWORK3 and NETWORK4 have one diagonal band extending to the right edge and scattered entries to the left of the other end of the diagonal band. Other characteristics of these test matrices are listed in Table 1.

First, a comparison is made of Duff's fixed pivot row strategy (DUFF) and Gentleman's variable pivot row strategy (GENT) with the variable pair strategy (VPAIR) using Duff's column ordering. Since Duff's algorithm does not uniquely specify the order of rows and columns (ties are broken arbitrarily), different implementations may give varying results. Duff [4] and Duff and Reid [12] report slightly different results and the implementation used here shows a similar variation in results. Our implementation breaks all ties by using the original or natural ordering (smallest original column or row number).

The results on the fourteen test matrices are shown in Table 2. Intermediate fill-in is the total number of nonzeros created at any time during the decomposition. QR flops is the number of multiplications performed during the decomposition using standard Givens rotations. Next, a comparison is made between DUFF, GENT, and VPAIR using the minimum degree strategy for the column ordering. The results are shown in Table 3.

The test matrices can be divided into categories of small, nearly square ($m \approx n$) matrices, and larger rectangular ($m \gg n$) matrices. The first three test matrices are small, nearly square matrices, and tend to have fewer choices for ordering simply because there are fewer rows. Larger rectangular matrices should demonstrate the differences between the methods to a greater degree since there is a greater chance of operating on previous intermediate fill-in and many more choices for row orderings.

The results demonstrate the differences between the two categories of test matrices. Nearly square matrices show little difference between the various row ordering strategies. Because of their shape, these matrices have the characteristic that the number of rows in each selected set must decrease in the last few stages of the algorithm (see Fig. 1). Typical behavior of the larger rectangular test matrices is for a rapid growth in the number of rows in the selected sets in the last few stages. Such behavior is indicative of operations on intermediate fill-in. The exception to this type of behavior is ABB313, which displays uneven growth in the number of rows in the selected sets. Of all the larger rectangular matrices, VPAIR shows the least improvement for ABB313.

Using Duff's column ordering strategy, VPAIR shows a 33% median improvement in intermediate fill-in over DUFF for the eleven larger rectangular test matrices. The number of Givens rotations has a median improvement of 35%, while a median improvement of 47% is observed in QR flops. The number of nonzeros in R is about the same for VPAIR and DUFF, except for ASH608 and ASH958 where a significant improvement is seen using VPAIR, and for LSQUAR15, LSQUAR20, and NETWORK3 where DUFF yields a sparser R than VPAIR. Since row ordering does not determine the sparsity of R, the different row ordering strategies reduce the number of nonzeros in R by allowing the local column ordering strategies to perform better. (Recall the discussion following (1).) Although, the results for WELL1033, WELL1850, LSQUAR15, LSQUAR20, NETWORK3, and NETWORK4 are a reminder that reducing intermediate fill-in does not necessarily reduce fill-in in R. For the small nearly square test matrices, mixed results with few differences were obtained. GENT performed worse than DUFF on ABB313, ASH331, LSQUAR15, LSQUAR20, NETWORK3, and NETWORK4, while slightly outperforming DUFF on the other five larger rectangular test matrices.

Using the minimum degree column ordering strategy on the matrix A (as opposed to A^TA), VPAIR shows even greater improvement over DUFF. The median improvement on the larger rectangular test matrices is 57% for intermediate fill-in, 53% for the Givens rotations, and 71% for the flops required for the QR decomposition. Again there is not much distinction between the methods for the nearly square matrices. Gentleman's row ordering strategy performs much better than DUFF on seven of the largest test matrices, but not nearly as well as VPAIR. On the other four test matrices, LSQUAR15, LSQUAR20, NETWORK3, and LSQUAR4, DUFF does better than GENT, but still not as well as VPAIR.

In these examples, the minimum degree column ordering strategy produces a sparser R matrix than Duff's column strategy; using VPAIR, the median improvement is 27% for the larger rectangular test matrices. VPAIR coupled with the minimum

TABLE 2
Duff's column strategy.

Matrix	Row strategy	Intermed. Fill-in	Givens rotations	QR flops	Nonzeros in R
	DUFF	471	178	5,446	585
CURTIS55	GENT	528	194	6,584	626
	VPAIR	529	189	5,976	632
	DUFF	326	137	3,678	471
WILL58	GENT	383	149	4,364	516
	VPAIR	321	140	3,518	463
	DUFF	3,373	755	46,972	3,320
WILL200	GENT	2,881	735	41,660	2,848
	VPAIR	2,954	777	47,566	2,879
	DUFF	1,552	1,290	26,968	700
ASH219	GENT	1,528	1,292	26,908	674
	VPAIR	1,045	838	13,974	645
	DUFF	3,784	2,797	96,682	2,544
ABB313	GENT	4,292	3,305	128,958	2,544
	VPAIR	3,534	2,665	86,866	2,426
	DUFF	2,837	2,364	61,482	1,135
ASH331	GENT	3,277	2,722	80,098	1,217
	VPAIR	2,046	1,630	36,596	1,078
	DUFF	8,406	6,557	267,766	3,065
ASH608	GENT	7,590	6,197	197,942	2,609
	VPAIR	4,436	3,350	94,760	2,302
	DUFF	13,582	11,976	584,256	4,742
LSQUAR15	GENT	21,909	19,336	1,279,130	5,709
	VPAIR	9,575	7,114	409,046	5,597
	DUFF	15,023	11,754	481,258	5,185
ASH958	GENT	13,560	10,967	434,710	4,509
	VPAIR	7,694	5,379	159,394	4,231
	DUFF	5,135	7,194	192,254	2,673
WELL1033	GENT	4,368	6,339	164,794	2,761
	VPAIR	3,052	5,015	102,654	2,769
	DUFF	35,591	30,513	2,363,760	10,854
LSQUAR20	GENT	61,505	53,860	5,224,686	13,421
	VPAIR	26,473	14,537	1,329,440	17,712
	DUFF	16,120	13,628	474,032	9,532
NETWORK4	GENT	20,682	18,566	738,356	9,156
	VPAIR	13,364	11,076	298,160	9,328
	DUFF	12,716	15,357	449,162	4,511
NETWORK3	GENT	14,244	17,033	476,426	4,363
	VPAIR	8,458	10,303	219,222	5,307
	DUFF	22,878	22,718	1,079,980	8,918
WELL1850	GENT	19,549	19,977	876,876	8,330
	VPAIR	13,435	13,480	463,734	8,713

degree column strategy generally performs better than VPAIR with Duff's column strategy with respect to flops, Givens, and intermediate fill-in. On the other hand, DUFF with Duff's column strategy generally performs better with respect to flops, Givens, and intermediate fill-in than DUFF coupled with the minimum degree column ordering strategy.

Increased time spent on row and column ordering can, in principle, offset savings in time gained by reducing flops. The amount of overhead for ordering rows and columns required by the various strategies is highly dependent on the implementation and the machine architecture. An indication of overhead costs can be obtained from the number of seconds required to perform the QR decomposition without computing statistics or performing flops. The overhead times are reported in Table 4 for the larger rectangular matrices; the smaller test matrices do not show significant differences among the various strategies. Also reported is the total execution time, except in cases where only the sparsity pattern was available and not the actual floating-point entries in the matrix.

Column and row ordering strategies can be divided into those that use information on the structure of the matrix as opposed to those that merely use the number of entries in the columns or rows. Methods that use more information about the structure typically require more overhead if the structure is fixed, but can have an overall advantage if they lead to sparser active submatrices during the decomposition process. Duff's column ordering strategy does not use information about the structure, while the minimum degree column ordering strategy does. For the row ordering strategies, DUFF and VPAIR use information on the structure while GENT does not.

The results in Table 4 show that, in these examples, the overhead for the minimum degree algorithm exceeds the overhead for Duff's column ordering except in a few cases. However, the results in Tables 2 and 3 show that R is sparser using the minimum degree column ordering that, in applications, might lead to savings elsewhere in the code. Also, a more efficient implementation of the minimum degree column ordering heuristic might be possible, which would reduce the overhead.

In examining overhead among the row ordering strategies, the results are mixed for ASH219, ABB313, and ASH331. For the larger test matrices, VPAIR generally performs better than GENT which, in turn, performs better than DUFF. There are, however, some exceptions to this pattern. The discussion at the end of the last section indicates that GENT could potentially be more efficient than DUFF or VPAIR. DUFF does tend to require more overhead than GENT on these examples. On the other hand, VPAIR generally has less total overhead than GENT because of its ability to reduce intermediate fill-in, as indicated by the values in Tables 2 and 3. The total execution times, including flops, follow the same pattern as the overhead times in Table 4. For all methods, the percentage of the total execution time devoted to overhead is about 85%. Note that timings could vary considerably for different implementations and architectures and may reflect some inefficiencies in coding. These computations were performed on a 486-33 in an 8-MHz mode.

Results in the literature using the same test matrices can also be used to judge the performance of VPAIR. Liu [6] uses a variable pivot row strategy with a minimum degree (MD) column ordering strategy on A^TA. Table 5 shows Liu's results on seven test matrices. For each Givens rotation, opcount is incremented by the number of nonzeros in the transformed pivot row and is roughly one-fourth of the QR flops using standard Givens. VPAIR shows a reduction ranging from 74–90% in the opcount compared to Liu's method.

Ostrouchov [3] uses a sophisticated implementation of Duff's fixed pivot row strat-

TABLE 3
Minimum degree column strategy.

Matrix	Row strategy	Intermed. Fill-in	Givens rotations	QR flops	Nonzeros in R
	DUFF	423	220	6,842	495
CURTIS55	GENT	430	227	7,590	495
	VPAIR	415	212	6,426	495
	DUFF	288	166	3,960	404
WILL58	GENT	288	166	4,044	404
	VPAIR	288	166	3,992	404
	DUFF	2,768	1,033	60,186	2437
WILL200	GENT	2,806	1,071	66,074	2437
	VPAIR	2,755	1,020	55,222	2437
	DUFF	1,583	1,508	29,554	513
ASH219	GENT	1,451	1,376	24,090	513
	VPAIR	861	786	10,790	513
	DUFF	2,726	2,692	69,892	1591
ABB313	GENT	2,403	2,369	56,124	1591
	VPAIR	2,171	2,137	46,948	1591
	DUFF	3,620	3,492	88,804	790
ASH331	GENT	2,654	2,526	54,332	790
	VPAIR	1,497	1,369	21,832	790
	DUFF	8,496	8,050	271,204	1662
ASH608	GENT	7,348	6,902	200,740	1662
	VPAIR	3,356	2,910	54,140	1662
	DUFF	11,484	11,637	524,814	2983
LSQUAR15	GENT	14,959	15,112	642,814	2983
	VPAIR	5,368	5,521	154,418	2983
	DUFF	17,923	17,196	664,390	2643
ASH958	GENT	11,470	10,743	305,518	2643
	VPAIR	5,384	4,657	88,754	2643
	DUFF	6,681	8,842	283,602	2571
WELL1033	GENT	4,673	6,834	174,630	2571
	VPAIR	2,893	5,054	98,790	2571
	DUFF	31,568	30,795	2,066,794	6549
LSQUAR20	GENT	42,318	41,545	2,582,878	6549
	VPAIR	12,928	12,155	454,046	6549
	DUFF	15,000	13,616	407,632	8424
NETWORK4	GENT	20,068	18,684	628,088	8424
	VPAIR	12,436	11,052	267,680	8424
	DUFF	12,256	15,261	418,786	4147
NETWORK3	GENT	14,722	17,727	480,446	4147
	VPAIR	7,000	10,005	193,378	4147
	DUFF	32,190	33,539	1,830,114	7409
WELL1850	GENT	21,247	22,596	954,842	7409
	VPAIR	11,876	13,225	385,782	7409

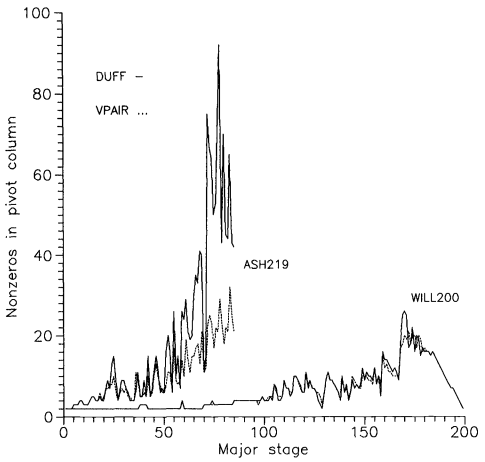


FIG. 1. *Types of test matrices.*

egy coupled with the MD column ordering with Duff's column ordering as a tie breaker (MD + Duff). Ostrouchov applies his method to ABB313 [13], WELL1033, and WELL1850 [14] and the results are summarized in Table 5. The number of Givens rotations and the opcount show improvements of 21–51% and 32–70%, respectively, for VPAIR over Ostrouchov's results. Ostrouchov recommends using the Duff tie breaker with the minimum degree column ordering. The results in Table 3 are obtained using the natural column ordering as the tie breaker and do not vary substantially if Duff's algorithm is used instead as the tie breaker. Only a small improvement in opcount using the Duff tie breaker is obtained for most of the larger rectangular test matrices; the test matrices that did not show improvement were the four supplied by Ashkenazi that all have the nonzeros banded about the principal diagonal.

4. Finite element application. A mixed finite element application for a diffusion problem is presented here [1]. For these applications the matrices do not necessarily have full rank and chance cancellation occurs frequently. The diffusion problem is a four-square unit square centered at the origin and subjected to uniform boundary flux in the positive y direction. There is no lateral boundary flux. The domain is split into elements equally in both directions. The number of elements n_e varies from 16 to 900. The major part of the solution involves a QR decomposition of the gradient matrix. The problem is rank deficient due to the presence of a constant mode. Table 6 shows the characteristics of some of the gradient matrices.

Determination of rank using the QR decomposition is achieved by counting the number of nonzero diagonal entries in R. However, roundoff may cause zero entries to be nonzero. In practice, the diagonal elements are assumed to be zero if they are less than a prescribed tolerance times the norm of the column. A more sophisticated rank determination algorithm using the QR decomposition is reported by Bischof and Hansen [15].

It is also important to determine if individual entries are zero or nonzero. A simple tolerance test will not suffice since the entries tend to become smaller as the decomposition proceeds and more nonzero entries occur in each column. This is because the orthogonal transformation does not change the norm of the column and thus intermediate fill-in requires that the magnitude of other entries be reduced. In performing the Givens rotation on a pair of nonzero entries, the resulting entries are compared. If the magnitude of the smaller entry is less than a tolerance times the

TABLE 4
Execution times (seconds).

Matrix	Row strategy	Column strategy: Duff Overhead	Duff Total	Minimum degree Overhead	Minimum degree Total
	DUFF	0.33		0.49	
ASH219	GENT	0.28		0.32	
	VPAIR	0.39		0.44	
	DUFF	1.05	1.26	1.21	1.43
ABB313	GENT	0.83	1.21	0.99	1.15
	VPAIR	1.21	1.59	1.26	1.48
	DUFF	0.83		1.59	
ASH331	GENT	0.60		0.77	
	VPAIR	0.93		0.82	
	DUFF	4.61		9.12	
ASH608	GENT	1.92		3.57	
	VPAIR	2.36		2.25	
	DUFF	10.87	13.35	15.71	18.83
LSQUAR15	GENT	11.09	14.83	14.33	17.47
	VPAIR	5.38	6.64	4.73	5.38
	DUFF	11.70		19.17	
ASH958	GENT	4.28		6.59	
	VPAIR	4.28		4.01	
	DUFF	3.41	4.28	12.91	15.38
WELL1033	GENT	2.03	2.80	7.58	9.01
	VPAIR	1.65	2.15	6.10	7.20
	DUFF	66.24	79.59	91.40	106.83
LSQUAR20	GENT	64.43	81.18	84.81	98.26
	VPAIR	17.63	25.87	16.53	18.45
	DUFF	5.50	6.53	14.22	15.05
NETWORK4	GENT	7.19	8.96	17.52	19.01
	VPAIR	6.31	7.80	11.97	13.46
	DUFF	6.81	7.53	10.44	10.98
NETWORK3	GENT	5.44	6.60	10.82	11.76
	VPAIR	5.00	5.71	6.92	7.19
	DUFF	36.58	37.90	133.36	150.22
WELL1850	GENT	13.35	16.59	49.93	54.98
	VPAIR	8.90	11.87	29.50	32.58

magnitude of the larger entry, then the smaller entry is assumed to be zero.

The a priori column ordering is obtained through nested domain decomposition as illustrated in [5, Fig. 2.6] and the only exception to this ordering occurs when a zero or small final pivot is obtained. Then, the appropriate column is shifted to the right and all the columns in between are shifted to the left. The sparsity of R (ignoring

TABLE 5
Comparison of results for test matrices.

Column ordering	Liu MD	Ostrouchov MD + Duff	Robey and Sulsky MD + Duff
Row ordering	Liu	DUFF	VPAIR
		ABB313	
Nonzeros in R	1627	1630	1630
Givens	n/a	2596	2047
Opcount	56,044	17,112	11,585
		LSQUAR15	
Nonzeros in R	2786	n/a	2971
Givens	n/a	n/a	5624
Opcount	167,004	n/a	38,332
		WELL1033	
Nonzeros in R	2575	2589	2589
Givens	n/a	7242	4913
Opcount	234,056	46,651	22,363
		LSQUAR20	
Nonzeros in R	6118	n/a	6688
Givens	n/a	n/a	12,187
Opcount	418,444	n/a	110,579
		NETWORK4	
Nonzeros in R	8300	n/a	8368
Givens	n/a	n/a	10,692
Opcount	395,088	n/a	65,584
		NETWORK3	
Nonzeros in R	4091	n/a	4123
Givens	n/a	n/a	9947
Opcount	361,340	n/a	46,499
		WELL1850	
Nonzeros in R	7410	7416	7414
Givens	n/a	26,355	12,839
Opcount	754,444	302,271	91,573

n/a = Results not available

rank deficiency and numerical roundoff) is fixed by this column ordering.

The three local row ordering strategies of Duff (DUFF) [4], Gentleman (GENT) [7] and the variable pair strategy (VPAIR) are implemented and compared. Also, the a priori row ordering induced by nested domain decomposition (NDD) [5] is implemented. NDD on this problem has a theoretical bound of $O(n_e^{3/2})$ on flops due to the underlying grid. The results are shown as log–log plots in Figs. 2, 3, and 4. Table 7 shows that there are substantial differences in performance on a 900-element problem.

The variable pair strategy outperforms the fixed pivot row method and Gentleman's variable pivot strategy as measured by flops, number of Givens rotations, and the amount of intermediate fill-in with better performance as the number of elements increases. In Fig. 2, the observed growth in the QR flops is $O(n_e^{1.57})$ for the VPAIR

TABLE 6
Characteristics of the gradient matrices.

Elements in each direction	Total elements	Rows	Columns	Nonzeros	Sparsity (%)
5	25	150	36	400	7.4
10	100	600	121	1,600	2.2
15	225	1350	256	3,600	1.0
20	400	2400	441	6,400	.60
25	625	3750	676	10,000	.39
30	900	5400	961	14,400	.28

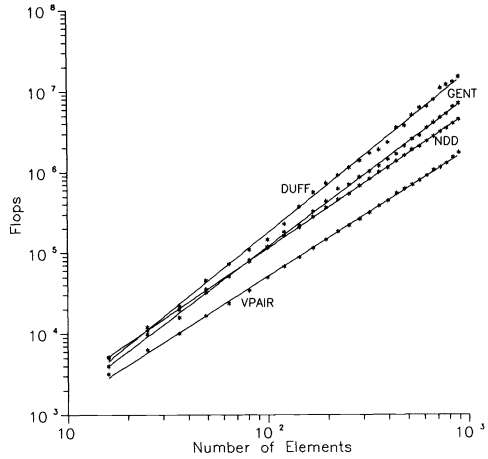


FIG. 2. *QR flops.*

strategy, versus $O(n_e^{1.85})$ for GENT and $O(n_e^{1.99})$ for DUFF over this size range for this problem. The number of flops, intermediate fill-in, and Givens rotations are also less for VPAIR than for NDD. The observed growth in flops for NDD is $O(n_e^{1.68})$; slightly larger than for VPAIR.

Since the row ordering for the nested domain decomposition is obtained as a side benefit of the column ordering, there is no overhead required for row ordering using NDD. To gauge the overhead using VPAIR, r and M (c.f. §2) can be monitored. The maximum r and maximum M for each size problem are used to obtain an observed bound on row ordering overhead of $O(n_e^{1.64})$. Thus, on this problem, the observed row ordering overhead for VPAIR grows faster than the observed growth in flops for VPAIR, but not as fast as the observed growth in flops for NDD.

5. Conclusion. The QR decomposition is a useful solution technique in some applications where Gaussian elimination is not suitable. The QR decomposition allows rank determination in practice and is numerically more stable than Gaussian elimination. Since the QR decomposition is generally more expensive than Gaussian elimination, it must be carefully implemented to be competitive as a general solution technique. An efficient algorithm has been presented for large, sparse matrices that often arise in applications.

A strategy for ordering rows is developed and compared with several other row

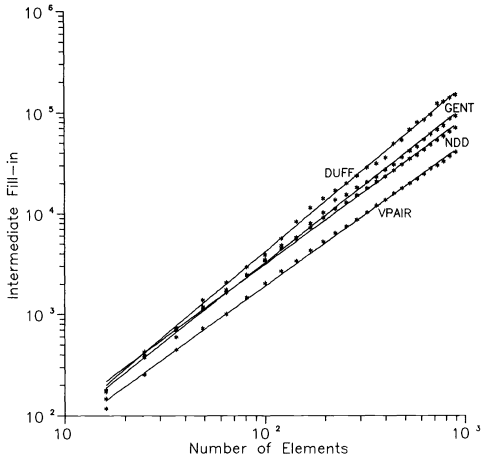


FIG. 3. *Intermediate fill-in.*

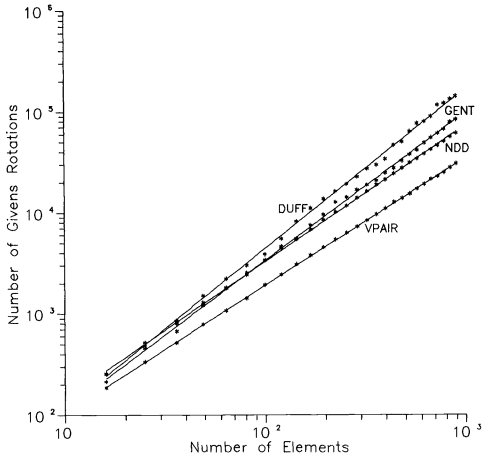


FIG. 4. *Givens rotations.*

ordering strategies in conjunction with several column ordering strategies. The new strategy is that of minimizing local fill-in for a variable pair of rows. For each active submatrix, the order of magnitude overhead to perform the next Givens rotation is roughly the same for the variable pair strategy as for Duff's [4] fixed pivot row ordering strategy. The new strategy also can be used to augment coarse row ordering strategies from nested domain decomposition (NDD) or nested dissection.

For test matrices from the Harwell–Boeing sparse matrix collection [8] and test matrices from George, Heath, and Ng [9], Duff's [4] column ordering and the minimum degree column ordering [3] are used. QR flops, intermediate fill-in, and Givens rotations all show a 33–71% median improvement for the variable pair strategy over Duff's fixed pivot strategy for the larger rectangular test matrices. For the same test matrices, the observed total overhead is much less for the new strategy than for Duff's fixed pivot row ordering strategy. Since the sparsity of R is about the same, the low cost to compute a solution using back substitution is maintained.

Although Duff [4] concluded that his fixed pivot strategy performed similarly to Gentleman's [7] variable pivot strategy, the results in §3 suggest that on these test problems, the issue is not clear cut. Gentleman's strategy performs as well or better than Duff's strategy in some cases, and moreover, requires less row ordering overhead

TABLE 7
Comparison of row orderings for a mixed finite element problem.

Number of elements	Row strategy	Intermed. fill-in	Givens rotations	QR flops	Nonzeros in R
900	DUFF	151,159	141,816	15,103,219	19,667
	GENT	93,295	83,566	7,110,751	19,667
	NDD	71,225	61,202	4,498,832	19,667
	VPAIR	41,006	30,563	1,754,660	19,667

in the process. However, Gentleman's scheme generally does not perform as well as VPAIR on these problems. While it is as yet not possible to definitively state that the variable pair strategy will perform better than the other row ordering strategies on a broad class of problems, the results reported here show that significant improvements are possible. Also, many current applications generate much larger matrices than those considered here and VPAIR appears to have an advantage over other methods on general, large sparse matrices.

Row ordering strategies are also compared on a finite element application where nested domain decomposition [5] is used to determine the column ordering. Comparing the variable pair strategy to the schemes of Duff and Gentleman, a very dramatic reduction is seen in magnitude and growth of QR flops, intermediate fill-in, and the number of Givens rotations, while the sparsity of R is identical. The variable pair strategy can be utilized as a fine-grained row ordering strategy in conjunction with the coarse row ordering of the NDD, although the results indicate that the variable pair strategy also performs well by itself. Furthermore, the variable pair strategy is expected to outperform nested dissection on problems without an underlying grid.

Acknowledgments. The authors thank John Gilbert and two anonymous referees for comments that improved this article. The authors also wish to thank George Ostrouchov for graciously providing data for Table 5 and Joseph Liu for providing some of the test matrices.

REFERENCES

[1] T. H. ROBEY, *The primal mixed finite element method and the* LBB *condition,* Numerical Methods for Partial Differential Equations, 8 (1992), pp. 357–379.

[2] A. GEORGE AND M. T. HEATH, *Solution of sparse linear least squares problems using Givens rotations,* Linear Algebra Appl., 34 (1980), pp. 69–83.

[3] G. OSTROUCHOV, *Symbolic Givens reduction and row-ordering in large sparse least squares problems,* SIAM J. Sci. Statist. Comput., 8 (1987), pp. 248–264.

[4] I. S. DUFF, *Pivot selection and row ordering in Givens reduction on sparse matrices,* Computing, 13 (1974), pp. 239–248.

[5] A. GEORGE AND E. NG, *On row and column orderings for sparse least squares problems,* SIAM J. Numer. Anal., 20 (1983), pp. 326–344.

[6] J. W. H. LIU, *On general row merging schemes for sparse Givens transformations,* SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1190–1211.

[7] W. M. GENTLEMAN, *Row elimination for solving sparse linear systems and least squares problems,* Lecture Notes in Mathematics, 506, Springer-Verlag, New York, 1975, pp. 122–133.

[8] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Users' guide for the Harwell–Boeing sparse matrix collection,* 1988.

[9] A. GEORGE, M. T. HEATH, AND E. NG, *A comparison of some methods for solving sparse linear least-squares problems,* SIAM J. Sci. Statist. Comput., 4 (1983), pp. 177–187.

[10] A. R. CURTIS AND J. K. REID, *The solution of large sparse unsymmetric systems of linear equations,* J. Inst. Math. Appl., 8 (1971), pp. 344–353.

[11] R. A. WILLOUGHBY, *Sparse matrix algorithms and their relation to problem classes and computer architecture,* in Large Sparse Sets of Linear Equations, J. K. Reid, ed., Academic Press, London, New York, 1971, pp. 255–277.

[12] I. S. DUFF AND J. K. REID, *A comparison of some methods for the solution of sparse overdetermined systems of linear equations,* J. Inst. Math. Appl., 17 (1976), pp. 267–280.

[13] G. OSTROUCHOV, *Symbolic Givens Reduction in Large Sparse Least Squares Problems*, Tech. report ORNL-6102, Oak Ridge National Laboratory, Oak Ridge, TN, 1984.

[14] G. OSTROUCHOV, private communication, January 1992.

[15] C. H. BISCHOF AND P. C. HANSEN, *Structure-preserving and rank-revealing QR-factorizations,* Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 1989.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1226–1255, October 1994

012

BLOCK-TRIANGULARIZATIONS OF PARTITIONED MATRICES UNDER SIMILARITY/EQUIVALENCE TRANSFORMATIONS*

HISASHI ITO†, SATORU IWATA‡, AND KAZUO MUROTA‡

Abstract. A partitioned matrix, of which the column- and row-sets are divided into certain numbers of groups, arises from a mathematical formulation of discrete physical or engineering systems. This paper addresses the problem of the block-triangularization of a partitioned matrix under similarity/equivalence transformation with respect to its partitions of the column- and row-sets. Such block-triangularization affords a mathematical representation of the hierarchical decomposition of a physical system into subsystems if the transformation used is of physical significance. A module is defined from a partitioned matrix, and the simplicity of the module is proved to be equivalent to the nonexistence of a nontrivial block-triangular decomposition. Moreover, the existence and the uniqueness of the block-triangular forms are deduced from the Jordan–Hölder theorem for modules. The results cover many block-triangularization methods hitherto discussed in the literature such as the Jordan normal form and the strongly connected-component decomposition in the case of partition-respecting similarity transformations, and the rank normal form, the Dulmage–Mendelsohn decomposition, and the combinatorial canonical form of layered mixed matrices in the case of partition-respecting equivalence transformations.

Key words. block-triangular form, Jordan–Hölder theorem, modular lattice, partitioned matrix

AMS subject classifications. 15A21, 15A30

1. Introduction.

In the analysis of a discrete physical/engineering system generally it is all important to recognize "subsystems" and "hierarchy" among them in an appropriate manner. When the system is described in terms of a matrix (of any kind), the hierarchical decomposition of the whole system into partially ordered subsystems is reduced to the block-triangularization of the matrix under a suitably chosen class of "admissible transformations." To be more precise, the block-triangularization of the matrix is almost tantamount to finding a good or "canonical" mathematical description of the system, which hopefully reveals the "subsystems" and the "hierarchy" of physical significance.

In this paper it is assumed that the matrix, say A, in question is a partitioned matrix of which the row set and the column set are divided into a certain number of groups, respectively. Two types of admissible transformations are considered: (i) similarity transformations that respect or conform with the partition structure

$$\begin{bmatrix} S_1 & O & \cdots & O \\ O & S_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ O & \cdots & O & S_\nu \end{bmatrix}^{-1} \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1\nu} \\ A_{21} & A_{22} & \cdots & A_{2\nu} \\ \vdots & \vdots & \ddots & \vdots \\ A_{\nu 1} & A_{\nu 2} & \cdots & A_{\nu\nu} \end{bmatrix} \begin{bmatrix} S_1 & O & \cdots & O \\ O & S_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ O & \cdots & O & S_\nu \end{bmatrix},$$

and (ii) equivalence transformations that conform to the partition structure that is defined similarly (see §§2.1 and 3.1 for the precise definitions). In this paper the transfor-

* Received by the editors August 5, 1992; accepted for publication June 29, 1993. This research was performed at the Department of Mathematical Engineering and Information Physics at the University of Tokyo.

† Department of Computer Science and Information Mathematics, The University of Electro-Communications, Tokyo 182, Japan (his@lit.cs.uec.ac.jp).

‡ Research Institute for Mathematical Sciences, Kyoto University, Kyoto 606, Japan.

mations of types (i) and (ii) are called PS-transformations and PE-transformations, respectively. The main objective of this paper is to investigate the block-triangularization of A under such transformations by means of the module theoretical framework. In particular, the Jordan–Hölder theorem for modules plays the key role in discussing the existence and the uniqueness of the block-triangular form.

The present problem formulation covers many important instances of the hierarchical decomposition discussed so far in the literature as briefly mentioned in the following.

When a general similarity transformation ($S^{-1}AS$ with S nonsingular) is allowed, the Jordan normal form yields a block-triangular decomposition, whereas when a general equivalence ($S_{\mathrm{r}}AS_{\mathrm{c}}$ with S_{r} and S_{c} nonsingular) is admissible, the rank normal form serves as a block-triangular decomposition. Needless to say, these normal forms play fundamental roles in many application areas.

In the design and the analysis of large-scale engineering systems it is often meaningful to consider a most-restricted class of similarity/equivalence transformations, namely, permutations of the rows and the columns. When the rows and the columns are permuted simultaneously as $P^{\mathrm{T}}AP$ with P a permutation matrix, a block-triangularization is given by the decomposition of the directed graph associated with A into strongly connected components; see, e.g., Aho, Hopcroft, and Ullman [1]. On the other hand, when the rows and the columns are permuted independently as $P_{\mathrm{r}}AP_{\mathrm{c}}$ with P_{r} and P_{c} permutation matrices, a block-triangularization is given by the canonical decomposition of the bipartite graph associated with A into elementary or irreducible components. This decomposition is due to Dulmage and Mendelsohn [10] (see, also, Brualdi and Ryser [8], Lovász and Plummer [19]) and is often referred to as the DM-decomposition. Both the strongly connected-component decomposition and the DM-decomposition can be computed efficiently by purely graph-theoretical algorithms.

The underlying mathematical structure of the DM-decomposition can be understood in a more abstract context of matroids such as the Jordan–Hölder type theorem for submodular functions (Iri [15]). With the aid of this general principle, the DM-decomposition is extended to the combinatorial canonical form (CCF) for a layered mixed matrix that is proposed as a mathematical tool for describing the combinatorial structure of physical systems (Murota [20], Murota [22], Murota, Iri, and Nakamura [23]; see Remark 3.2 in §3.3 for more about this). A layered mixed matrix is "mixed" in the sense that it consists of two different kinds of entries, independent variables and fixed constants, and the CCF combines the combinatorial technique for the DM-decomposition and the standard linear algebraic technique for the LU-decomposition (or Gaussian elimination). The CCF can be computed by an efficient algorithm that makes use of the matroid-theoretical algorithm for submodular flows. In this way, combinatorial mathematics can provide us with effective decomposition methods, often supported by efficient algorithms, when we want to exploit the combinatorial structure such as sparsity, independence of individual element characteristics, and dependence stemming from incidence relations. See Iri [14], Murota [20], Recski [25], and Šiljak [27] for applications of graph, network, and matroid theories to the design and the analysis of engineering systems.

In the area of physics and chemistry, especially in quantum physics and chemistry, it is standard technique to decompose a system with respect to the symmetry of the system. The underlying mathematical structure for this technique can be explained

by the group representation theory [9]; namely, the decomposition corresponds to the decomposition of a group representation into homogeneous or irreducible components. An attempt is made in Murota [21] to combine the symmetry-exploiting decomposition method with the matroid-theoretical decomposition method. It may be noted that the group representation theory can be embraced in the more general theory of modules.

The module-theoretical approach has turned out to be useful also in analyzing and decomposing automata, dynamical systems, and stochastic processes. In automata theory any finite machine can be decomposed into loop-free connections of permutation reset machines; the Jordan–Hölder theorem for groups [7] yields a further decomposition of a permutation reset machine (the Krohn–Rhodes theorem; see Arbib [3]). In mathematical systems theory a finite module over polynomials plays an important role, and the structure theorem on such a module gives the hierarchical decomposition of a dynamical system (see Kalman, Falb, and Arbib [18]). A stochastic process can also be formulated in terms of a module, called a stochastic module, as shown by Heller [12], [13]; then the Krull–Remak–Schmidt theorem gives a decomposition of a stochastic process.

The admissible similarity transformation (PS-transformation) introduced above has a close relation to a certain type of stochastic process called the hidden Markov process. It has recently been shown by Ito, Amari, and Kobayashi [17] that (roughly speaking) two Markov chains generate the same hidden Markov process if and only if the two transition matrices are connected by a PS-transformation. (This solves the identifiability problem of hidden Markov processes posed by Blackwell and Koopmans [6]; see also Rosenblatt [26].) This result, when combined with Heller's framework of stochastic modules, leads to the observation of Ito [16] that the PS-transformation can be formulated in a module-theoretical framework. The present paper adopts this line of approach to investigate the block-triangularization under PS- or PE-transformations.

The outline of this paper is as follows. In §2 we consider the block-triangularization of a partitioned square matrix under PS-transformations and show the existence and the uniqueness of PS-irreducible components by the Jordan–Hölder theorem for modules. We also discuss the ordering among PS-irreducible components to explain the "hierarchy" revealed by the block-triangularization. In §3 we consider the block-triangularization (with rank conditions) of a partitioned, not necessarily square, matrix under PE-transformations. The uniqueness of PE-irreducible components and the ordering among them will be derived similarly although such a block-triangularization does not always exist. We also exploit another approach using a submodular function to explain a necessary and sufficient condition for the existence of a block-triangularization with rank conditions and the relationship of our new block-triangularization to the previously known decompositions such as the DM-decomposition and the CCF. In §4 we discuss an algorithm for block-triangularization of a partitioned square matrix in a nondegenerate case and also an algorithm to check the existence of a block triangularization under PE-transformations.

2. Decomposition under partition-respecting similarity.

2.1. Problem formulation. In this section we discuss block-triangularization of a partitioned *square* matrix under restricted similarity transformations that preserve the partition structure. More precisely, let F be a field, e.g., the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$, and A be an n-dimensional square matrix over the field

F with given partitions of rows and columns:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1\nu} \\ A_{21} & A_{22} & \cdots & A_{2\nu} \\ \vdots & \vdots & \ddots & \vdots \\ A_{\nu 1} & A_{\nu 2} & \cdots & A_{\nu\nu} \end{bmatrix},$$

where $A_{\alpha\beta}$ is an $n_\alpha \times n_\beta$ matrix called (α, β)-submatrix of A. To describe the partition structure in terms of matrix operation, we introduce a family of $n \times n$ projection matrices Π_α $(\alpha = 1, \ldots, \nu)$, where the (α, α)-submatrix of Π_α is the unit matrix I_{n_α} of dimension n_α and the other submatrices are zeros. Sometimes we denote the partitioned square matrix by the pair (A, Π), where $\Pi = \{\Pi_\alpha\}_{\alpha=1}^{\nu}$. A transformation $\widetilde{A} := S^{-1}AS$ of A is said to be a *partition-respecting similarity transformation* (PS-transformation) if the off-diagonal submatrices of S are zeros, i.e.,

$$\Pi_\alpha S = S\Pi_\alpha \quad \text{for} \quad \alpha = 1, \ldots, \nu.$$

We say that A is *Π-similar* to A' if there exists a PS-transformation that transforms A to A'. Then our problem is to bring A into a block-triangular form by means of a PS-transformation.

Alternatively, our problem can be formulated as follows. Let V be an n-dimensional F-linear space F^n, which is a direct sum of lower dimensional spaces:

$$V = \bigoplus_{\alpha=1}^{\nu} V_\alpha, \qquad \dim_F V_\alpha = n_\alpha.$$

A linear transformation

$$f : V \to V$$

is constructed by a family of linear transformations

$$f_{\alpha\beta} : V_\beta \to V_\alpha \quad (1 \leq \alpha,\ \beta \leq \nu).$$

When we take arbitrary basis for each space V_α, f is represented by a partitioned square matrix A, where $A_{\alpha\beta}$ corresponds to $f_{\alpha\beta}$. In this context PS-transformations are naturally derived as basis transformations of V_α.

It may be in order here to give a definition to the notion of a block-triangular form. Let $\widetilde{A}$ be a square matrix representing a linear transformation from a linear space V to itself. Then the row set $R = \{1, 2, \ldots, n\}$ is to be identified with the column set $C = \{1, 2, \ldots, n\}$. For $R_* \subseteq R$ and $C_* \subseteq C$, let $\widetilde{A}[R_*, C_*]$ mean the submatrix of $\widetilde{A}$ with row set R_* and column set C_*. When we split R into a certain number of disjoint nonempty blocks $(R_1, \ldots, R_b)$, C is naturally split into blocks $(C_1, \ldots, C_b)$, where

$$R_k = C_k \quad \text{for} \quad k = 1, \ldots, b$$

according to the correspondence between R and C. We say that $\widetilde{A}$ is in a *block-triangular form* or is *block-triangularized* with respect to the blocks $(R_1, \ldots, R_b)$ when $\widetilde{A}$ satisfies the following condition:

$$\widetilde{A}[R_k, C_l] = O \quad \text{if} \quad 1 \leq l < k \leq b.$$

If $\widetilde{A}$ is block-triangularized in the above sense, it is clear that we can put it into an explicit upper block-triangular form $\bar{A} = P^{\mathrm{T}}\widetilde{A}P$ in the usual sense by using a certain permutation matrix P, where the superscript T denotes the transpose of a matrix.

A partitioned square matrix (A, Π), or simply A, is said to be PS-*irreducible* if it can never be transformed into a block-triangular form with two or more nonempty blocks by any PS-transformation. If $\widetilde{A}$ is a block-triangular matrix obtained from A by a PS-transformation and, in addition, all the diagonal blocks $\widetilde{A}[R_k, C_k]$ for $k = 1, \ldots, b$ are PS-irreducible, we say that $\widetilde{A}$ is a (PS-)*irreducible decomposition* of A, and $\widetilde{A}[R_k, C_k]$ are (PS-)*irreducible components* of A.

Our problem of block-triangularization above includes two well-known extreme cases. The first case is with the trivial partition structure: $\nu = 1$. In this case our problem has been completely solved by the Jordan normal form (in the case of $F = \mathbb{C}$) [11]. The other extreme case is with the finest partition structure: $\nu = n$. In this case our problem is essentially solved by the decomposition of a directed graph into strongly connected components with efficient algorithms [4].

On the analogy of strongly connected-component decompositions, a partial order is induced among the blocks $\{C_k \mid k = 1, \ldots, b\}$ in a natural manner by the zero/nonzero structure of a block-triangular matrix $\widetilde{A}$. The partial order $\preceq$ is the reflexive and transitive closure of the relation defined by: C_k is "smaller" than or equal to C_l if $\widetilde{A}[R_k, C_l] \neq O$, where R_k is identified with C_k for $k = 1, \ldots, b$. We denote this poset $(\{C_k\}_{k=1}^{b}, \preceq)$ by $\mathcal{P}(\widetilde{A})$.

As is well known [2], [5], the ideals of the poset $\mathcal{P}(\widetilde{A})$ constitute a distributive lattice, which we denote by $\mathcal{D}(\widetilde{A})$. Note that a subset J of C can be naturally identified with a subspace of V, which we denote by $\psi(J)$, i.e.,

$$\psi(J) = \mathrm{span}\{(0, \ldots, 0, \overset{j}{\overset{\vee}{1}}, 0, \ldots, 0)^{\mathrm{T}} \mid j \in J\}.$$

Then

$$\psi(J_1 \cup J_2) = \psi(J_1) + \psi(J_2)$$

and

$$\psi(J_1 \cap J_2) = \psi(J_1) \cap \psi(J_2),$$

and hence

$$\psi(\mathcal{D}(\widetilde{A})) = \{\psi(J) \mid J \in \mathcal{D}(\widetilde{A})\}$$

is a distributive sublattice of the modular lattice formed by the subspaces of V.

We say that $\widetilde{A} = S^{-1}AS$ is *finer*, as a decomposition of A, than $\widetilde{A}' = S'^{-1}AS'$ if $S'\psi(\mathcal{D}(\widetilde{A}')) = \{S'\widetilde{W}' \mid \widetilde{W}' \in \psi(\mathcal{D}(\widetilde{A}'))\}$ is a proper sublattice of $S\psi(\mathcal{D}(\widetilde{A})) = \{S\widetilde{W} \mid \widetilde{W} \in \psi(\mathcal{D}(\widetilde{A}))\}$. Furthermore, we say that $\widetilde{A}$, Π-similar to A, is a *finest-possible* decomposition of (A, Π) if there does not exist a block-triangular matrix $\widetilde{A}'$ that is also Π-similar to A and finer than $\widetilde{A}$.

Example 2.1. Consider the following five-dimensional partitioned square matrix

A over $F = \mathbb{Q}$:

$$A = \begin{array}{c} \\ \circ \\ \circ \\ \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \circ & \bullet & \bullet & \bullet \end{array} \\ \left[\begin{array}{cc|ccc} 3 & 2 & -2 & 5 & 13 \\ 0 & 2 & 1 & 0 & 1 \\ \hline 0 & -1 & 6 & 1 & 0 \\ 4 & 8 & 0 & 2 & -6 \\ -2 & -4 & 0 & 1 & 7 \end{array}\right] \end{array},$$

where $\nu = 2$, $n_1 = 2$, and $n_2 = 3$. Let S be the nonsingular matrix

$$S = \left[\begin{array}{cc|ccc} 2 & -1 & \multicolumn{3}{c}{\multirow{2}{*}{O}} \\ -1 & 1 & & & \\ \hline \multicolumn{2}{c|}{\multirow{3}{*}{O}} & 1 & 0 & 0 \\ & & 0 & 3 & -2 \\ & & 0 & -1 & 1 \end{array}\right].$$

Then we have

$$\widetilde{A} = S^{-1}AS = \begin{array}{c} \\ \circ \\ \circ \\ \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \circ & \bullet & \bullet & \bullet \end{array} \\ \left[\begin{array}{cc|ccc} 2 & 1 & -1 & 1 & 4 \\ 0 & 3 & 0 & 0 & 5 \\ \hline 1 & -1 & 6 & 3 & -2 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & -2 & 0 & 0 & 5 \end{array}\right] \end{array}.$$

By using the permutation matrix

$$P = \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

we can transform $\widetilde{A}$ into an explicit upper block-triangular form

$$\bar{A} = P^{\mathrm{T}}\widetilde{A}P = \begin{array}{c} \\ \circ \\ \bullet \\ \bullet \\ \circ \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \bullet & \bullet & \circ & \bullet \end{array} \\ \left[\begin{array}{c|c|c|c|c} 2 & -1 & 1 & 1 & 4 \\ \hline 1 & 6 & 3 & -1 & -2 \\ \hline & & 4 & \multicolumn{2}{c}{\multirow{2}{*}{O}} \\ \cline{3-3} & \multicolumn{2}{c|}{\multirow{3}{*}{O}} & & \\ \cline{4-5} & & & 3 & 5 \\ & & & -2 & 5 \end{array}\right] \end{array}$$

with three diagonal blocks

$$D_1 = \widetilde{A}[C_1, R_1] = \left[\begin{array}{c|c} 2 & -1 \\ \hline 1 & 6 \end{array}\right], \qquad D_2 = \widetilde{A}[C_2, R_2] = [4],$$

$$D_3 = \widetilde{A}[C_3, R_3] = \left[\begin{array}{c|c} 3 & 5 \\ \hline -2 & 5 \end{array}\right].$$

Note that D_k's are also partitioned square matrices with partitions defined by

$$\Phi_1 = \left\{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right\}, \quad \Phi_2 = \{[0], [1]\}, \quad \Phi_3 = \left\{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right\}.$$

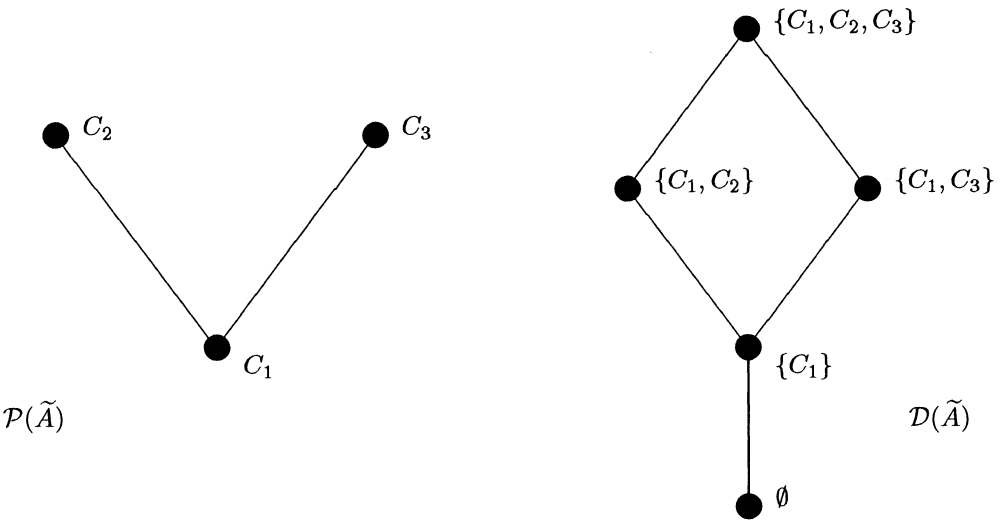


FIG. 1. *The partial order $\mathcal{P}(\widetilde{A})$ and the distributive lattice $\mathcal{D}(\widetilde{A})$ in Example* 2.1.

Since (D_1, Φ_1), (D_2, Φ_2), and (D_3, Φ_3) are PS-irreducible, $\widetilde{A}$ is an irreducible decomposition of A. Furthermore, it turns out to be a finest-possible decomposition of A. The Hasse diagrams for $\mathcal{P}(\widetilde{A})$ and $\mathcal{D}(\widetilde{A})$ are illustrated in Fig. 1.

The main results of this section are as follows. In §2.3 we define a module $M(A, \Pi)$ from a partitioned square matrix (A, Π). It is shown that the PS-irreducibility is equivalent to the simplicity of this module (see Theorem 2.3) and that the PS-irreducible components are determined uniquely (see Theorem 2.4) by the Jordan–Hölder theorem for modules. In §2.4, it is shown that a finest-possible decomposition of (A, Π) is not unique, but is obtained by a maximal distributive sublattice of the modular lattice $\mathcal{L}(A, \Pi)$ of the submodules of $M(A, \Pi)$ (see Theorem 2.7).

2.2. Jordan–Hölder theorem for modules. In this section we prepare a general framework to investigate partitioned matrices by means of module theory. Let M be an n-dimensional F-linear space, Σ be a finite set of symbols, and Σ^* the free monoid generated by Σ, where the unity is the empty string Λ. We can extend the monoid Σ^* to a monoid algebra

$$F\langle\Sigma^*\rangle = \{\sum_{s\in\Sigma^*} c_s s \mid c_s \in F, \text{ and } c_s = 0 \text{ for all } s \in \Sigma^* \text{ but finite } s\}$$

over F.

When we have an action of Σ^* on M, which satisfies the conditions:

$$\begin{aligned} \Lambda x &= x, & x &\in M, \\ s(x+y) &= sx + sy, & s &\in \Sigma^*, \quad x, y \in M, \\ (st)x &= s(tx), & s, t &\in \Sigma^*, \quad x \in M, \end{aligned}$$

it is quite natural to consider the n-dimensional F-linear space M to be a left $F\langle\Sigma^*\rangle$-module.

Let $\mathcal{R}$ be an arbitrary ring, $\mathcal{M}$ an $\mathcal{R}$-module, and

$$\mathcal{C} : 0 = \mathcal{M}_0 \subsetneq \mathcal{M}_1 \subsetneq \cdots \subsetneq \mathcal{M}_h = \mathcal{M}$$

a strictly increasing finite chain of $\mathcal{R}$-submodules of $\mathcal{M}$. We say that the *length* of $\mathcal{C}$ is h. $\mathcal{C}$ is called a *composition series* of $\mathcal{M}$ if each factor module $\mathcal{M}_i/\mathcal{M}_{i-1}$ $(i = 1, 2, \ldots, h)$ has no nontrivial submodules. The $F\langle\Sigma^*\rangle$-module M has a composition series, since it is also an n-dimensional F-linear space. The following Jordan–Hölder theorem [9] plays a crucial role not only in this section, but also in §3.

THEOREM 2.1 (Jordan–Hölder [7], [9]). *If*

$$\mathcal{C} : 0 = \mathcal{M}_0 \subsetneq \mathcal{M}_1 \subsetneq \cdots \subsetneq \mathcal{M}_h = \mathcal{M}$$

and

$$\mathcal{C}' : 0 = \mathcal{M}'_0 \subsetneq \mathcal{M}'_1 \subsetneq \cdots \subsetneq \mathcal{M}'_{h'} = \mathcal{M}$$

are any two composition series of $\mathcal{R}$-module $\mathcal{M}$, then $\mathcal{C}$ is equivalent to $\mathcal{C}'$, which means that $h = h'$ and there exists a permutation σ of $\{1, 2, \ldots, h\}$, such that

$$\mathcal{M}_i/\mathcal{M}_{i-1} \cong \mathcal{M}'_{\sigma(i)}/\mathcal{M}'_{\sigma(i)-1}.$$

2.3. Decomposition into irreducible components. Given a partitioned square matrix (A, Π), we consider an $F\langle\Sigma^*\rangle$-module M (see the general framework of §2.2) by setting

$$\Sigma = \{z_0, z_1, \ldots, z_\nu\}$$

and defining the action of Σ^* on V as follows:

$$\left.\begin{array}{l} z_0 x := Ax, \\ z_\alpha x := \Pi_\alpha x \quad (\alpha = 1, \ldots, \nu) \end{array}\right\} \quad x \in V.$$

Then $M = V$ becomes an $F\langle\Sigma^*\rangle$-module as well as an n-dimensional F-linear space. We denote this module by $M(A, \Pi)$ to emphasize its dependence on the matrix A and the partition structure Π. For another partitioned square matrix A' with common Π, we obtain another $F\langle\Sigma^*\rangle$-module $M(A', \Pi)$. The following lemma states that the module $M(A, \Pi)$ remains isomorphic under PS-transformations.

LEMMA 2.2. *$M(A, \Pi) \cong M(A', \Pi)$ if and only if A is Π-similar to A'.*

Proof. ($\Longrightarrow$) Let $\varphi : M(A, \Pi) \to M(A', \Pi)$ be the module isomorphism. Since $F \subseteq F\langle\Sigma^*\rangle$, we can consider the two modules F-linear spaces and φ a bijective linear transformation that is to be represented by a nonsingular matrix S. For all $x \in M(A, \Pi)$, we have

$$\begin{aligned} A'Sx &= z_0\varphi(x) = \varphi(z_0 x) = SAx, \\ \Pi_\alpha Sx &= z_\alpha\varphi(x) = \varphi(z_\alpha x) = S\Pi_\alpha x \quad \text{for } \alpha = 1, \ldots, \nu. \end{aligned}$$

Therefore $A'S = SA$ and $\Pi_\alpha S = S\Pi_\alpha$ for $\alpha = 1, \ldots, \nu$.

($\Longleftarrow$) Suppose that there exists a nonsingular matrix S such that $A'S = SA$ and $\Pi_\alpha S = S\Pi_\alpha$ for $\alpha = 1, \ldots, \nu$. For all $x \in M(A, \Pi)$ and $\xi(z_0, z_1, \ldots, z_\nu) \in F\langle\Sigma^*\rangle$, we have

$$S\xi(z_0, z_1, \ldots, z_\nu)x = S\xi(A, \Pi_1, \ldots, \Pi_\nu)x = \xi(A', \Pi_1, \ldots, \Pi_\nu)Sx = \xi(z_0, z_1, \ldots, z_\nu)Sx.$$

Therefore the morphism $\varphi : M(A, \Pi) \to M(A', \Pi)$ represented by S is a module homomorphism. Since S is nonsingular, φ is bijective. Hence it holds that $M(A, \Pi) \cong M(A', \Pi)$. □

Lemma 2.2 suggests that the decomposition of the partitioned square matrix (A, Π) can be obtained through that of the $F\langle\Sigma^*\rangle$-module $M(A, \Pi)$, which in turn corresponds to the decomposition of the module along a chain. Now we explain the correspondence between an irreducible decomposition of (A, Π) and a composition series of $M(A, \Pi)$. Let $\mathcal{L}(A, \Pi)$ be the modular lattice of the submodules of $M(A, \Pi)$. A composition series of $M(A, \Pi)$ is a maximal chain of $\mathcal{L}(A, \Pi)$.

Let $\mathcal{C}$ be a strictly increasing finite chain of the submodules of $M(A, \Pi)$:

$$\mathcal{C} : 0 = M_0 \subsetneq M_1 \subsetneq \cdots \subsetneq M_h = M(A, \Pi).$$

The $F\langle\Sigma^*\rangle$-submodule M_i is an invariant subspace of A, which satisfies $\Pi_\alpha M_i \subseteq M_i$ for $\alpha = 1, \ldots, \nu$. Since $\sum_{\alpha=1}^{\nu} \Pi_\alpha = I_n$, we have

$$M_i = \bigoplus_{\alpha=1}^{\nu} \Pi_\alpha M_i.$$

We can obtain increasing chains

$$\Pi_\alpha \mathcal{C} : 0 = \Pi_\alpha M_0 \subseteq \Pi_\alpha M_1 \subseteq \cdots \subseteq \Pi_\alpha M_h = V_\alpha$$

for $\alpha = 1, \ldots, \nu$. Let $B_{i\alpha}$ be a set of linearly independent column vectors spanning $\Pi_\alpha M_i$ for $i = 0, \ldots, h$ such that

$$\emptyset = B_{0\alpha} \subseteq B_{1\alpha} \subseteq \cdots \subseteq B_{h\alpha}.$$

Then $B_i = \bigcup_{\alpha=1}^{\nu} B_{i\alpha}$ spans M_i. Order n column vectors of B_h as $[B_{h1}, B_{h2}, \ldots, B_{h\nu}]$ to get a nonsingular matrix $S = \bigoplus_{\alpha=1}^{\nu} S_\alpha$, and put $\widetilde{A} := S^{-1}AS$. Let C_l be the column subset corresponding to $\hat{B}_l = B_l - B_{l-1}$ for $l = 1, \ldots, h$, and R_k be the row subset identified with C_k for $k = 1, \ldots, h$. Since M_i is an invariant subspace of A,

$$\widetilde{A}\left[\bigcup_{k=i+1}^{h} R_k, \bigcup_{l=1}^{i} C_l\right] = O \quad \text{for} \quad i = 1, \ldots, h.$$

Or equivalently,

$$\widetilde{A}[R_k, C_l] = O \quad \text{if} \quad 1 \leq l < k \leq h.$$

That is, $\widetilde{A}$ is in a block-triangular form with respect to $(R_1, \ldots, R_b) = (C_1, \ldots, C_b)$, where the number of blocks b is given by the length h of the chain $\mathcal{C}$. It is also clear that (A, Π) is PS-irreducible if and only if $M(A, \Pi)$ is simple.

Suppose the chain $\mathcal{C}$ is a composition series. Put

$$D_k := \widetilde{A}[R_k, C_k], \qquad \Phi_k := \{\Pi_\alpha[R_k, C_k]\}_{\alpha=1}^{\nu} \quad \text{for } k = 1, \ldots, h,$$

then (D_k, Φ_k) is also a partitioned square matrix. We can construct another $F\langle\Sigma^*\rangle$-module $N_k = M(D_k, \Phi_k)$. Then

$$N_k \cong M_k / M_{k-1}.$$

It follows from the simplicity of M_k/M_{k-1} that D_k is PS-irreducible and $\widetilde{A}$ is an irreducible decomposition of A. Conversely, if $(\widetilde{A}, \Pi)$ is in a block-triangular form with PS-irreducible diagonal blocks, the subspaces $\psi(\bigcup_{k=1}^{i} C_k)$ for $i = 1, \ldots, b$ are submodules that constitute a composition series of $M(\widetilde{A}, \Pi)$. (Recall that $\psi(J)$ denotes the subspace of V that corresponds to $J \subseteq C$.) Thus we have the following theorem.

THEOREM 2.3. *A* PS*-irreducible decomposition of* (A, Π) *is obtained by a composition series of* $M(A, \Pi)$. *In particular* (A, Π) *is* PS*-irreducible if and only if* $M(A, \Pi)$ *is a simple module.*

Now we have already seen how to get an irreducible decomposition, but have not yet discussed the uniqueness. Suppose both $(\widetilde{A}, \Pi)$ and $(\widetilde{A}', \Pi)$ are irreducible decompositions of (A, Π) with blocks $(R_1, \ldots, R_b) = (C_1, \ldots, C_b)$ and $(R'_1, \ldots, R'_{b'}) = (C_1, \ldots, C'_{b'})$, respectively. Put

$$\begin{aligned} D_k &:= \widetilde{A}[R_k, C_k], \quad \Phi_k := \{\Pi_\alpha[R_k, C_k]\}_{\alpha=1}^{\nu} \quad \text{for } k = 1, \ldots, b, \\ D'_l &:= \widetilde{A}'[R'_l, C'_l], \quad \Phi'_l := \{\Pi_\alpha[R'_l, C'_l]\}_{\alpha=1}^{\nu} \quad \text{for } l = 1, \ldots, b'. \end{aligned}$$

Then it follows directly from the Jordan–Hölder theorem (Theorem 2.1) as well as Lemma 2.2 that $b = b'$ and there exists a permutation σ of $\{1, 2, \ldots, b\}$ such that $\Phi_k = \Phi'_{\sigma(k)}$ and D_k is Φ_k-similar to $D'_{\sigma(k)}$. Hence we have the following theorem on the uniqueness of irreducible components of a partitioned square matrix.

THEOREM 2.4. *The set of* PS*-irreducible components of a partitioned square matrix is unique to within* PS*-transformations of each component.*

Example 2.2. Consider the same A as in Example 2.1. Let S' be the nonsingular matrix

$$S' = \left[\begin{array}{cc|ccc} 2 & -1 & & O & \\ -1 & 1 & & & \\ \hline & & 1 & 0 & 0 \\ & O & 0 & -2 & 1 \\ & & 0 & 1 & 0 \end{array}\right].$$

Then we have

$$\widetilde{A}' = S'^{-1} A S' = \begin{array}{c} \\ \circ \\ \circ \\ \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \circ & \bullet & \bullet & \bullet \end{array} \\ \left[\begin{array}{cc|ccc} 2 & 1 & -1 & 4 & 5 \\ 0 & 3 & 0 & 5 & 5 \\ \hline 1 & -1 & 6 & -2 & 1 \\ 0 & -2 & 0 & 5 & 1 \\ 0 & 0 & 0 & 0 & 4 \end{array}\right] \end{array}.$$

By using the permutation matrix

$$P' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

we can transform $\widetilde{A}'$ into an explicit upper block-triangular form

$$\bar{A}' = P'^{\mathrm{T}} \widetilde{A}' P' = \begin{array}{c} \\ \circ \\ \bullet \\ \circ \\ \bullet \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \bullet & \circ & \bullet & \bullet \end{array} \\ \left[\begin{array}{c|c|c|c|c} 2 & -1 & 1 & 4 & 5 \\ \hline 1 & 6 & -1 & -2 & 1 \\ \hline & & 3 & 5 & 5 \\ \cline{3-5} & O & -2 & 5 & 1 \\ \cline{4-5} & & & & 4 \end{array}\right] \end{array}.$$

Then $\widetilde{A}'$ is also an irreducible decomposition with three irreducible components

$$D'_1 = \left[\begin{array}{c|c} 2 & -1 \\ \hline 1 & 6 \end{array}\right], \quad D'_2 = \left[\begin{array}{c|c} 3 & 5 \\ \hline -2 & 5 \end{array}\right], \quad D'_3 = [4]$$

and

$$\Phi'_1 = \left\{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right\}, \quad \Phi'_2 = \left\{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right\}, \quad \Phi'_3 = \{[0], [1]\}.$$

Thus $\Phi_k = \Phi'_{\sigma(k)}$ and D_k is Φ_k-similar to $D'_{\sigma(k)}$ for $k = 1, 2, 3$, where $\sigma = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{smallmatrix}\right)$.

2.4. Ordering among irreducible components. We have already seen that an irreducible decomposition can be obtained by taking the basis along a composition series of $M(A, \Pi)$, and that the irreducible components are uniquely determined in the sense of Theorem 2.4. We have not yet discussed the strictly upper block-triangular part. Our main purpose is to transform (A, Π) into a block-triangular form that makes the hierarchical structure as clear as possible. In this sense an irreducible decomposition is not the final goal. We want to make the upper part of the block-triangular form as simple as possible. To this end we consider the structure of the whole lattice $\mathcal{L}(A, \Pi)$ rather than a single chain (composition series of $M(A, \Pi)$).

Recall the definition of $\psi(\mathcal{D}(\widetilde{A}))$ in §2.1. Then we have the following lemmas.

LEMMA 2.5. *Suppose $\widetilde{A} = S^{-1}AS$ is a* PS*-transformation of A. Then $S\psi(\mathcal{D}(\widetilde{A}))$ is a distributive sublattice of the modular lattice $\mathcal{L}(A, \Pi)$. Moreover,*

$$S\psi(\mathcal{D}(\widetilde{A})) = \{W \in \mathcal{L}(A, \Pi) \mid W \text{ is spanned by a subset of column vectors of } S\}.$$

Proof. Let J be an ideal of $\mathcal{P}(\widetilde{A})$, i.e., $J \in \mathcal{D}(\widetilde{A})$. Then $\Pi_\alpha \psi(J) \subseteq \psi(J)$ for $\alpha = 1, \ldots, \nu$ and $\widetilde{A}\psi(J) \subseteq \psi(J)$ due to the block-triangularity of $\widetilde{A}$. Therefore $\psi(J) \in \mathcal{L}(\widetilde{A}, \Pi)$. On the other hand, $\widetilde{W} \in \mathcal{L}(\widetilde{A}, \Pi)$ if and only if $S\widetilde{W} \in \mathcal{L}(A, \Pi)$. Hence $S\psi(\mathcal{D}(\widetilde{A}))$ is a distributive sublattice of $\mathcal{L}(A, \Pi)$. □

LEMMA 2.6. *For any distributive sublattice $\mathcal{D}'$ of $\mathcal{L}(A, \Pi)$, there exists a* PS*-transformation $\widetilde{A} = S^{-1}AS$ of A such that $S\psi(\mathcal{D}(\widetilde{A})) \supseteq \mathcal{D}'$.*

Proof. According to Birkhoff's representation theorem, we can represent the distributive lattice $\mathcal{D}'$ by the set of ideals of a poset $\mathcal{P}'$. More precisely, let $\mathcal{P}' = (\{W_1, \ldots, W_g\}, \subseteq)$ be a poset that consists of the join-irreducible elements of $\mathcal{D}'$ except zero [2]. We assume $k \leq l$ if $W_k \subseteq W_l$. Let $B_{i\alpha}$ be a set of linearly independent column vectors spanning $\Pi_\alpha W_i$ for $i = 1, \ldots, g$ such that

$$B_{k\alpha} \subseteq B_{l\alpha} \quad \text{if} \quad W_k \subseteq W_l.$$

Then $B_i = \bigcup_{\alpha=1}^{\nu} B_{i\alpha}$ spans W_i. Order the n column vectors of $B = \bigcup_{i=1}^{g} B_i$ to get a nonsingular matrix $S = \bigoplus_{\alpha=1}^{\nu} S_\alpha$ and put $\widetilde{A} := S^{-1}AS$. Let C_l be the column subset corresponding to

$$\hat{B}_l = B_l - \bigcup_{k: W_k \subsetneq W_l} B_k.$$

Note that B_l spanning W_l is divided into disjoint subsets as

$$B_l = \bigcup_{k: W_k \subseteq W_l} \hat{B}_k.$$

Since W_l is an invariant subspace of A,

$$\widetilde{A}[R_k, C_l] = O \quad \text{unless} \quad W_k \subseteq W_l,$$

which implies that

$$W_k \subseteq W_l \quad \text{if} \quad C_k \preceq C_l.$$

Hence $S\psi(\mathcal{D}(\widetilde{A})) \supseteq \mathcal{D}'$. □

The following theorem is a direct consequence of Lemmas 2.5 and 2.6. Recall the definition of a finest-possible decomposition in §2.1.

THEOREM 2.7. *$\widetilde{A} = S^{-1}AS$ is a finest-possible decomposition of A if and only if $S\psi(\mathcal{D}(\widetilde{A}))$ is a maximal distributive sublattice of $\mathcal{L}(A, \Pi)$.*

Since a maximal distributive sublattice of a modular lattice always contains a maximal chain of the whole lattice, we have the following corollary.

COROLLARY 2.8. *A finest-possible decomposition of a partitioned square matrix is an irreducible decomposition.*

If $\mathcal{L}(A, \Pi)$ is distributive, $\mathcal{L}(A, \Pi)$ itself is the only maximal distributive sublattice of $\mathcal{L}(A, \Pi)$. Then also we have the following corollary.

COROLLARY 2.9. *If $\mathcal{L}(A, \Pi)$ is distributive, there exists a block-triangularized partitioned matrix $(\widetilde{A}, \Pi)$, Π-similar to A, such that $\mathcal{D}(\widetilde{A}) \cong \mathcal{L}(A, \Pi)$.*

The following lemma gives a sufficient condition for (A, Π) to make $\mathcal{L}(A, \Pi)$ distributive in the case of $F = \mathbb{C}$.

LEMMA 2.10. *Assume $F = \mathbb{C}$. If, for each $\alpha = 1, \ldots, \nu$, $A_{\alpha\alpha}$ has n_α distinct eigenvalues, then $\mathcal{L}(A, \Pi)$ is distributive.*

Proof. Let W be an element of $\mathcal{L}(A, \Pi)$. It follows from $AW \subseteq W$ and $\Pi_\alpha W \subseteq W$ for $\alpha = 1, \ldots, \nu$ that

$$\Pi_\alpha A \Pi_\alpha W \subseteq \Pi_\alpha W \quad \text{for} \quad \alpha = 1, \ldots, \nu,$$

which implies $\Pi_\alpha W$ is an invariant subspace of $A_{\alpha\alpha}$. Since $A_{\alpha\alpha}$ has distinct eigenvalues, an invariant space of $A_{\alpha\alpha}$ is the direct sum of a certain subset of the one-dimensional eigenspaces. Since $W = \bigoplus_{\alpha=1}^{\nu} \Pi_\alpha W$, W corresponds to a certain subset of the set of eigenvalues of $A_{\alpha\alpha}$ for $\alpha = 1, \ldots, \nu$. Thus $\mathcal{L}(A, \Pi)$ can be regarded as a sublattice of the Boolean lattice. Hence $\mathcal{L}(A, \Pi)$ is distributive. □

This lemma tells us that we can uniquely determine the finest block-triangularization of (A, Π), if $A_{\alpha\alpha}$ has distinct eigenvalues. We state a simple algorithm for such a matrix in §4.

Next, we consider the case where $\mathcal{L}(A, \Pi)$ is not distributive, and show that $\mathcal{L}(A, \Pi)$ can have infinitely many maximal distributive sublattices.

LEMMA 2.11. *Suppose that $\mathcal{L}(A, \Pi)$ is not distributive. Then there exists a block-triangular matrix $\widetilde{A}$, Π-similar to A, with a pair of blocks (C_k, C_l) such that $\widetilde{A}[R_k, C_l] = O$, $\widetilde{A}[R_l, C_k] = O$, $\widetilde{A}[R_k, C_k] = \widetilde{A}[R_l, C_l]$, and $\Pi_\alpha[R_k, C_k] = \Pi_\alpha[R_l, C_l]$ for $\alpha = 1, \ldots, \nu$.*

Proof. Since $\mathcal{L}(A, \Pi)$ is not distributive, it has a sublattice $\mathcal{L}' = \{M_1, M_2, M_3, M_4, M_5\}$ shown in Fig. 2, where $M_5 = M_2 + M_3 = M_3 + M_4 = M_4 + M_2$, and $M_1 = M_2 \cap M_3 = M_3 \cap M_4 = M_4 \cap M_2$. It follows from the isomorphism theorem on modules that

$$M_5/M_2 \cong M_5/M_3 \cong M_5/M_4 \cong M_2/M_1 \cong M_3/M_1 \cong M_4/M_1.$$

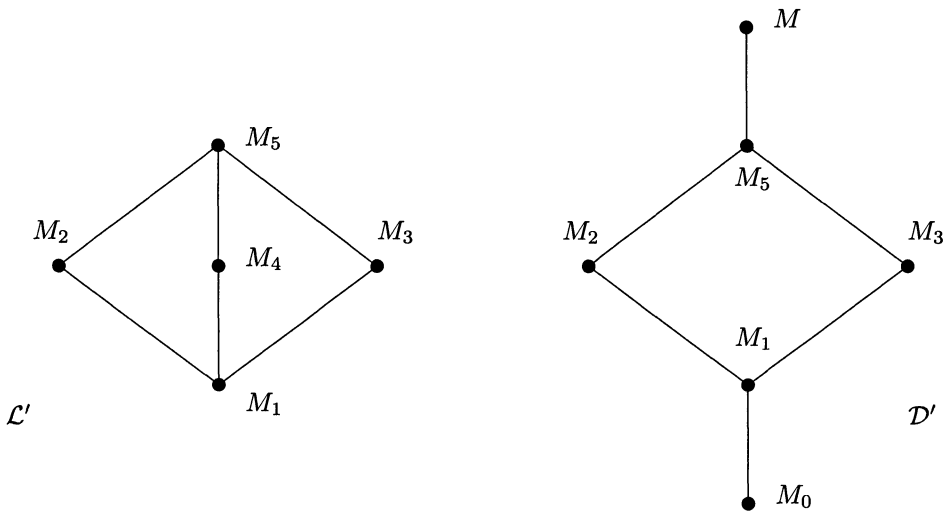


FIG. 2. *The lattices $\mathcal{L}'$ and $\mathcal{D}'$ in the proof of Lemma* 2.11.

Consider the distributive sublattice $\mathcal{D}' = \{M_0, M_1, M_2, M_3, M_5, M\}$ of $\mathcal{L}(A, \Pi)$, where M_1 and M_5 might coincide with $M_0 = 0$ and $M = M(A, \Pi)$, respectively. It follows from Lemma 2.6 and $M_2/M_1 \cong M_3/M_1$ that there exists a block-triangular matrix $\tilde{A}$, Π-similar to A, such that

$$\bar{A} = P^{\mathrm{T}}\tilde{A}P = \begin{array}{c} \\ R_k \\ R_l \end{array} \begin{array}{c} \begin{array}{cccc} C_k & C_l & & \end{array} \\ \begin{bmatrix} * & * & * & * \\ O & D & O & * \\ O & O & D & * \\ O & O & O & * \end{bmatrix} \end{array}$$

with a permutation matrix P. □

THEOREM 2.12. *Assume F is infinite. If $\mathcal{L}(A, \Pi)$ is not distributive, there exist infinitely many maximal distributive sublattices of $\mathcal{L}(A, \Pi)$.*

Proof. Consider the block-triangular matrix $\tilde{A}$ in the proof of Lemma 2.11. We may assume here that $\tilde{A}$ is a PS-irreducible decomposition. Consider the PS-transformation $\tilde{A}_\theta = Z_\theta^{-1}\tilde{A}Z_\theta$ by the nonsingular matrix

$$Z_\theta = P \begin{bmatrix} I & O & O & O \\ O & I & \theta I & O \\ O & O & I & O \\ O & O & O & I \end{bmatrix} P^{\mathrm{T}},$$

where $\theta \in F$ and I denotes the unit matrices of suitable dimensions. Then $\tilde{A}_\theta$ is also a PS-irreducible decomposition and there exists the composition series $\mathcal{C}_\theta$ from which the decomposition is obtained. It is easy to verify that $\mathcal{C}_\theta \neq \mathcal{C}_{\theta'}$ unless $\theta = \theta'$. This shows that there exist infinitely many composition series. Moreover, there exist infinitely many maximal distributive sublattices since a maximal distributive sublattice of $\mathcal{L}(A, \Pi)$ can contain only finite numbers of composition series. □

Now we are going to propose a canonical decomposition of partitioned square matrices. In case that $\mathcal{L}(A, \Pi)$ is distributive, we should define the canonical decom-

position of (A, Π) by the (unique) finest-possible decomposition $\widetilde{A}$, where $\mathcal{D}(\widetilde{A}) \cong \mathcal{L}(A, \Pi)$. However, when $\mathcal{L}(A, \Pi)$ is not distributive, obviously we can never find a Π-similar matrix $\widetilde{A}$ such that $\mathcal{D}(\widetilde{A}) \cong \mathcal{L}(A, \Pi)$. We have already defined a finest-possible decomposition $\widetilde{A}$ in the sense that $\widetilde{A}$ makes $\mathcal{D}(\widetilde{A})$ as close to $\mathcal{L}(A, \Pi)$ as possible. When $\mathcal{L}(A, \Pi)$ is not distributive, finest-possible decompositions are not unique. A canonical decomposition should represent the common structure among these finest-possible decompositions. Thus we are led to the definition of a *canonical decomposition of a partitioned square matrix* as a block-triangular form that realizes the largest common sublattice of the maximal distributive sublattices of $\mathcal{L}(A, \Pi)$. Note that a canonical decomposition is not necessarily an irreducible decomposition, let alone a finest-possible decomposition.

Example 2.3. Consider a partitioned square matrix (A, Π) that is in a block-triangular form with five PS-irreducible components L_1, L_2, L_3, D, and D, such that

$$\bar{A} = P^{\mathrm{T}}AP = \begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \begin{bmatrix} L_1 & O & O & H_1 & O \\ O & L_2 & O & O & H_2 \\ O & O & L_3 & H_3 & -H_3 \\ O & O & O & D & O \\ O & O & O & O & D \end{bmatrix} \end{array},$$

with a permutation matrix P, where $\Pi_\alpha[R_4, C_4] = \Pi_\alpha[R_5, C_5]$ for $\alpha = 1, \ldots, \nu$, $H_j \neq O$ for $j = 1, 2, 3$ and there exist no PS-transformations among L_1, L_2, L_3, and D.

The whole lattice $\mathcal{L}(A, \Pi)$ (in the case that F is an infinite field) and the sublattice $\mathcal{D}(A)$ are illustrated in Fig. 3. $\mathcal{D}(A)$ is a maximal distributive sublattice of $\mathcal{L}(A, \Pi)$. Thus A itself is a finest-possible decomposition of A. Consider the PS-transformation by the nonsingular matrix Z' defined by

$$Z'[R_k, C_l] = \begin{cases} I & \text{if } k = l \text{ or } (k, l) = (4, 5), \\ O & \text{otherwise,} \end{cases}$$

and put $A' = Z'^{-1}AZ'$. Then we have

$$\bar{A}' = P^{\mathrm{T}}A'P = \begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \begin{bmatrix} L_1 & O & O & H_1 & H_1 \\ O & L_2 & O & O & H_2 \\ O & O & L_3 & H_3 & O \\ O & O & O & D & O \\ O & O & O & O & D \end{bmatrix} \end{array}.$$

The lattice $\mathcal{D}(A')$ illustrated in Fig. 3 is also a maximal distributive sublattice of $\mathcal{L}(A, \Pi)$. We also consider the PS-transformation by the nonsingular matrix Z'' defined by

$$Z''[R_k, C_l] = \begin{cases} I & \text{if } k = l \text{ or } (k, l) = (5, 4), \\ O & \text{otherwise,} \end{cases}$$

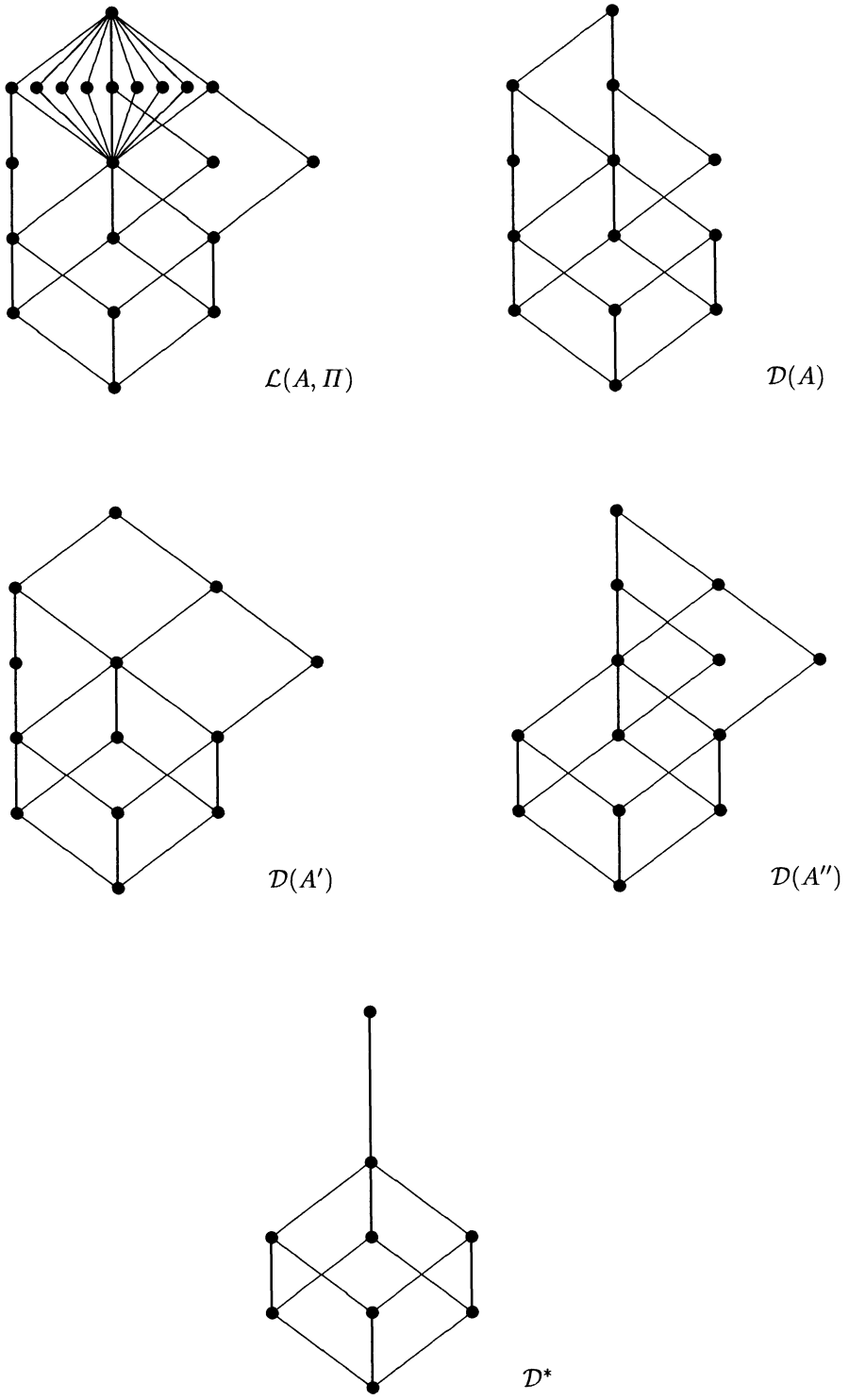


FIG. 3. *The lattice $\mathcal{L}(A, \Pi)$, three maximal distributive sublattices $\mathcal{D}(A)$, $\mathcal{D}(A')$, $\mathcal{D}(A'')$, and the distributive sublattice $\mathcal{D}^*$ for the canonical decomposition in Example* 2.3.

and put $A'' = Z''^{-1}AZ''$. Then we have

$$\bar{A}'' = P^{\mathrm{T}}A''P = \begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \begin{bmatrix} L_1 & O & O & H_1 & O \\ O & L_2 & O & H_2 & H_2 \\ O & O & L_3 & O & -H_3 \\ O & O & O & D & O \\ O & O & O & O & D \end{bmatrix} \end{array}.$$

The lattice $\mathcal{D}(A'')$ illustrated in Fig. 3 is also a maximal distributive sublattice of $\mathcal{L}(A, \Pi)$. The common sublattice $\mathcal{D}^*$, corresponding to the canonical decomposition, of the maximal distributive sublattices of $\mathcal{L}(A, \Pi)$ is also illustrated in Fig. 3.

3. Decomposition under partition-respecting equivalency.

3.1. Problem formulation. In this section we discuss the block-triangularization of a partitioned, not necessarily square, matrix under restricted equivalence transformations which preserve the partition structure. More precisely, let A be an $m \times n$ matrix over a field F with partitions:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1\nu} \\ A_{21} & A_{22} & \cdots & A_{2\nu} \\ \vdots & \vdots & \ddots & \vdots \\ A_{\mu 1} & A_{\mu 2} & \cdots & A_{\mu\nu} \end{bmatrix},$$

where $A_{\alpha\beta}$ is an $m_\alpha \times n_\beta$ matrix called (α, β)-submatrix of A. To describe the partition structure in terms of matrix operation, we introduce a family of $m \times m$ projection matrices Π_α for $\alpha = 1, \ldots, \mu$ and another family of $n \times n$ projection matrices Γ_β for $\beta = 1, \ldots, \nu$. The (α, α)-submatrix of Π_α is the unit matrix I_{m_α} of dimension m_α and the other submatrices are zeros. Similarly, the (β, β)-submatrix of Γ_β is the unit matrix I_{n_β} of dimension n_β and all the other submatrices are zeros. Sometimes we denote the partitioned matrix by the triple (A, Π, Γ), where $\Pi = \{\Pi_\alpha\}_{\alpha=1}^{\mu}$ and $\Gamma = \{\Gamma_\beta\}_{\beta=1}^{\nu}$. A transformation $\widetilde{A} := S_{\mathrm{r}}^{-1}AS_{\mathrm{c}}$ of A is said to be a *partition-respecting equivalence transformation* (PE-transformation) if the off-diagonal submatrices of S_{r} and S_{c} are zeros, i.e.,

$$\Pi_\alpha S_{\mathrm{r}} = S_{\mathrm{r}}\Pi_\alpha \quad \text{for} \quad \alpha = 1, \ldots, \mu$$

and

$$\Gamma_\beta S_{\mathrm{c}} = S_{\mathrm{c}}\Gamma_\beta \quad \text{for} \quad \beta = 1, \ldots, \nu.$$

We say that A is (Π, Γ)-equivalent to A' if there exists a PE-transformation that transforms A to A'. Then our problem is to bring A into a block-triangular form by means of a PE-transformation.

Alternatively, our problem can be formulated as follows. Let U and V be F-linear spaces F^m and F^n, respectively, which are direct sums of lower dimensional spaces

$$U = \bigoplus_{\alpha=1}^{\mu} U_\alpha, \qquad \dim_F U_\alpha = m_\alpha,$$

$$V = \bigoplus_{\beta=1}^{\nu} V_\beta, \qquad \dim_F V_\beta = n_\beta.$$

A linear transformation

$$f : V \to U$$

is constructed by a family of linear transformations

$$f_{\alpha\beta} : V_\beta \to U_\alpha, \quad \alpha = 1, \ldots, \mu; \quad \beta = 1, \ldots, \nu.$$

When we take arbitrary bases for each space U_α and V_β, f is represented by a partitioned matrix A, where $A_{\alpha\beta}$ corresponds to $f_{\alpha\beta}$. In this context PE-transformations are naturally derived as basis transformations of U_α and V_β.

Now we define precisely the notion of a block-triangular form, following Murota [22]. Let $\widetilde{A}$ be a partitioned matrix. We say that $\widetilde{A}$ is in a *block-triangular form* or is *block-triangularized* if the row set $R = \{1, 2, \ldots, m\}$ and the column set $C = \{1, 2, \ldots, n\}$ are split into a certain number of disjoint blocks: $(R_0; R_1, \ldots, R_b; R_\infty)$ and $(C_0; C_1, \ldots, C_b; C_\infty)$ in such a way that

$$\begin{array}{lll} |R_0| < |C_0| & \text{or} & |R_0| = |C_0| = 0, \\ |R_k| = |C_k| > 0 & \text{for} & k = 1, \ldots, b, \\ |R_\infty| > |C_\infty| & \text{or} & |R_\infty| = |C_\infty| = 0, \end{array}$$

and

$$\widetilde{A}[R_k, C_l] = O \quad \text{if} \quad 0 \le l < k \le \infty.$$

$\widetilde{A}$ is said to be *properly* block-triangularized if, in addition,

$$\operatorname{rank}\widetilde{A}[R_k, C_k] = \min(|R_k|, |C_k|) \quad \text{for} \quad k = 0, 1, \ldots, b, \infty$$

is satisfied. $\widetilde{A}[R_0, C_0]$ and $\widetilde{A}[R_\infty, C_\infty]$ are called *horizontal tail* and *vertical tail* of $\widetilde{A}$, respectively. It is clear that if $\widetilde{A}$ is block-triangularized in the above sense, we can put it into an explicit upper block-triangular form $\bar{A} = P_{\mathrm{r}}\widetilde{A}P_{\mathrm{c}}$ in the usual sense by using certain permutation matrices P_{r} and P_{c}.

A partitioned matrix (A, Π, Γ), or simply A, is said to be PE-*irreducible* if $\operatorname{rank}A = \min(m, n)$ and it can never be transformed into a proper block-triangular form with two or more nonempty blocks by any PE-transformations. If $\widetilde{A}$ is a proper block-triangular matrix obtained from A by a PE-transformation and, in addition, all the diagonal blocks $\widetilde{A}[R_k, C_k]$ for $k = 0, 1, \ldots, b, \infty$ are PE-irreducible, we say that $\widetilde{A}$ is a (PE)-*irreducible decomposition* of A and $\widetilde{A}[R_k, C_k]$ are the (PE)-*irreducible components* of A.

Our problem of proper block-triangularization above contains two well-known extreme cases. The first case is with the trivial partition structure: $\mu = 1$, $\nu = 1$. In this case our problem has been completely solved by the rank normal form [11]. The other extreme case is with the finest partition structure: $\mu = m$, $\nu = n$. In this case our problem is solved by the DM-decomposition of bipartite graphs [10] with efficient algorithms. The CCF of layered mixed matrices, which is a proper extension of DM-decomposition, can be regarded as a special case; we explain this relationship in §3.3.

On the analogy of DM-decomposition, a partial order is induced among the blocks $\{C_k \mid k = 1, \ldots, b\}$ in a natural manner by the zero/nonzero structure of a block-triangular matrix $\widetilde{A}$. The partial order $\preceq$ is the reflexive and transitive closure of the

relation defined by: C_k is "smaller" than or equal to C_l if $\widetilde{A}[R_k, C_l] \neq O$. We denote this poset $(\{C_1, \ldots, C_b\}, \preceq)$ by $\mathcal{P}(\widetilde{A})$.

As is well known [2], [5], the ideals of the poset $\mathcal{P}(\widetilde{A})$ constitute a distributive lattice, which we denote by $\mathcal{D}(\widetilde{A})$. Note that a subset J of C can be naturally identified with a subspace of V. As in §2 we denote this correspondence by $\psi(J)$. Then

$$\psi(\mathcal{D}(\widetilde{A})) = \{\psi(J) \mid J \in \mathcal{D}(\widetilde{A})\}$$

is a distributive sublattice of the modular lattice formed by the subspaces of V.

We say that $\widetilde{A} = S_{\mathrm{r}}^{-1} A S_{\mathrm{c}}$ is *finer*, as a decomposition of A, than $\widetilde{A}' = S'^{-1}_{\mathrm{r}} A S'_{\mathrm{c}}$ if $S'_{\mathrm{c}}\psi(\mathcal{D}(\widetilde{A}')) = \{S_{\mathrm{c}}\widetilde{W}' \mid \widetilde{W}' \in \psi(\mathcal{D}(\widetilde{A}'))\}$ is a proper sublattice of $S_{\mathrm{c}}\psi(\mathcal{D}(\widetilde{A})) = \{S_{\mathrm{c}}\widetilde{W} \mid \widetilde{W} \in \psi(\mathcal{D}(\widetilde{A}))\}$. Furthermore, we say that $\widetilde{A}$, (Π, Γ)-equivalent to A, is a *finest-possible decomposition* of (A, Π, Γ) if there does not exist a proper block-triangular matrix $\widetilde{A}'$ that is also (Π, Γ)-equivalent to A and finer than $\widetilde{A}$.

Example 3.1. Consider the following 4×5 matrix A over $F = \mathbb{Q}$:

$$A = \begin{array}{c} \\ \star \\ \star \\ \diamond \\ \diamond \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \circ & \circ & \bullet & \bullet \end{array} \\ \left[\begin{array}{ccc|cc} 1 & 1 & 1 & 1 & 0 \\ 0 & 2 & 1 & 1 & 1 \\ \hline 2 & -2 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 & 4 \end{array}\right] \end{array},$$

where $\mu = 2$, $\nu = 2$, $m_1 = 2$, $m_2 = 2$, $n_1 = 3$, $n_2 = 2$. Let S_{r} and S_{c} be the nonsingular matrices

$$S_{\mathrm{r}} = \left[\begin{array}{cc|cc} 1 & 1 & \multicolumn{2}{c}{\multirow{2}{*}{O}} \\ 1 & 0 & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{O}} & 0 & 1 \\ & & 1 & 0 \end{array}\right], \qquad S_{\mathrm{c}} = \left[\begin{array}{ccc|cc} 0 & 1 & 1 & \multicolumn{2}{c}{\multirow{3}{*}{O}} \\ 0 & 1 & 0 & & \\ 1 & 0 & 0 & & \\ \hline \multicolumn{3}{c|}{\multirow{3}{*}{O}} & 1 & 0 \\ & & & 0 & 1 \end{array}\right].$$

Then we have

$$\widetilde{A} = S_{\mathrm{r}}^{-1} A S_{\mathrm{c}} = \begin{array}{c} \\ \star \\ \star \\ \diamond \\ \diamond \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \circ & \circ & \bullet & \bullet \end{array} \\ \left[\begin{array}{ccc|cc} 1 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 \\ \hline 0 & 3 & 0 & 0 & 4 \\ 0 & 0 & 2 & 0 & 2 \end{array}\right] \end{array}.$$

By using the permutation matrices

$$P_{\mathrm{r}} = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right], \qquad P_{\mathrm{c}} = \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

we can transform $\widetilde{A}$ into an explicit upper block-triangular form

$$\bar{A} = P_{\mathrm{r}} \widetilde{A} P_{\mathrm{c}} = \begin{array}{c} \\ \star \\ \diamond \\ \star \\ \diamond \end{array} \begin{array}{c} \begin{array}{ccccc} \circ & \bullet & \circ & \circ & \bullet \end{array} \\ \left[\begin{array}{c|c|cc|c} 1 & 1 & 2 & 0 & 1 \\ \hline & & 3 & 0 & 4 \\ \multicolumn{2}{c|}{\multirow{2}{*}{O}} & & 1 & -1 \\ & & & 2 & 2 \end{array}\right] \end{array}.$$

Then $\widetilde{A}$ is in a proper block-triangular form with two square blocks, a nonempty horizontal tail ($|R_0| = 1$, $|C_0| = 2$) and an empty vertical tail.

The main results of this section are as follows. We exploit two approaches, one based on module theory and the other on submodularity. It should be remarked that a proper block-triangularization does not always exist. It is shown that the existence of a proper block-triangularization is equivalent to the existence of an invariant module (see Theorem 3.5), and that it is also equivalent to the equality of the rank of A to the minimum value of a submodular function (see Theorem 3.15). Sufficient conditions for the existence are given by the nonsingularity (see Lemma 3.2), or the genericity (see Theorem 3.16). The PE-irreducibility is characterized either by the simplicity of the associated module (see Theorem 3.6) or by the minimizers of the associated submodular function (see Theorem 3.15). The uniqueness of irreducible decomposition is due to the Jordan–Hölder theorem for modules (see Theorem 3.7). It will be shown that a finest-possible decomposition is obtained by a maximal distributive sublattice of the modular lattice formed by the submodules of the associated module; the modular lattice can also be characterized as the lattice of the minimizers of the associated submodular function (see Theorem 3.10).

3.2. Decomposition into irreducible components and their ordering. This section is devoted to the construction of an irreducible decomposition and a finest-possible decomposition by means of module and lattice theories.

To begin with, we observe some properties of a proper block-triangular matrix and investigate a necessary condition for the proper block-triangularizability of a partitioned matrix (A, Π, Γ). Suppose $\widetilde{A} = S_{\mathrm{r}}^{-1} A S_{\mathrm{c}}$ is in a proper block-triangular form with blocks $(C_0; C_1, \ldots, C_b; C_\infty)$ and (Π, Γ)-equivalent to A. Put $\widetilde{W} = \psi(C_0)$, which is the subspace of V that corresponds to C_0. Since $\widetilde{A}$ is also a partitioned matrix, we have $\Gamma_\beta \widetilde{W} \subseteq \widetilde{W}$ for $\beta = 1, \ldots, \nu$. It follows from the definition of a proper block-triangular form that $\Pi_\alpha \widetilde{A}\widetilde{W} \subseteq \widetilde{A}\widetilde{W}$ for $\alpha = 1, \ldots, \mu$ and $\mathrm{Ker}\widetilde{A} \subseteq \widetilde{W}$. Put $W = S_{\mathrm{c}}\widetilde{W}$, and then we have

$$\Gamma_\beta W \subseteq W \quad \text{for} \quad \beta = 1, \ldots, \nu,$$

$$\Pi_\alpha AW \subseteq AW \quad \text{for} \quad \alpha = 1, \ldots, \mu,$$

and

$$\mathrm{Ker}A \subseteq W.$$

Let $\mathcal{L}(A, \Pi, \Gamma)$ be the family of such subspaces W of V. Then we have the following lemma.

LEMMA 3.1. *If a partitioned matrix (A, Π, Γ) can be transformed into a proper block-triangular form by means of a* PE*-transformation, then $\mathcal{L}(A, \Pi, \Gamma)$ is nonempty.*

If $\mathrm{rank}A = \min(m, n)$ holds, $W = 0$ or $W = V$ satisfies the three conditions above. Hence we have the following lemma.

LEMMA 3.2. *If $\mathrm{rank}A = \min(m, n)$, then $\mathcal{L}(A, \Pi, \Gamma)$ is nonempty.*

The converse of Lemma 3.1 is also true, which is shown later (see Theorem 3.5).

Now we assume in this subsection that $\mathcal{L}(A, \Pi, \Gamma)$ is nonempty. Then $\mathcal{L}(A, \Pi, \Gamma)$ is a modular lattice. We denote the maximum element of $\mathcal{L}(A, \Pi, \Gamma)$ by $V_\top$, and the minimum by $V_\perp$.

Let Σ^* be a free monoid generated by a finite set

$$\Sigma = \{z_1, \ldots, z_\nu, z_{\nu+1}, \ldots, z_{\nu+\mu}\}.$$

The action of Σ^* on the quotient linear space $V_\top/V_\bot$ is defined as follows:

$$\left.\begin{array}{ll} z_\beta(x+V_\bot) := \gamma_\beta(x) + V_\bot & (\beta = 1, \ldots, \nu), \\ z_{\nu+\alpha}(x+V_\bot) := f^{-1}(\pi_\alpha(f(x))) + V_\bot & (\alpha = 1, \ldots, \mu) \end{array}\right\} x \in V_\top,$$

where $\gamma_\beta : V \to V$ and $\pi_\alpha : U \to U$ are the projections represented by Γ_β and Π_α, respectively.

LEMMA 3.3. *The action of Σ^* on $V_\top/V_\bot$ is well defined.*

Proof. Consider a vector $y \in V_\top$ such that $y - x \in V_\bot$. Since $\Gamma_\beta V_\bot \subseteq V_\bot$, we have $\gamma_\beta(y) - \gamma_\beta(x) \in V_\bot$. Similarly, it follows from $\Pi_\alpha A V_\bot \subseteq A V_\bot$ that $\pi_\alpha(f(y)) - \pi_\alpha(f(x)) \in f(V_\bot)$. Consider two vectors $x', y' \in V_\top$ such that $\pi_\alpha(f(x)) = f(x')$, $\pi_\alpha(f(y)) = f(y')$. The existence of such vectors follows from $\Pi_\alpha A V_\top \subseteq A V_\top$. Since $\mathrm{Ker} A \subseteq V_\bot$, we have $f^{-1}(\pi_\alpha(f(x))) \subseteq x' + V_\bot$, $f^{-1}(\pi_\alpha(f(y))) \subseteq y' + V_\bot$ and $y' - x' \in V_\bot$. Therefore $f^{-1}(\pi_\alpha(f(x))) + V_\bot = f^{-1}(\pi_\alpha(f(y))) + V_\bot$. Hence the action of Σ^* on $V_\top/V_\bot$ is well defined. □

Thus $V_\top/V_\bot$ becomes an $F\langle\Sigma^*\rangle$-module, as well as an F-linear space. We denote this module by $M(A, \Pi, \Gamma)$. Note that $W/V_\bot$ is a submodule of M if and only if $W \in \mathcal{L}(A, \Pi, \Gamma)$. That is to say $\mathcal{L}(A, \Pi, \Gamma)$ coincides with the modular lattice that consists of the submodules of $M(A, \Pi, \Gamma)$. For another partitioned matrix A' with common (Π, Γ), we obtain another modular lattice $\mathcal{L}(A', \Pi, \Gamma)$ and another $F\langle\Sigma^*\rangle$-module $M(A', \Pi, \Gamma)$. Note that $M(A', \Pi, \Gamma)$ is isomorphic under PE-transformations, i.e., $M(A, \Pi, \Gamma) \cong M(A', \Pi, \Gamma)$ if A is (Π, Γ)-equivalent to A'. The converse is true for square and nonsingular matrices.

LEMMA 3.4. *Suppose A and A' are square and nonsingular. Then $M(A, \Pi, \Gamma) \cong M(A', \Pi, \Gamma)$ if and only if A is (Π, Γ)-equivalent to A'.*

Proof. Let $\varphi : M(A, \Pi, \Gamma) \to M(A', \Pi, \Gamma)$ be the module isomorphism. Since $F \subseteq F\langle\Sigma^*\rangle$, $M(A, \Pi, \Gamma) = M(A', \Pi, \Gamma) = V$ as the F-linear spaces, and φ a bijective linear transformation that is to be represented by a nonsingular matrix S_c. Put

$$S_\mathrm{r} = A' S_\mathrm{c} A^{-1},$$

then $A' S_\mathrm{c} = S_\mathrm{r} A$ is satisfied. For any $x \in M(A, \Pi, \Gamma) = V$, we have

$$\Gamma_\beta S_\mathrm{c} x = z_\beta \varphi(x) = \varphi(z_\beta x) = S_\mathrm{c} \Gamma_\beta x \quad \text{for} \quad \beta = 1, \ldots, \nu,$$

which implies $\Gamma_\beta S_\mathrm{c} = S_\mathrm{c} \Gamma_\beta$ for $\beta = 1, \ldots, \nu$. For any $x \in M(A, \Pi, \Gamma) = V$, we also have

$$A'^{-1} \Pi_\alpha A' S_\mathrm{c} x = z_{\nu+\alpha} \varphi(x) = \varphi(z_{\nu+\alpha} x) = S_\mathrm{c} A^{-1} \Pi_\alpha A x \quad \text{for} \quad \alpha = 1, \ldots, \mu.$$

It follows from $A' S_\mathrm{c} = S_\mathrm{r} A$ that

$$A'^{-1} \Pi_\alpha S_\mathrm{r} A x = A'^{-1} S_\mathrm{r} \Pi_\alpha A x.$$

Because A and A' are nonsingular, we have obtained $\Pi_\alpha S_\mathrm{r} = S_\mathrm{r} \Pi_\alpha$ for $\alpha = 1, \ldots, \mu$. □

Now we explain how to get an irreducible decomposition. Let $\mathcal{C}$ be a maximal chain of $\mathcal{L}(A, \Pi, \Gamma)$:

$$\mathcal{C} : V_\bot = W_0 \subsetneq W_1 \subsetneq \cdots \subsetneq W_h = V_\top.$$

Since $\Gamma_\beta W_i \subseteq W_i$ for $\beta = 1, \ldots, \nu$ and $\sum_{\beta=1}^{\nu} \Gamma_\beta = I_n$, we have

$$W_i = \bigoplus_{\beta=1}^{\nu} \Gamma_\beta W_i.$$

For $\beta = 1, \ldots, \nu$, we can obtain increasing chains

$$\Gamma_\beta \mathcal{C} : \Gamma_\beta V_\perp = \Gamma_\beta W_0 \subseteq \Gamma_\beta W_1 \subseteq \cdots \subseteq \Gamma_\beta W_h = \Gamma_\beta V_\top.$$

Let $B^{\mathrm{c}}_{i\beta}$ be a set of linearly independent column vectors spanning $\Gamma_\beta W_i$ for $i = 0, 1, \ldots, h$ and $B^{\mathrm{c}}_{\infty\beta}$ spanning V_β such that

$$B^{\mathrm{c}}_{0\beta} \subseteq B^{\mathrm{c}}_{1\beta} \subseteq \cdots \subseteq B^{\mathrm{c}}_{h\beta} \subseteq B^{\mathrm{c}}_{\infty\beta}.$$

Then $B^{\mathrm{c}}_i = \bigcup_{\beta=1}^{\nu} B^{\mathrm{c}}_{i\beta}$ spans W_i for $i = 0, 1, \ldots, h$, and $B^{\mathrm{c}} = \bigcup_{\beta=1}^{\nu} B^{\mathrm{c}}_{\infty\beta}$ becomes a basis of V. Order the n column vectors of B^{c} as $[B^{\mathrm{c}}_{\infty 1}, B^{\mathrm{c}}_{\infty 2}, \ldots, B^{\mathrm{c}}_{\infty\nu}]$ to get a nonsingular matrix $S_{\mathrm{c}} = \bigoplus_{\beta=1}^{\nu} S_{\mathrm{c}\beta}$.

Similarly since $\Pi_\alpha A W_i \subseteq A W_i$ for $\alpha = 1, \ldots, \mu$ and $\sum_{\alpha=1}^{\mu} \Pi_\alpha = I_m$, we have

$$A W_i = \bigoplus_{\alpha=1}^{\mu} \Pi_\alpha A W_i.$$

We obtain increasing chains

$$\Pi_\alpha A \mathcal{C} : \Pi_\alpha A V_\perp = \Pi_\alpha A W_0 \subseteq \Pi_\alpha A W_1 \subseteq \cdots \subseteq \Pi_\alpha A W_h = \Pi_\alpha A V_\top$$

for $\alpha = 1, \ldots, \mu$. Let $B^{\mathrm{r}}_{i\alpha}$ be a set of linearly independent column vectors spanning $\Pi_\alpha A W_i$ for $i = 0, 1, \ldots, h$ and $B^{\mathrm{r}}_{\infty\alpha}$ spanning U_α such that

$$B^{\mathrm{r}}_{0\alpha} \subseteq B^{\mathrm{r}}_{1\alpha} \subseteq \cdots \subseteq B^{\mathrm{r}}_{h\alpha} \subseteq B^{\mathrm{r}}_{\infty\alpha}.$$

Then $B^{\mathrm{r}}_i = \bigcup_{\alpha=1}^{\mu} B^{\mathrm{r}}_{i\alpha}$ spans $A W_i$ for $i = 0, 1, \ldots, h$, and $B^{\mathrm{r}} = \bigcup_{\alpha=1}^{\mu} B^{\mathrm{r}}_{\infty\alpha}$ becomes a basis of U. Order the m column vectors of B^{r} as $[B^{\mathrm{r}}_{\infty 1}, B^{\mathrm{r}}_{\infty 2}, \ldots, B^{\mathrm{r}}_{\infty\mu}]$ to get a nonsingular matrix $S_{\mathrm{r}} = \bigoplus_{\alpha=1}^{\mu} S_{\mathrm{r}\alpha}$.

Put $\widetilde{A} := S_{\mathrm{r}}^{-1} A S_{\mathrm{c}}$. Let C_i be the column subset corresponding to $\hat{B}^{\mathrm{c}}_i$, and R_i the row subset corresponding to $\hat{B}^{\mathrm{r}}_i$, where

$$\begin{array}{ll} \hat{B}^{\mathrm{c}}_0 = B^{\mathrm{c}}_0, & \hat{B}^{\mathrm{r}}_0 = B^{\mathrm{r}}_0, \\ \hat{B}^{\mathrm{c}}_i = B^{\mathrm{c}}_i - B^{\mathrm{c}}_{i-1}, & \hat{B}^{\mathrm{r}}_i = B^{\mathrm{r}}_i - B^{\mathrm{r}}_{i-1} \quad \text{for} \quad i = 1, \ldots, h, \\ \hat{B}^{\mathrm{c}}_\infty = B^{\mathrm{c}}_\infty - B^{\mathrm{c}}_h, & \hat{B}^{\mathrm{r}}_\infty = B^{\mathrm{r}}_\infty - B^{\mathrm{r}}_h. \end{array}$$

Then we have

$$\widetilde{A}[R_k, C_l] = O \quad \text{if} \quad 0 \leq l < k \leq \infty.$$

Since $\Pi_\alpha A W_k \subseteq A W_k$ for $k = 0, 1, \ldots, h$, we have

$$\mathrm{rank} \widetilde{A}[R_k, C_k] = |R_k| \quad \text{for } k = 0, 1, \ldots, h.$$

On the other hand, since $\mathrm{Ker} A \subseteq V_\perp$, we also have

$$\mathrm{rank} \widetilde{A}[R_k, C_k] = |C_k| \quad \text{for } k = 1, \ldots, h, \infty.$$

Hence,

$$|R_k| = |C_k| \quad \text{for } k = 1, \dots, h$$

and

$$\operatorname{rank}\widetilde{A}[R_k, C_k] = \min(|R_k|, |C_k|) \quad \text{for} \quad k = 0, 1, \dots, h, \infty.$$

That is to say, $\widetilde{A}$ is in a proper block-triangular form, where the number of square blocks b is given by the length h of $\mathcal{C}$. Thus we have the following theorem.

THEOREM 3.5. *There exists a proper block-triangular matrix that is* (Π, Γ)-*equivalent to* A, *if and only if* $\mathcal{L}(A, \Pi, \Gamma)$ *is nonempty.*

Before showing that $\widetilde{A}$ obtained above is an irreducible decomposition, we give a necessary and sufficient condition for PE-irreducibility. Recall Lemma 3.2.

THEOREM 3.6. *Suppose* $\operatorname{rank} A = \min(m, n)$. *Then* A *is* PE-*irreducible if and only if there exists no element in* $\mathcal{L}(A, \Pi, \Gamma)$ *except* 0 *or* V, *namely, if and only if* $\mathcal{L}(A, \Pi, \Gamma) \subseteq \{0, V\}$.

Proof. Suppose $\widetilde{A}$ is in a proper block-triangular form with two or more nonempty blocks and (Π, Γ)-equivalent to A. If C_0 is nonempty, put $W = S_c\psi(C_0)$; otherwise $W = S_c\psi(C_1)$. Then W is a nontrivial subspace of V that belongs to $\mathcal{L}(A, \Pi, \Gamma)$. This is a contradiction. The converse is immediate from the construction above. □

Now we are ready to show that the proper block-triangular matrix $\widetilde{A}$ obtained from a maximal chain $\mathcal{C}$ of $\mathcal{L}(A, \Pi, \Gamma)$ is an irreducible decomposition of A. A composition series of $M(A, \Pi, \Gamma)$ is given by

$$\mathcal{C}/V_\perp : 0 = M_0 \subsetneq M_1 \subsetneq \cdots \subsetneq M_h = M(A, \Pi, \Gamma),$$

where $M_i = W_i/V_\perp$ for $i = 0, 1, \dots, h$. Put

$$D_k := \widetilde{A}[R_k, C_k], \quad \Phi_k := \{\Pi_\alpha[R_k, C_k]\}_{\alpha=1}^{\mu}, \quad \Psi_k := \{\Gamma_\beta[R_k, C_k]\}_{\beta=1}^{\nu},$$

for $k = 0, 1, \dots, \infty$, then (D_k, Φ_k, Ψ_k) is also a partitioned matrix. For each $k = 1, \dots, h$, we can construct another $F\langle\Sigma^*\rangle$-module $N_k = M(D_k, \Phi_k, \Psi_k)$. Then $N_k \cong M_k/M_{k-1}$. This implies that D_k is PE-irreducible for $k = 1, \dots, h$. Moreover, since $V_\perp$ and $V_\top$ are the minimum and maximum elements of $\mathcal{L}(A, \Pi, \Gamma)$, respectively, it follows from Theorem 3.6 that D_0 and D_∞ are PE-irreducible. Thus $\widetilde{A}$ is an irreducible decomposition of A. Conversely, if $(\widetilde{A}, \Pi, \Gamma)$ is in a proper block-triangular form with irreducible diagonal blocks, the quotient spaces $\psi(\bigcup_{k=0}^{i} C_k)/\psi(C_0)$ for $i = 0, \dots, b$ determine submodules that constitute a composition series of $M(\widetilde{A}, \Pi, \Gamma)$.

We have already seen how to construct an irreducible decomposition, but we have not yet discussed the uniqueness. Suppose both $\widetilde{A} = S_r^{-1}AS_c$ and $\widetilde{A}' = S'^{-1}_r AS'_c$ are irreducible decompositions of (A, Π, Γ) with blocks $(R_0; R_1, \dots, R_b; R_\infty)$, $(C_0; C_1, \dots, C_b; C_\infty)$ and $(R'_0; R'_1, \dots, R'_{b'}; R'_\infty)$, $(C'_0; C'_1, \dots, C'_{b'}; C'_\infty)$, respectively. Put

$$D_k := \widetilde{A}[R_k, C_k], \quad \Phi_k := \{\Pi_\alpha[R_k, C_k]\}_{\alpha=1}^{\mu}, \quad \Psi_k := \{\Gamma_\beta[R_k, C_k]\}_{\beta=1}^{\nu},$$

for $k = 0, 1, \dots, b, \infty$, and

$$D'_l := \widetilde{A}'[R'_l, C'_l], \quad \Phi'_l := \{\Pi_\alpha[R'_l, C'_l]\}_{\alpha=1}^{\mu}, \quad \Psi'_l := \{\Gamma_\beta[R'_l, C'_l]\}_{\beta=1}^{\nu},$$

for $l = 0, 1, \dots, b', \infty$. Since $V_\perp$ is the unique minimum element of $\mathcal{L}(A, \Pi, \Gamma)$, it is obvious that $\Phi_0 = \Phi'_0$, $\Psi_0 = \Psi'_0$ and D_0 is (Φ_0, Ψ_0)-equivalent to D'_0. In this sense

the horizontal tail is unique. The uniqueness of the vertical tail is shown as follows. Put $\widetilde{S}_{\mathrm{r}} = S_{\mathrm{r}}^{-1}S'_{\mathrm{r}}$ and $\widetilde{S}_{\mathrm{c}} = S_{\mathrm{c}}^{-1}S'_{\mathrm{c}}$, then $\widetilde{S}_{\mathrm{r}}\widetilde{A}' = \widetilde{A}\widetilde{S}_{\mathrm{c}}$. It follows from $\psi(\bigcup_{k=0}^{b} C_k) = \widetilde{S}_{\mathrm{c}}\psi(\bigcup_{l=0}^{b'} C'_l) = S_{\mathrm{c}}V_{\top}$ that $\widetilde{S}_{\mathrm{r}}[R_\infty, \bigcup_{l=0}^{b'} R'_l] = O$ and $\widetilde{S}_{\mathrm{c}}[C_\infty, \bigcup_{k=0}^{b} C'_k] = O$ hold. An easy calculation leads to

$$\widetilde{S}_{\mathrm{r}}[R_\infty, R'_\infty]D_\infty = D'_\infty\widetilde{S}_{\mathrm{c}}[C_\infty, C'_\infty].$$

That is to say $\Phi_\infty = \Phi'_\infty$, $\Psi_\infty = \Psi'_\infty$, and D_∞ is $(\Phi_\infty, \Psi_\infty)$-equivalent to D'_∞. For nonsingular irreducible components, it follows from the Jordan–Hölder theorem for modules (Theorem 2.1) and Lemma 3.4 that $b = b'$ and there exists a permutation σ of $\{1, 2, \ldots, b\}$, such that $\Phi_k = \Phi'_{\sigma(k)}$, $\Psi_k = \Psi'_{\sigma(k)}$, and D_k is (Φ_k, Ψ_k)-equivalent to $D'_{\sigma(k)}$. Hence we have the following theorem on the uniqueness of irreducible components of a partitioned matrix.

THEOREM 3.7. *The set of* PE*-irreducible components of a partitioned matrix is unique to within* PE*-transformations of each component.*

As to the distributive lattice $\mathcal{D}(\widetilde{A})$ and the modular lattice $\mathcal{L}(A, \Pi, \Gamma)$, we have the following results that are quite similar to those in §2. We state them without proofs.

LEMMA 3.8. *If $\widetilde{A} = S_{\mathrm{r}}^{-1}AS_{\mathrm{c}}$ is (Π, Γ)-equivalent to A, then $S_{\mathrm{c}}\psi(\mathcal{D}(\widetilde{A}))$ is a sublattice of $\mathcal{L}(A, \Pi, \Gamma)$.*

LEMMA 3.9. *For any distributive sublattice $\mathcal{D}'$ of $\mathcal{L}(A, \Pi, \Gamma)$, there exists a* PE*-transformation $\widetilde{A} = S_{\mathrm{r}}^{-1}AS_{\mathrm{c}}$ such that $S_{\mathrm{c}}\psi(\mathcal{D}(\widetilde{A})) \supseteq \mathcal{D}'$.*

THEOREM 3.10. *$\widetilde{A} = S_{\mathrm{r}}^{-1}AS_{\mathrm{c}}$ is a finest-possible decomposition of A if and only if $S_{\mathrm{c}}\psi(\mathcal{D}(\widetilde{A}))$ is a maximal distributive sublattice of $\mathcal{L}(A, \Pi, \Gamma)$.*

COROLLARY 3.11. *If $\mathcal{L}(A, \Pi, \Gamma)$ is distributive, there exists a block-triangular matrix $\widetilde{A}$, (Π, Γ)-equivalent to A, such that $\mathcal{D}(\widetilde{A}) \cong \mathcal{L}(A, \Pi, \Gamma)$.*

THEOREM 3.12. *Assume F is an infinite field. If $\mathcal{L}(A, \Pi, \Gamma)$ is not distributive, there exist infinitely many maximal distributive sublattices of $\mathcal{L}(A, \Pi, \Gamma)$.*

3.3. Submodular function and the modular lattice. In this section we show another way to derive the PE-irreducible decomposition by using a certain submodular function p. The rank of a partitioned matrix is expressed as the minimum value of p, and the PE-irreducible decomposition is then derived from the decomposition of the submodular function p into minors along a maximal chain of the lattice formed by the minimizers of p. This is a direct extension of the approach that led to the previously known cases such as the DM-decomposition, the CCF of layered mixed matrices, and the decomposition of multilayered matrices. (See Remarks 3.1 and 3.2 at the end of this section for a more specific account.) As we have already seen, a proper block-triangularization of (A, Π, Γ) exists if and only if $\mathcal{L}(A, \Pi, \Gamma)$ is nonempty. We show that $\mathcal{L}(A, \Pi, \Gamma)$, when nonempty, agrees with the lattice of the minimizers of p and discuss some conditions for $\mathcal{L}(A, \Pi, \Gamma)$ to be nonempty.

Let $\mathcal{W}$ be the modular lattice, which consists of subspaces W of V such that

$$\Gamma_\beta W \subseteq W \quad \text{for} \quad \beta = 1, \ldots, \nu.$$

Put

$$p(W) := \sum_{\alpha=1}^{\mu} \dim \Pi_\alpha AW - \dim W, \qquad W \in \mathcal{W}.$$

LEMMA 3.13. *The function* $p : \mathcal{W} \to \mathbb{Z}$ *is submodular*:

$$p(W_1 + W_2) + p(W_1 \cap W_2) \leq p(W_1) + p(W_2), \qquad W_j \in \mathcal{W} \ (j = 1, 2).$$

Proof. It is sufficient to show that $\dim \Pi_\alpha AW$ is submodular. It follows from $\Pi_\alpha A(W_1 + W_2) = \Pi_\alpha AW_1 + \Pi_\alpha AW_2$ and $\Pi_\alpha A(W_1 \cap W_2) \subseteq \Pi_\alpha AW_1 \cap \Pi_\alpha AW_2$ that

$$\begin{aligned}\dim \Pi_\alpha AW_1 + \dim \Pi_\alpha AW_2 &= \dim(\Pi_\alpha AW_1 + \Pi_\alpha AW_2) + \dim(\Pi_\alpha AW_1 \cap \Pi_\alpha AW_2) \\ &\geq \dim \Pi_\alpha A(W_1 + W_2) + \dim \Pi_\alpha A(W_1 \cap W_2). \qquad \square\end{aligned}$$

Since $\sum_{\alpha=1}^{\mu} \Pi_\alpha = I_m$, we have $\bigoplus_{\alpha=1}^{\mu} \Pi_\alpha AW \supseteq AW$, in general, which implies $p(W) \geq \dim AW - \dim W \geq -\dim \mathrm{Ker} A$. Hence

$$-\dim \mathrm{Ker} A \leq \min_{W \in \mathcal{W}} p(W).$$

By adding n to the both sides, we obtain the following lemma.

LEMMA 3.14. *It holds that*

$$\mathrm{rank} A \leq n + \min_{W \in \mathcal{W}} p(W).$$

It is well known that the set of the minimizers of a submodular function on a lattice is a sublattice [24] and that a sublattice of a modular lattice is also modular [2], [5]. Therefore

$$\mathcal{L}(p) := \{W \in \mathcal{W} \mid p(W) = \min_{W' \in \mathcal{W}} p(W')\}$$

is a modular lattice.

THEOREM 3.15. *There exists a proper block-triangular matrix that is* (Π, Γ) *equivalent to* A*, if and only if* $\mathrm{rank} A = n + \min_{W \in \mathcal{W}} p(W)$*. Then* $\mathcal{L}(A, \Pi, \Gamma)$ *is nonempty and* $\mathcal{L}(A, \Pi, \Gamma) \cong \mathcal{L}(p)$.

Proof. Recall that $p(W) \geq \dim AW - \dim W \geq -\dim \mathrm{Ker} A$. The first inequality holds with equality if and only if $\Pi_\alpha AW \subseteq AW$ is satisfied for $\alpha = 1, \ldots, \mu$, and the second inequality holds with equality if and only if $\mathrm{Ker} A \subseteq W$. Therefore $p(W) = -\dim \mathrm{Ker} A$ if and only if $W \in \mathcal{L}(A, \Pi, \Gamma)$. That is to say $\mathrm{rank} A = n + \min p(W)$ if and only if $\mathcal{L}(A, \Pi, \Gamma)$ is nonempty. In this case $\mathcal{L}(p) \cong \mathcal{L}(A, \Pi, \Gamma)$. Then Theorem 3.5 completes the proof. $\square$

When $\mathrm{rank} A = \min(m, n)$ holds, we see that $\min p(W) \leq \min(p(0), p(V)) \leq \min(0, m - n) = \mathrm{rank} A - n$, which implies the equality in Lemma 3.14. This, together with Theorem 3.15, gives an alternative proof to Lemma 3.2.

Another sufficient condition for the existence of a proper block-triangularization is given in terms of the genericity of a partitioned matrix. The genericity refers to the situation where the nonzero entries of A can be regarded as independent parameters. Let K be the prime field of F; in particular, if $F = \mathbb{R}$ or $\mathbb{C}$, K is the field of rational numbers $\mathbb{Q}$. Then we have the following theorem.

THEOREM 3.16. *If the set of nonzero entries of* A *is algebraically independent over* K*, then*

$$\mathrm{rank} A = n + \min_{W \in \mathcal{W}} p(W),$$

and, as a consequence, there exists a proper block-triangular matrix (Π, Γ)*-equivalent to* A.

Proof. Consider the DM-decomposition of (the bipartite graph associated with) A, and assume that the column set C and the row set R are split into blocks $(\overline{C}_0; \overline{C}_1, \ldots, \overline{C}_d; \overline{C}_\infty)$ and $(\overline{R}_0; \overline{R}_1, \ldots, \overline{R}_d; \overline{R}_\infty)$, where

$$\begin{array}{lll} |\overline{R}_0| < |\overline{C}_0| & \text{or} & |\overline{R}_0| = |\overline{C}_0| = 0, \\ |\overline{R}_k| = |\overline{C}_k| > 0 & \text{for} & k = 1, \ldots, d, \\ |\overline{R}_\infty| > |\overline{C}_\infty| & \text{or} & |\overline{R}_\infty| = |\overline{C}_\infty| = 0, \end{array}$$

and

$$A[\overline{R}_k, \overline{C}_l] = O \quad \text{if} \quad 0 \leq l < k \leq \infty.$$

Since all the nonzero entries of A are algebraically independent over K, we have

$$\mathrm{rank} A[\overline{R}_k, \overline{C}_k] = \min(|\overline{R}_k|, |\overline{C}_k|) \quad \text{for} \quad k = 0, 1, \ldots, d, \infty.$$

Let $\overline{W}$ be the subspace of V corresponding to the column subset $\overline{C}_0$. It is clear that $\Gamma_\beta \overline{W} \subseteq \overline{W}$ for $\beta = 1, \ldots, \nu$, which implies $\overline{W} \in \mathcal{W}$. Since $\mathrm{rank} A[\overline{R}_0, \overline{C}_0] = |\overline{R}_0|$, it is also clear that

$$p(\overline{W}) = |\overline{R}_0| - |\overline{C}_0| = -\dim \mathrm{Ker} A. \qquad \square$$

This theorem implies in particular that "almost all" matrices (in the case of $F = \mathbb{C}$ or $\mathbb{R}$) have a proper block-triangularization.

Example 3.2. Consider the following 4×4 partitioned matrix A:

$$A = \begin{array}{c} \\ \star \\ \star \\ \diamond \\ \diamond \end{array} \begin{array}{c} \begin{array}{cccc} \circ & \circ & \bullet & \bullet \end{array} \\ \left[\begin{array}{cc|cc} t_1 & 0 & 0 & t_2 \\ 0 & t_3 & t_4 & 0 \\ \hline 0 & t_5 & 0 & t_6 \\ t_7 & 0 & t_8 & 0 \end{array}\right] \end{array},$$

where $\mu = 2$, $\nu = 2$, $m_1 = 2$, $m_2 = 2$, $n_1 = 2$, and $n_2 = 2$. Suppose that $\{t_i \mid i = 1, \ldots, 8; 0 < t_i \in \mathbb{R}\}$ is algebraically independent over the field $\mathbb{Q}$. Let S_r and S_c be the nonsingular matrices:

$$S_\mathrm{r} = \left[\begin{array}{cc|cc} t_1\sqrt{t_2t_3t_5t_8} & t_1\sqrt{t_2t_3t_5t_8} & \multicolumn{2}{c}{\multirow{2}{*}{O}} \\ -t_3\sqrt{t_1t_4t_6t_7} & t_3\sqrt{t_1t_4t_6t_7} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{O}} & -t_5\sqrt{t_1t_4t_6t_7} & t_5\sqrt{t_1t_4t_6t_7} \\ & & t_7\sqrt{t_2t_3t_5t_8} & t_7\sqrt{t_2t_3t_5t_8} \end{array}\right],$$

$$S_\mathrm{c} = \left[\begin{array}{cc|cc} \sqrt{t_2t_3t_5t_8} & \sqrt{t_2t_3t_5t_8} & \multicolumn{2}{c}{\multirow{2}{*}{O}} \\ -\sqrt{t_1t_4t_6t_7} & \sqrt{t_1t_4t_6t_7} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{O}} & t_3t_7\sqrt{t_1t_2t_5t_6} & t_3t_7\sqrt{t_1t_2t_5t_6} \\ & & -t_1t_5\sqrt{t_3t_4t_7t_8} & t_1t_5\sqrt{t_3t_4t_7t_8} \end{array}\right].$$

Then we have

$$\widetilde{A} = S_\mathrm{r}^{-1} A S_\mathrm{c} = \begin{array}{c} \\ \star \\ \star \\ \diamond \\ \diamond \end{array} \begin{array}{c} \begin{array}{cccc} \circ & \circ & \bullet & \bullet \end{array} \\ \left[\begin{array}{cc|cc} 1 & 0 & -\sqrt{t_2t_4t_5t_7} & 0 \\ 0 & 1 & 0 & \sqrt{t_2t_4t_5t_7} \\ \hline 1 & 0 & \sqrt{t_1t_3t_6t_8} & 0 \\ 0 & 1 & 0 & \sqrt{t_1t_3t_6t_8} \end{array}\right] \end{array}.$$

By using the permutation matrices

$$P_{\mathrm{r}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad P_{\mathrm{c}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we can transform $\widetilde{A}$ into an explicit upper block-triangular form:

$$\bar{A} = P_{\mathrm{r}}\widetilde{A}P_{\mathrm{c}} = \begin{array}{c} \\ \star \\ \diamond \\ \star \\ \diamond \end{array} \begin{array}{c} \begin{array}{cccc} \circ & & \bullet & \end{array} \\ \left[\begin{array}{cc|cc} 1 & -\sqrt{t_2t_4t_5t_7} & \multicolumn{2}{c}{\multirow{2}{*}{O}} \\ 1 & \sqrt{t_1t_3t_6t_8} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{O}} & 1 & \sqrt{t_2t_4t_5t_7} \\ & & 1 & \sqrt{t_1t_3t_6t_8} \end{array}\right] \end{array} \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \circ \\ \\ \bullet \\ \end{array}.$$

Then $\widetilde{A}$ is in a proper block-triangular form with two blocks. If we had an algebraic relation: $t_1t_3t_6t_8 = t_2t_4t_5t_7$, then $\widetilde{A}$ obtained above would not be in a *proper* block-triangular form.

Remark 3.1. As stated in Theorem 3.15, a proper block-triangularization exists if and only if $\mathrm{rank}A = n + \min_{W \in \mathcal{W}} p(W)$. The PE-irreducible decomposition can be constructed also from the decomposition of the submodular function p into minors along a maximal chain of the lattice $\mathcal{L}(p)$, just as has been done for the previously known cases such as the DM-decomposition. In particular, the rank condition: $\mathrm{rank}\tilde{A}[R_k, C_k] = \min(|R_k|, |C_k|)$ follows from the above identity when combined with the fact that the submodular function is modular when restriced onto the sublattice $\mathcal{L}(p)$. However, this approach based on a submodular function fails to capture such complications arising from identical diagonal blocks as indicated in Example 2.3.

Remark 3.2. The block-triangularization under the PE-transformation includes the following previously known cases: the DM-decomposition [8], [10], [19], the CCF of layered mixed matrices [20], [22], [23], and the decomposition of multilayered matrices [20], [23]. Not only is the PE-transformation an extension of the admissible transformations used in those decompositions, but also the submodular function p introduced above is an extension of the submodular functions used in their constructions. In all those cases, the column set $C = \mathrm{Col}(A)$ is partitioned into singletons (with $\nu = n$), and A takes the form:

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_\mu \end{bmatrix},$$

where $A_\alpha = [A_{\alpha 1}, A_{\alpha 2}, \ldots, A_{\alpha n}]$ is the $m_\alpha \times n$ submatrix of A. Such a matrix A is termed a multilayered matrix in [20], [23]. In this case $\mathcal{W}$ is isomorphic, as a lattice, to the boolean lattice 2^C, and p can be identified with the submodular function $\tilde{p} : 2^C \to \mathbb{Z}$ defined by

$$\tilde{p}(J) = \sum_{\alpha=1}^{\mu} \mathrm{rank}A_\alpha[\mathrm{Row}(A_\alpha), J] - |J|, \qquad J \subseteq C,$$

where $\mathrm{Row}(A_\alpha)$ denotes the row set of A_α. This agrees with the submodular function used in the decomposition of multilayered matrices.

The notion of layered mixed matrix, introduced for the combinatorial-structural analysis of discrete systems, is defined as follows. Let $K \subseteq F$ be fields (K is not assumed to be a prime field). A matrix A is called a layered mixed matrix with respect to F/K if it can be put into the form

$$A = \begin{bmatrix} Q \\ T \end{bmatrix}$$

by rearranging the rows, where (i) the entries of Q belong to K, and (ii) the entries of T belong to F and the set of the nonzero entries of T is algebraically independent over K. The admissible transformation for $A = \left[\begin{smallmatrix} Q \\ T \end{smallmatrix}\right]$ is defined as an equivalence transformation of the form

$$P_{\mathrm{r}} \begin{bmatrix} S_Q & O \\ O & I \end{bmatrix} \begin{bmatrix} Q \\ T \end{bmatrix} P_{\mathrm{c}},$$

where S_Q is a nonsingular matrix over the subfield K and I denotes the identity matrix of size $|\mathrm{Row}(T)| = m_{\mathrm{T}}$. It is known that there exists a unique finest-possible proper block-triangular matrix under this transformation. This is the CCF. We can regard A as a multilayered matrix with $\mu = m_{\mathrm{T}} + 1$, $m_\alpha = 1$ for $\alpha = 2, \ldots, \mu$, where $Q = A_1$ and

$$T = \begin{bmatrix} A_2 \\ \vdots \\ A_\mu \end{bmatrix}.$$

Then the admissible transformation above is essentially the same as the PE-transformation of this paper, in which $S_{\mathrm{r}} = \mathrm{diag}[S_Q, 1, \ldots, 1]$ and $S_{\mathrm{c}} = \mathrm{diag}[1, \ldots, 1]$. The function $\tilde{p}(J)$ can be rewritten as follows. Denoting by $\gamma(J)$ the number of nonzero rows in the submatrix $T[\mathrm{Row}(T), J]$, we see

$$\gamma(J) = \sum_{\alpha=2}^{\mu} \mathrm{rank} A_\alpha[\mathrm{Row}(A_\alpha), J].$$

Hence, putting $\rho(J) = \mathrm{rank} Q[\mathrm{Row}(Q), J]$, we obtain

$$\tilde{p}(J) = \rho(J) + \gamma(J) - |J|,$$

which agrees with the function used in the construction of the CCF. Note that the assumed algebraic independence in the T-part guarantees the equality in Lemma 3.14.

Finally, the DM-decomposition is a special case of the CCF in which Q-part is empty. In other words, $\tilde{p}(J) = \gamma(J) - |J|$, which is sometimes referred to as the surplus function [19], where $\gamma(J)$ is now defined as the number of nonzero rows in the submatrix $A[\mathrm{Row}(A), J]$.

For those special cases, efficient algorithms have been constructed based on the results from network (submodular) flow theory. Namely, the family of the minimizers, as well as the minimum value, of the submodular function $\tilde{p}$ can be computed efficiently. It remains still open whether the algorithms can be extended to the decomposition of a general partitioned matrix. The extension will involve the minimization of a submodular function over a modular (nondistributive) lattice.

4. Discussions.

4.1. Decomposition algorithm for partitioned square matrices. In this section we discuss an algorithm for the block-triangularization of a partitioned square matrix (A, Π) under PS-transformations in the case of $F = \mathbb{C}$. The following simple algorithm gives a block-triangular matrix Π-similar to (A, Π). Furthermore, if $A_{\alpha\alpha}$ has distinct eigenvalues for $\alpha = 1, \ldots, \nu$, this algorithm gives the finest decomposition of (A, Π) whose uniqueness has been guaranteed by Corollary 2.9 and Lemma 2.10.

ALGORITHM FOR THE BLOCK-TRIANGULARIZATION OF (A, Π).

Step 1. Let S_α be a nonsingular matrix that transforms $A_{\alpha\alpha}$ into its Jordan normal form. Put

$$S := \bigoplus_{\alpha=1}^{\nu} S_\alpha, \qquad \widehat{A} := S^{-1}AS.$$

Step 2. Let $\widehat{G} = (\widehat{V}, \widehat{E})$ be a directed graph that represents the zero-nonzero pattern of the matrix $\widehat{A}$, where the vertex set $\widehat{V}$ is identified with the column set C of $\widehat{A}$ as well as with the row set R, and where the arc set $\widehat{E}$ is defined by $\widehat{E} := \{(j, i) \mid [\widehat{A}]_{ij} \neq 0\}$. Find the strongly connected component decomposition $(\widehat{G}_1, \ldots, \widehat{G}_b)$ of $\widehat{G}$, where the strongly connected components are indexed in such a way that there does not exist a path from the component $\widehat{G}_l$ to $\widehat{G}_k$ if $1 \leq l < k \leq b$. Accordingly, decompose C and R into the blocks $(C_1, \ldots, C_b)$ and $(R_1, \ldots, R_b)$, respectively.

When $A_{\alpha\alpha}$ has distinct eigenvalues, every element of $\mathcal{L}(A, \Pi)$ is spanned by certain eigenvectors of $A_{\alpha\alpha}$ for $\alpha = 1, \ldots, \nu$ (see the proof of Lemma 2.10). The column vectors of S_α are the eigenvectors of $A_{\alpha\alpha}$ for $\alpha = 1, \ldots, \nu$. Therefore it follows from Lemma 2.5 that

$$\mathcal{D}(\widehat{A}) \cong \mathcal{L}(A, \Pi),$$

which implies $\widehat{A}$ is the finest decomposition of (A, Π). Thus we have the following theorem.

THEOREM 4.1. *Suppose that for each $\alpha = 1, \ldots, \nu$, $A_{\alpha\alpha}$ has n_α distinct eigenvalues. Then the output $\widehat{A}$ of the above algorithm is the finest decomposition of (A, Π) with $\mathcal{D}(\widehat{A}) \cong \mathcal{L}(A, \Pi)$.*

Remark 4.1. In the case of $F = \mathbb{R}$, the above algorithm, in which the complex Jordan normal form is replaced by real Jordan normal form, works correctly if $A_{\alpha\alpha}$ has n_α distinct complex eigenvalues for $\alpha = 1, \ldots, \nu$ because every element of $\mathcal{L}(A, \Pi)$ is spanned by certain eigenvectors or pairs of vectors (in a certain two-dimensional eigenspace), corresponding to real or nonreal eigenvalues, respectively.

4.2. Algorithm for proper block-triangularizability. We have already seen in §3 that a partitioned matrix (A, Π, Γ) can be transformed into a proper block-triangular form by a PE-transformation if and only if $\mathcal{L}(A, \Pi, \Gamma)$ is nonempty. In this subsection we give an algorithm to determine whether $\mathcal{L}(A, \Pi, \Gamma)$ is empty or nonempty by searching for the minimum element $V_\perp$ of $\mathcal{L}(A, \Pi, \Gamma)$.

Construct an alternating sequence $V^{(0)}, U^{(1)}, V^{(1)}, U^{(2)}, \ldots$, of subspaces $V^{(\kappa)} \subseteq$

V and $U^{(\kappa)} \subseteq U$ by

$$V^{(0)} := 0,$$
$$U^{(\kappa)} := \bigoplus_{\alpha=1}^{\mu} \pi_\alpha(f(V^{(\kappa-1)})), \quad V^{(\kappa)} := \bigoplus_{\beta=1}^{\nu} \gamma_\beta(f^{-1}(U^{(\kappa)})) \quad (\kappa = 1, 2, \ldots).$$

Then we have

$$V^{(\kappa+1)} \supseteq f^{-1}(U^{(\kappa+1)}) \supseteq f^{-1}(f(V^{(\kappa)})) \supseteq V^{(\kappa)} \quad (\kappa = 0, 1, \ldots).$$

This implies $V^{(0)} \subseteq V^{(1)} \subseteq \cdots$ and $U^{(1)} \subseteq U^{(2)} \subseteq \cdots$. Furthermore, if $V^{(\kappa)} = V^{(\kappa+1)}$ or $U^{(\kappa)} = U^{(\kappa+1)}$, then $V^{(\kappa)} = V^{(\kappa+1)} = V^{(\kappa+2)} = \cdots$ and $U^{(\kappa+1)} = U^{(\kappa+2)} = \cdots$. There exists a number $q \leq \min(n, m) + 1$ such that $V^{(q)} = V^{(q+1)} = \cdots$ and $U^{(q)} = U^{(q+1)} = \cdots$, where $n = \dim V$ and $m = \dim U$. Thus we can obtain $V^{(\infty)}$ and $U^{(\infty)}$ within finite steps.

THEOREM 4.2. *$\mathcal{L}(A, \Pi, \Gamma)$ is nonempty if and only if $U^{(\infty)} \subseteq \mathrm{Im} f$.*

Proof. ($\Longrightarrow$) We prove $V^{(\kappa)} \subseteq V_\perp$ by induction. Obviously $V^{(0)} \subseteq V_\perp$. Suppose $V^{(\kappa)} \subseteq V_\perp$. Then, since $\gamma_\beta(V_\perp) \subseteq V_\perp$, $\pi_\alpha(f(V_\perp)) \subseteq f(V_\perp)$, and $\mathrm{Ker} f \subseteq V_\perp$, we have $V^{(\kappa+1)} = \bigoplus_{\beta=1}^{\nu} \gamma_\beta(f^{-1}(\bigoplus_{\alpha=1}^{\mu} \pi_\alpha(f(V^{(\kappa)})))) \subseteq V_\perp$. Hence $V^{(\kappa)} \subseteq V_\perp$ for $\kappa = 0, 1, \ldots$. This implies $U^{(\kappa+1)} = \bigoplus_{\alpha=1}^{\mu} \pi_\alpha(f(V^{(\kappa)})) \subseteq \bigoplus_{\alpha=1}^{\mu} \pi_\alpha(f(V_\perp)) \subseteq f(V_\perp) \subseteq \mathrm{Im} f$.

($\Longleftarrow$) We check the three conditions in §3.2 for $V^{(\infty)} \in \mathcal{L}(A, \Pi, \Gamma)$. (i) $\mathrm{Ker} f \subseteq V^{(\infty)}$ follows from $\mathrm{Ker} f \subseteq V^{(1)}$. (ii) It follows from $V^{(\infty)} = \bigoplus_{\beta=1}^{\nu} \gamma_\beta(f^{-1}(U^{(\infty)}))$ that $\gamma_\beta(V^{(\infty)}) = \gamma_\beta(f^{-1}(U^{(\infty)}))$. Then we have $\gamma_\beta(V^{(\infty)}) \subseteq \bigoplus_{\beta=1}^{\nu} \gamma_\beta(V^{(\infty)}) = V^{(\infty)}$. (iii) First note that $U^{(\infty)} \subseteq \mathrm{Im} f$ implies $f(f^{-1}(U^{(\infty)})) = U^{(\infty)}$. By applying f to $V^{(\infty)} = \bigoplus_{\beta=1}^{\nu} \gamma_\beta(f^{-1}(U^{(\infty)})) \supseteq f^{-1}(U^{(\infty)})$, we see $f(V^{(\infty)}) \supseteq f(f^{-1}(U^{(\infty)})) = U^{(\infty)} = \bigoplus_{\alpha=1}^{\mu} \pi_\alpha(f(V^{(\infty)})) \supseteq \pi_\alpha(f(V^{(\infty)}))$. □

Thus we have obtained a constructive procedure to determine whether or not (A, Π, Γ) is properly block-triangularizable. As is evident from the proof, $V^{(\infty)} = V_\perp$.

Remark 4.2. We can start the above procedure with an arbitrary subspace $V^{(0)}$ of V. In this case, the set $\mathcal{L}^* = \{W \mid V^{(0)} \subseteq W \in \mathcal{L}(A, \Pi, \Gamma)\}$ is nonempty if and only if $U^{(\infty)} \subseteq \mathrm{Im} f$. And $V^{(\infty)}$ gives the minimum element in $\mathcal{L}^*$.

Acknowledgments. Example 3.2 is credited to T. Fujita, who completed his graduation thesis at the University of Tokyo under Murota's supervision. The authors are grateful to Professor Masanori Fushimi for his support and encouragement.

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.
[2] M. AIGNER, *Combinatorial Theory*, Springer-Verlag, Berlin, 1979.
[3] M. A. ARBIB, *Theories of Abstract Automata*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1969.
[4] C. BERGE, *Graphes et Hypergraphes*, Dunod, Paris, 1970.
[5] G. BIRKHOFF, *Lattice Theory*, 3rd ed., American Math. Soc., Providence, RI, 1979.
[6] D. BLACKWELL AND L. KOOPMANS, *On the identifiability problem for functions of finite Markov chains*, Ann. Math. Statist., 28 (1957), pp. 1011–1015.
[7] N. BOURBAKI, *Éléments de Mathématique*; *Livre* II; *Algèbre*, Chap. 1, 6, Hermann, Paris, 1962.
[8] R. A. BRUALDI AND H. J. RYSER, *Combinatorial Matrix Theory*, Cambridge University Press, London, 1991.

[9] C. W. CURTIS AND I. REINER, *Representation Theory of Finite Groups and Associative Algebras*, Interscience (John Wiley), New York, 1962.
[10] A. L. DULMAGE AND N. S. MENDELSOHN, *A structure theory of bipartite graphs of finite exterior dimension*, Trans. Roy. Soc. Canada, Sec. III, 53 (1959), pp. 1–13.
[11] F. R. GANTMACHER, *The Theory of Matrices*, Chelsea, New York, 1959.
[12] A. HELLER, *On stochastic processes derived from Markov chains*, Ann. Math. Statist., 36 (1965), pp. 1286–1291.
[13] ———, *Probabilistic automata and stochastic transformations*, Math. Systems Theory, 1 (1967), pp. 197–208.
[14] M. IRI, *Applications of matroid theory*, Mathematical Programming—The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 158–201.
[15] ———, *Structural theory for the combinatorial systems characterized by submodular functions*, Progress in Combinatorial Optimization, W. R. Pulleyblank, ed., Academic Press, New York, 1984, pp. 197–219.
[16] H. ITO, *An Algebraic Study on Discrete Stochastic Systems*, Ph.D. thesis, Dept. of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, 1992.
[17] H. ITO, S. AMARI, AND K. KOBAYASHI, *Identifiability of hidden Markov information sources and their minimum degrees of freedom*, IEEE Trans. Inform. Theory, IT-38 (1992), pp. 324–333.
[18] R. E. KALMAN, P. L. FALB, AND M. A. ARBIB, *Topics in Mathematical System Theory*, McGraw-Hill, New York, 1969.
[19] L. LOVÁSZ AND M. PLUMMER, *Matching Theory*, North-Holland, Amsterdam, 1986.
[20] K. MUROTA, *Systems Analysis by Graphs and Matroids—Structural Solvability and Controllability*, Algorithms and Combinatorics, Vol. 3, Springer-Verlag, Berlin, 1987.
[21] ———, *Hierarchical decomposition of symmetric discrete systems by matroid and group theories*, Math. Programming, Ser. A, 59 (1993), pp. 377–404.
[22] ———, *Mixed matrices—Irreducibility and decomposition*, Combinatorial and Graph-Theoretical Problems in Linear Algebra, R. A. Brualdi, S. Friedland, and V. Klee, eds., The IMA Volumes in Mathematics and Its Applications, Vol. 50, Springer-Verlag, 1993, pp. 39–71.
[23] K. MUROTA, M. IRI, AND M. NAKAMURA, *Combinatorial canonical form of layered mixed matrices and its application to block-triangularization of systems of equations*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 123–149.
[24] O. ORE, *Studies on directed graphs* I, Ann. of Math., 63 (1956), pp. 383–406.
[25] A. RECSKI, *Matroid Theory and Its Applications in Electric Network Theory and in Statics*, Springer-Verlag, Berlin, 1989.
[26] R. ROSENBLATT, *Markov Processes—Structure and Asymptotic Behavior*, Springer-Verlag, Berlin, 1971.
[27] D. D. ŠILJAK, *Decentralized Control of Complex Systems*, Academic Press, New York, 1991.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1256–1265, October 1994

013

NUMERICAL RANGE OF MATRIX POLYNOMIALS*

CHI-KWONG LI† AND LEIBA RODMAN†

Abstract. Let M_n be the algebra of all $n \times n$ complex matrices. Suppose

$$P(\lambda) = A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_0$$

is a matrix polynomial, where $A_i \in M_n$ and λ is a complex variable. The *numerical range* of $P(\lambda)$ is defined as

$$W(P(\lambda)) = \{\mu \in \mathbb{C} : x^*P(\lambda)x = 0 \text{ for some nonzero } x \in \mathbb{C}^n\}.$$

The numerical range of matrix polynomials has important applications to overdamped vibration systems with finite number of degrees of freedom and it is also related to stability theory. In this paper, the subject is studied systematically. The emphasis is on the relationship between the geometrical properties of $W(P(\lambda))$ and the algebraic and analytic properties of $P(\lambda)$. A factorization result, based on geometric properties of $W(P(\lambda))$ for certain classes of matrix polynomials with not necessarily hermitian coefficients is proved, and the set $W(P(\lambda))$ for a linear polynomial with hermitian matrices as coefficients is studied in detail. The results indicate that the information on $W(P(\lambda))$ is very useful in understanding matrix polynomials and also reflects the fact that it is highly nontrivial to give a complete description of the set $W(P(\lambda))$.

Key words. numerical range, matrix polynomial, factorizations

AMS subject classifications. 15A60, 15A22, 47A56

1. Introduction.

Let M_n be the algebra of all $n \times n$ complex matrices. Suppose

$$P(\lambda) = A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_0$$

is a matrix polynomial, where $A_i \in M_n$ and λ is a complex variable. Define the *numerical range* of $P(\lambda)$ as

$$W(P(\lambda)) = \{\mu \in \mathbb{C} : x^*P(\lambda)x = 0 \text{ for some nonzero } x \in \mathbb{C}^n\}.$$

If $P(\lambda) = \lambda I - A$, then $W(P(\lambda))$ reduces to the *numerical range* of A defined and denoted by

$$W(A) = \{x^*Ax : x \in \mathcal{S}\},$$

where $\mathcal{S}$ denotes the unit sphere in $\mathbb{C}^n$, i.e.,

$$\mathcal{S} = \{x \in \mathbb{C}^n : x^*x = 1\}.$$

In this sense, the numerical range of a matrix polynomial is a generalization of the classical numerical range.

One important application where the numerical range of matrix polynomials plays a significant role is overdamped vibration systems with a finite number of degrees of

*Received by the editors February 15, 1993; accepted for publication July 4, 1993. The research of the first author was supported by National Science Foundation grant DMS-91-00344 and the research of the second author was supported by National Science Foundation grant DMS-91-23841.

†Department of Mathematics, College of William and Mary, Williamsburg, Virginia 23187-8795 (ckli@cs.wm.edu and lxrodm@cs.wm.edu).

freedom, e.g., see [1], [6, Chap. 7], and [2, Chap. 10]. The matrix polynomial there is of the form $A_2\lambda^2 + A_1\lambda + A_0$, where A_0, A_1, A_2 are $n \times n$ positive definite matrices. Factorizations of such a polynomial of the form $A_2(\lambda I - Y_1)(\lambda I - Y_2)$, which are based on the properties of its numerical range, are crucial in the theory of overdamped vibration systems ([5] is a pioneering work in this direction). An extension of these ideas to the matrix and operator polynomials of arbitrary degree leads to the notion of hyperbolic polynomials and to important factorization results (quoted in Theorem 3.1) based on the properties of the numerical range. In fact, Theorem 3.1 provided an impetus for this work.

Another motivation for the study of numerical ranges of matrix polynomials arises from stability theory. Recall that a scalar polynomial $p(\lambda)$ is said to be *stable* if all its roots have negative real parts. In terms of the numerical range of $p(\lambda)$, stability means that $W(p(\lambda)) \subseteq \{\nu \in \mathbb{C} : \mathrm{Re}(\nu) < 0\}$. Given n stable scalar polynomials, $p_1(\lambda), \ldots, p_n(\lambda)$, it is of interest to know whether all the polynomials $q(\lambda)$ obtained by taking convex combination of the n given polynomials, i.e., $q(\lambda) = \sum_{i=1}^{n} \eta_i p(\lambda)$ with $\eta_i \geq 0$ and $\sum \eta_i = 1$, are also stable. Let m be the highest degree of the polynomials $p_i(\lambda)$, and for $j = 0, \ldots, m$ let $A_j \in M_n$ be the diagonal matrix whose ith diagonal entry equals the coefficient of λ^j in $p_i(\lambda)$. If $P(\lambda) = \sum_{j=1}^{m} A_j \lambda^j$, then one easily checks that $q(x)$ is a convex combination of $p_i(\lambda)$'s if and only if $q(x)$ is of the form $x^*P(\lambda)x$ with $x \in \mathcal{S}$. Thus all convex combinations of the polynomials $p_i(\lambda)$'s are stable if and only if $W(P(\lambda)) \subseteq \{\nu \in \mathbb{C} : \mathrm{Re}(\nu) < 0\}$. Thus, results on stable polynomials can be translated to results about $W(P(\lambda))$ and vice versa.

Numerical ranges of matrix polynomials have appeared in the literature (see [2, §10.6]), but have not been studied systematically. We believe that this paper is the first systematic study of the subject. Here we consider the following general problem.

Study the relationship between the geometrical properties of $W(P(\lambda))$ and the algebraic and analytic properties of $P(\lambda)$.

We describe some basic properties of $W(P(\lambda))$ in §2. A factorization result, based on geometric properties of $W(P(\lambda))$ for certain classes of matrix polynomials with not necessarily hermitian coefficients is proved in §3. In §4, we study the set $W(P(\lambda))$ for a linear polynomial with hermitian matrices as coefficients. The results indicate that the information on $W(P(\lambda))$ is very useful in understanding matrix polynomials and also reflects the fact that it is highly nontrivial to give a complete description of the set $W(P(\lambda))$. We denote the usual Euclidean norm on $\mathbb{C}^n$ by $\|x\| = (x^*x)^{1/2}$.

2. General properties. The following properties of $W(P(\lambda))$ can be verified readily.

PROPOSITION 2.1. *Suppose $P(\lambda) = A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_0$, where $A_m \neq 0$.*

(a) *$W(P(\lambda))$ is closed in $\mathbb{C}$.*

(b) *For any $\alpha \in \mathbb{C}$, $W(P(\lambda + \alpha)) = W(P(\lambda)) - \alpha$.*

(c) *If $Q(\lambda) = \sum_{j=0}^{m} \lambda^j A_{m-j}$, then $W(Q(\lambda)) \setminus \{0\} = \{\mu^{-1} : \mu \in W(P(\lambda)),\ \mu \neq 0\}$.*

(d) *For any $n \times r$ matrix S with* rank *r, $r \leq n$, $W(S^*P(\lambda)S) \subseteq W(P(\lambda))$. Equality holds if $r = n$.*

(e) *If A_i, $i = 0, \ldots, m$, have a common nonzero isotropic vector x, i.e., $x^*A_ix = 0$ for all i, then $W(P(\lambda)) = \mathbb{C}$.*

It is well known, e.g., see [4, §1.3], that the classical numerical range $W(A)$ is always a compact convex set. Condition (a) of Proposition 2.1 shows that $W(P(\lambda))$ is closed. However, $W(P(\lambda))$ need not be connected or bounded, as shown in the following examples.

Example 1. Let $P(\lambda) = \lambda^m \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right) - I$. Then

$$\begin{aligned} W(P(\lambda)) &= \{z \in \mathbb{C} : z^m q - 1 = 0 \text{ for some } q \in [-1,1]\} \\ &= \{re^{i\theta} : r \geq 1;\ \theta = k\pi/m,\ k = 0,1,\ldots,2m-1\} \end{aligned}$$

has $2m$ unbounded connected components.

One may wonder whether a connected component of $W(P(\lambda))$ is convex. The following example shows that it need not be.

Example 2. Let $P(\lambda) = \lambda^m I - \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. Then

$$\begin{aligned} W(P(\lambda)) &= \{z \in \mathbb{C} : z^m - q = 0 \text{ for some } q \in [-1,1]\} \\ &= \{re^{i\theta} : 0 \leq r \leq 1;\ \theta = k\pi/m,\ k = 0,1,\ldots,2m-1\} \end{aligned}$$

has a single nonconvex component.

In the following, we obtain results concerning the number of connected components and the boundedness of $W(P(\lambda))$.

Using the convexity of $W(B)$, it is easy to see that $W(B) \setminus \{0\}$ is disconnected if and only if μB is an indefinite hermitian matrix for some $\mu \in \mathbb{C}$. If this happens, then $W(B)$ is a line segment passing through the origin in $\mathbb{C}$ and hence $W(B) \setminus \{0\}$ has two components. We are now ready to state our result concerning the number of connected components of $W(P(\lambda))$.

THEOREM 2.2. *Let* $P(\lambda) = A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_0$, *where* $A_m \neq 0$. *Suppose* $W(P(\lambda))$ *has* r *connected components.*

(a) *Suppose* $W(A_m) \setminus \{0\}$ *is connected, i.e.,* μA_m *is not an indefinite hermitian matrix for all* $\mu \in \mathbb{C}$, *and suppose* s *is the minimum number of distinct zeros of the polynomial* $x^*P(\lambda)x$ *over all* $x \in \mathcal{S}$ *such that* $x^*A_m x \neq 0$. *Then*

$$r \leq s \leq m.$$

(b) *Suppose* $W(A_m)\setminus\{0\}$ *has disjoint connected components* $\mathcal{C}_1$ *and* $\mathcal{C}_2$, *i.e.,* μA_m *is an indefinite hermitian matrix for some* $\mu \in \mathbb{C}$, *and suppose* s_i *is the minimum number of distinct zeros of the polynomial* $x^*P(\lambda)x$ *over all* $x \in \mathcal{S}$ *such that* $x^*A_m x \in \mathcal{C}_i$ *for* $i = 1,2$. *Then*

$$r \leq s_1 + s_2 \leq 2m.$$

Proof. Consider one connected component $\mathcal{C}$ of $W(A_m) \setminus \{0\}$. Suppose $\hat{x} \in \mathcal{S}$ is such that $\hat{x}^*A_m\hat{x} \in \mathcal{C}$ and $\hat{x}P(\lambda)\hat{x} = 0$ has k distinct roots, where

$$k = \min\{\text{ number of distinct roots of } x^*P(\lambda)x = 0 : x \in \mathcal{S},\ x^*A_m x \in \mathcal{C}\}.$$

We make the following claim. If $y \in \mathcal{S}$ is such that $y^*A_m y \in \mathcal{C}$, then there exists a continuous function $z : [0,1] \to \mathcal{S}$ such that $z(0) = \hat{x}$, $z(1) = \mu y$ for some $\mu \in \mathbb{C}$ with $|\mu| = 1$, and $z(t)^*A_m z(t) \in \mathcal{C}$ for all $t \in [0,1]$.

Assume the claim has been verified. Since the solutions $\lambda_1(t),\ldots,\lambda_m(t)$ of the equation

$$z(t)^*P(\lambda)z(t) = 0$$

are continuous functions of t, the zeros of the polynomial $y^*P(\lambda)y = z(1)^*P(\lambda)z(1)$ are connected to those of $\hat{x}^*P(\lambda)\hat{x} = z(0)^*P(\lambda)z(0)$ by continuous curves in $W(P(\lambda))$. Therefore the zeros of the polynomial $y^*P(\lambda)y$ must lie in the connected components containing the zeros of the polynomial $\hat{x}^*P(\lambda)\hat{x}$. Thus there are at most k connected

components, determined by the distinct roots of $\hat{x}^*P(\lambda)\hat{x} = 0$, for the roots of those equations $y^*P(\lambda)y = 0$ with $y \in \mathcal{S}$ and $y^*A_m y \in \mathcal{C}$.

It remains to prove our claim. If y is a scalar multiple of $\hat{x}$, then a constant function z will satisfy the claim. We assume therefore that $\hat{x}$ and y are linearly independent. Let $\eta_0 = \hat{x}^*A_m\hat{x}$ and $\eta_1 = y^*A_m y$. By the discussion before the theorem, one sees that $z(t)^*A_m z(t) \in \mathcal{C}$ for all $t \in [0,1]$ is equivalent to the condition that $z(t)^*A_m z(t) \neq 0$ for all $t \in [0,1]$. To construct the required $z(t)$ we consider several cases. First, suppose $\eta_0 \neq \eta_1$ and suppose the line segment joining η_0 to η_1 does not contain zero. Let $z(t) = (\sqrt{1-t^2}\hat{x} + t\mu y)/\|\sqrt{1-t^2}\hat{x} + t\mu y\|$, where $\mu \in \mathbb{C}$ with $|\mu| = 1$ satisfies one of the following conditions:

(i) $\mu\hat{x}^*A_m y + \overline{\mu}y^*A_m\hat{x} = 0$,

(ii) $\mu\hat{x}^*A_m y + \overline{\mu}y^*A_m\hat{x} = M(\eta_1 - \eta_0)$ for some $M > 0$ if $\eta_0 + \nu(\eta_1 - \eta_0) \neq 0$ for all $\nu > 0$,

(iii) $\mu\hat{x}^*A_m y + \overline{\mu}y^*A_m\hat{x} = M(\eta_1 - \eta_0)$ for some $M < 0$ if $\eta_0 + \nu(\eta_1 - \eta_0) \neq 0$ for all $\nu < 0$.

The existence of μ satisfying (i), (ii), or (iii) can be easily checked. Indeed, assume (i) does not hold, i.e., $\mu\hat{x}^*A_m y + \overline{\mu}y^*A_m\hat{x} \neq 0$ for all $\mu \in \mathbb{C}$ with $|\mu| = 1$. Then the function $f(\theta) = e^{i\theta}\hat{x}^*A_m y + e^{-i\theta}y^*A_m\hat{x}$, $0 \leq \theta \leq 2\pi$, does not take the zero value and is obviously continuous; moreover, $f(\theta + \pi) = -f(\theta)$ for $0 \leq \theta \leq \pi$. Therefore, for any nonzero $\nu \in \mathbb{C}$, there exists a θ with $0 \leq \theta \leq 2\pi$ such that $f(\theta)$ is a positive multiple of ν. This property shows that (ii) or (iii) can be satisfied.

The function $z(t)$ is well defined because $\hat{x}$ and y are linearly independent and hence $\sqrt{1-t^2}\hat{x} + t\mu y \neq 0$ for all $t \in [0,1]$. Furthermore, by the choice of μ

$$z(t)^*A_m z(t) = \frac{\eta_0 + t^2(\eta_1 - \eta_0) + t\sqrt{1-t^2}(\mu\hat{x}^*Ay + \overline{\mu}y^*A\hat{x})}{\|\sqrt{1-t^2}\hat{x} + t\mu y\|^2} \neq 0$$

for all $t \in [0,1]$. Notice that the set $\{z(t)^*A_m z(t) : t \in [0,1]\}$ is a line segment in $W(A_m)$ not containing zero, and that $z(0)$ and $z(1)$ may not be the endpoints of this line segment.

Second, suppose $\eta_0 \neq \eta_1$, and the line segment joining these two points contains zero. In such a case, μA_m is not an indefinite hermitian matrix for any $\mu \in \mathbb{C}$; otherwise η_0 and η_1 are not in the same connected component of $W(A_m) \setminus \{0\}$. Thus there exists $\hat{y} \in \mathcal{S}$ such that $\eta_2 = \hat{y}^*A_m\hat{y}$, and the points η_i, $i = 0,1,2$, are not collinear. Clearly, both the line segment joining η_0 and η_2 and the line segment joining η_2 and η_1 do not contain zero. Thus both of them lie in $\mathcal{C}$. Now by arguments similar to those in the previous case, one can construct $z_1(t)$ with $t \in [0,1]$ such that $z_1(0) = \hat{x}$, $z_1(1) = \hat{\mu}\hat{y}$, and $z_1(t)^*A_m z_1(t) \neq 0$ for all $t \in [0,1]$; and then construct $z_2(t)$ with $t \in [0,1]$ such that $z_2(0) = \hat{\mu}\hat{y}$, $z_2(1) = \mu y$, and $z_2(t)^*A_m z_2(t) \neq 0$ for all $t \in [0,1]$. From z_1 and z_2, one easily obtains the required z.

Finally, suppose $\eta_0 = \eta_1$. If $W(A_m)$ is a singleton, then A_m is a scalar matrix. One may let $z(t) = (\sqrt{1-t^2}\hat{x} + ty)/\|\sqrt{1-t^2}\hat{x} + ty\|$. If $W(A_m)$ is not a singleton, there exists $\hat{y} \in \mathcal{S}$ such that $\eta_2 = \hat{y}^*A_m\hat{y} \in \mathcal{C}$ and the line segment joining η_0 and η_2 does not contain zero. One can then construct $z_1(t)$ and $z_2(t)$ as in the previous case and obtain the desired $z(t)$. □

Next we identify when $W(P(\lambda))$ is bounded.

THEOREM 2.3. *Let $P(\lambda) = A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_0$, where $A_m \neq 0$. Then $W(P(\lambda))$ is bounded if and only if $0 \notin W(A_m)$.*

Proof. If $0 \notin W(A_m)$, then $\mu = \min\{|z| : z \in W(A_m)\} > 0$. Thus there exists

$M > 0$ such that

$$|x^*A_mx\lambda^m| \geq |\mu\lambda^m| > \sum_{i=0}^{m-1} |x^*A_kx\lambda^i|,$$

whenever $x \in \mathcal{S}$ and $\lambda \in \mathbb{C}$ with $|\lambda| > M$. It then follows that $W(P(\lambda)) \subseteq \{\nu \in \mathbb{C} : |\nu| \leq M\}$.

To prove the converse, assume that $W(P(\lambda))$ is bounded but $0 \in W(A_m)$. Let $x \in \mathcal{S}$ be such that $x^*A_mx = 0$. There must be some A_j, $0 \leq j \leq m-1$ such that $x^*A_jx \neq 0$; otherwise, $W(P(\lambda)) = \mathbb{C}$, contradicting the assumption that $W(P(\lambda))$ is bounded. Since $A_m \neq 0$, we can find a sequence $\{y_p\}_{p=1}^{\infty}$, $y_p \in \mathcal{S}$, such that $\lim_{p\to\infty} y_p = x$ and $y_p^*A_my_p \neq 0$. Clearly, $|y_p^*A_jy_p| \geq \delta$ for some fixed $\delta > 0$, for all sufficiently large p. Since $W(P(\lambda))$ is bounded, the $(m-j)$th elementary symmetric function of the roots of the polynomial $y_p^*P(\lambda)y_p$, which is equal to $\pm y_p^*A_jy_p/y_p^*A_my_p$, is also bounded for all p. This clearly contradicts the construction of $\{y_p\}_{p=1}^{\infty}$. □

By Theorems 2.2 and 2.3 we have the following corollary.

COROLLARY 2.4. *If $W(P(\lambda))$ is bounded, then $W(P(\lambda))$ has at most m connected components, where m is the degree of $P(\lambda)$.*

The situation when $W(P(\lambda))$ is bounded and the number of connected components is the maximum allowed by Corollary 2.4, i.e., equal to the degree of $P(\lambda)$, is of special interest, and in the next section we encounter such situations. Here we only indicate the following fact.

PROPOSITION 2.5. *Assume that $W(P(\lambda))$ is bounded, with exactly m connected components $\Omega_1, \ldots, \Omega_m$, where m is the degree of $P(\lambda)$. Then for every $x \neq 0$ the equation $x^*P(\lambda)x = 0$ has exactly one root in each Ω_j $(1 \leq j \leq m)$.*

Proof. By Theorem 2.3, $0 \notin W(A_m)$; in particular, $W(A_m)\backslash\{0\}$ is connected. Arguing by contradiction, assume that for some nonzero $\hat{x} \in \mathbb{C}^n$, the equation $\hat{x}^*P(\lambda)\hat{x} = 0$ has roots in $\Omega_1, \ldots, \Omega_k$, where $k < m$. Proof of Theorem 2.2 then shows that for every nonzero $y \in \mathbb{C}^n$, the polynomial $y^*P(\lambda)y = 0$ has no roots in Ω_m. This contradicts the hypothesis that Ω_m is a connected component of $W(P(\lambda))$. □

Although $W(P(\lambda))$ is not always connected (and even if it is connected), it is not always convex as shown by Examples 1 and 2. One may ask (motivated by Example 1) whether the connected components of $W(P(\lambda))$ are convex if $W(P(\lambda))$ has a maximum number of disjoint connected components. However, the following example shows that they need not be.

Example 3. Let $P(\lambda) = \lambda^2 I + \lambda\varepsilon C + C$, where C is a nonscalar positive definite matrix and $\varepsilon > 0$. For sufficiently small $\varepsilon > 0$, $W(P(\lambda))$ is a union of two disjoint arcs lying on the circle $y^2 + x^2 + 2\varepsilon^{-1}x = 0$.

Note that all the results of this section (except for Proposition 2.1(a)) are valid (with basically the same proofs) for polynomials whose coefficients are bounded linear operators on a Hilbert space (in Proposition 2.1(d), S is a bounded linear operator between Hilbert spaces with zero kernel and in Theorem 2.3 the condition "$0 \notin W(A_m)$" should be replaced by "$0 \notin \overline{W(A_m)}$").

Before moving to §3, we would like to pose the following open problem.

Problem 1. Determine general conditions on $P(\lambda)$ so that $W(P(\lambda))$ is convex, connected, or its connected componenets are convex.

3. Numerical range and factorization. In this section we express a close connection between properties of the numerical range and factorization of matrix polynomials.

We start with known results. A matrix polynomial $P(\lambda)$ is called *hyperbolic* if $W(P(\lambda))$ is a bounded subset of $\mathbb{R}$. Clearly, the coefficients of a hyperbolic matrix

polynomial are hermitian matrices and (by Theorem 2.3) the leading coefficient is either positive definite or negative definite. For a hyperbolic $n \times n$ matrix polynomial $P(\lambda)$ of degree m denote by $\lambda_1(x) \leq \cdots \leq \lambda_m(x)$ the roots (all of them are real) of the equation $x^*P(\lambda)x = 0$, where $x \in \mathbb{C}^n$ is nonzero. The set

$$\Lambda_j = \{\lambda_j(x) : x \in \mathbb{C}^n \setminus \{0\}\} \tag{3.1}$$

is called the jth *spectral zone* of $P(\lambda)$; clearly, each Λ_j is a closed bounded interval (possibly degenerate) $[\alpha_j, \beta_j]$ on the real line. One can show, e.g., see [7, Satz 1] and [9, Thm. 31.5], that

$$\beta_j \leq \alpha_{j+1}, \qquad j = 1, \ldots, m-1. \tag{3.2}$$

A hyperbolic matrix polynomial $P(\lambda)$ of degree m is called *strongly hyperbolic* if $W(P(\lambda))$ has m connected components, i.e., all inequalities in (3.2) are strict.

An $n \times n$ matrix Z is called a (*right*) *matrix zero* of an $n \times n$ matrix polynomial $P(\lambda) = \sum_{j=0}^m A_j\lambda^j$ if $\sum_{j=0}^m A_jZ^j = 0$. Equivalently, $\lambda I - Z$ is a right divisor of $P(\lambda)$.

The following result has been proved in [8, Satz 1], [10, Thm. 3.2] (a weaker version of it can be found in [9, Thm. 31.2]), in the framework of polynomials whose coefficients are bounded linear operators on a Hilbert space.

THEOREM 3.1. *Let $P(\lambda) = \sum_{j=0}^m A_j\lambda^j$, $A_m \neq 0$, be a hyperbolic $n \times n$ matrix polynomial, and let Λ_j be its jth spectral zone* (3.1), *$j = 1, \ldots, m$.*

(a) *For every $j = 1, \ldots, m$, the polynomial $P(\lambda)$ has a unique right matrix zero Z_j such that $\sigma(Z_j) \subseteq \Lambda_j$; moreover, Z_j is similar to a hermitian matrix.*

(b) *$P(\lambda)$ admits a factorization*

$$P(\lambda) = A_m(\lambda I - Y_1)\cdots(\lambda I - Y_m),$$

where $Y_1, \ldots, Y_m$ are $n \times n$ matrices such that $\sigma(Y_j) \subseteq \Lambda_j$ and Y_j is similar to Z_j ($j = 1, \ldots, m$) with $Y_m = Z_m$.

(c) *If, in addition, $P(\lambda)$ is strongly hyperbolic, then for every permutation π of $\{1, \ldots, m\}$, there is a factorization*

$$P(\lambda) = A_m(\lambda I - Y_1(\pi))\cdots(\lambda I - Y_m(\pi)),$$

where $\sigma(Y_j(\pi)) \subseteq \Lambda_{\pi(j)}$, $j = 1, \ldots, m$.

In the following we obtain another factorization result (inspired by Theorem 3.1) based on the properties of numerical ranges of matrix polynomials, with the hyperbolicity hypothesis considerably relaxed.

THEOREM 3.2. *Let $P(\lambda)$ be a matrix polynomial of degree m such that $W(P(\lambda))$ is bounded and $W(P(\lambda))$ has exactly m connected components. Assume in addition that*

$$\mathrm{Ker}P(\lambda) = \mathrm{Ker}(P(\lambda))^*, \qquad \lambda \in \mathbb{C}. \tag{3.3}$$

Then $P(\lambda)$ admits a factorization

$$P(\lambda) = A_m(\lambda I - Y_1)\cdots(\lambda I - Y_m), \tag{3.4}$$

where A_m is the leading coefficient of $P(\lambda)$ and $Y_1, \ldots, Y_m$ are diagonalizable $n \times n$ matrices.

Proof. We verify first that $P(\lambda)$ has elementary divisors of first degree only, or, equivalently, that $P(\lambda)$ has no generalized eigenvectors (see [2, §1.4]), i.e., there do not exist $x, y \in \mathbb{C}^n$ with $x \neq 0$ such that

$$P(\lambda_0)x = 0, \qquad P'(\lambda_0)x + P(\lambda_0)y = 0 \tag{3.5}$$

for some $\lambda_0 \in \mathbb{C}$. Arguing by contradiction, we assume that such $x \neq 0$ and y do exist. Multiplying the second equation in (3.5) by x^* on the left and using (3.3), we have $x^*P'(\lambda_0)x = 0$. Thus, λ_0 is a multiple root of $x^*P(\lambda_0)x = 0$, a contradiction with Proposition 2.5.

Once we know that $P(\lambda)$ has elementary divisors of first degree only [2, Thm. 3.2.1] (a result originally proved in [11, Thm. 1]) guarantees the existence of the factorization (3.4). The matrices Y_j are diagonalizable because the property of having elementary divisors of first degree only is inherited by all matrix polynomials that are divisors of $P(\lambda)$ (see, e.g., [3, Thm. 5.6.1]); this property is based on the easily verified fact that the restriction of a diagonalizable matrix C to any C-invariant subspace is in turn diagonalizable. □

Several remarks concerning Theorem 3.2 are in order. First, a strongly hyperbolic matrix polynomial certainly satisfies the hypotheses of Theorem 3.2. In this sense, Theorem 3.2 is a generalization of Theorem 3.1 with weaker hypotheses and weaker conclusions. Second, condition (3.3) may be satisfied for matrix polynomials with nonhermitian coefficients. For example, the coefficients of $P(\lambda)$ may be commuting normal matrices. Third, in contrast with the scalar case, not every matrix polynomial admits a decomposition into a product of linear factors as in (3.4); for instance, the polynomial $P(\lambda) = \lambda^2 I_2 + \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ does not.

In Theorem 3.1(c), Y_j can be chosen so that all its eigenvalues are in the jth spectral zone of $W(P(\lambda))$. We could not come to an analogous conclusion in Theorem 3.2. This naturally gives rise to the following problem.

Problem 2. Under the hypotheses of Theorem 3.2, does $P(\lambda)$ admit a factorization of the form

$$P(\lambda) = A_m(\lambda I - Y_1(\pi)) \cdots (\lambda I - Y_m(\pi)),$$

where $\sigma(Y_j(\pi)) \subseteq \Lambda_{\pi(j)}$, $j = 1, \ldots, m$? Here $\Lambda_1, \ldots, \Lambda_m$ are the connected components of $W(P(\lambda))$, and π is any given permutation of $\{1, \ldots, m\}$.

Theorem 3.1 shows that the answer to Problem 2 is affirmative provided

$$W(P(\lambda)) \subseteq \mathbb{R}.$$

4. Linear polynomials with hermitian coefficients. In this section we study the numerical range of linear matrix polynomials with hermitian coefficients. For such polynomials, we provide a full characterization of the geometry of the numerical range in terms of the algebraic properties of the coefficients.

Let $P(\lambda) = \lambda A - B$, where A and B are hermitian matrices. By Proposition 2.1(e), if A and B have a common nonzero isotropic vector, then $W(P(\lambda)) = \mathbb{C}$. To avoid this trivial situation, we consider the cases in which A and B have no nonzero common isotropic vector. In all of these cases, e.g., see [12, §2] or [13, §8], A and B are simultaneously diagonalizable by a nonsingular congruence. In particular, there exists a nonsingular $S \in M_n$ such that $S^*AS = I_r \oplus -I_s \oplus 0_t$ and $B = B_1 \oplus B_2 \oplus B_3$, where $r + s + t = n$ and $B_1 \in M_r$, $B_2 \in M_s$ and $B_3 \in M_t$ are diagonal matrices. By Proposition 2.1(d), we may assume that $S = I$. Furthermore, we may assume $r \geq s$;

otherwise, replace A by $-A$ and B by $-B$. With this technical assumption on $P(\lambda)$, we are ready to prove the following result.

THEOREM 4.1. *Suppose* $P(\lambda) = A\lambda - B$, *where* $A = I_r \oplus -I_s \oplus 0_t$, $B = B_1 \oplus B_2 \oplus B_3$, $r \geq s$, $r + s + t = n$, *and* $B_1 \in M_r$, $B_2 \in M_s$ *and* $B_3 \in M_t$ *are diagonal matrices. Assume* A *and* B *have no common nonzero isotropic vector. Then we have exactly one of the following cases.*

(a) A *is positive definite, i.e.,* $A = I$, *and* $W(P(\lambda)) = W(B)$. *In particular,* $W(P(\lambda))$ *is a singleton if and only if* B *is a scalar matrix;* $W(P(\lambda))$ *is a positive (nonnegative) line segment if and only if* B *is positive (semi)definite.*

(b) A *is a singular positive semidefinite matrix;* $W(P(\lambda)) = [\alpha, \infty)$ *if* B_3 *is positive definite, and* $W(P(\lambda)) = (-\infty, \beta]$ *if* B_3 *is negative definite, where* $W(B_1) = [\alpha, \beta]$. *In this case,* B_3 *must either be positive definite or negative definite because* A *and* B *have no common nonzero isotropic vector.*

(c) A *is indefinite and* B *is positive (negative) definite and* $W(P(\lambda)) = \mathbb{R} \setminus [\alpha, \beta]$, *where* $W(B^{-1/2}AB^{-1/2}) = [1/\alpha, 1/\beta]$ $(W((-B)^{-1/2}(-A)(-B)^{-1/2}) = [1/\alpha, 1/\beta])$. *In this case,* $W(P(\lambda))$ *is the union of two disjoint unbounded intervals and* $0 \notin W(P(\lambda))$.

(d) A *is indefinite and* B *is a singular positive (negative) semidefinite matrix, and* $W(P(\lambda)) = \{\mu^{-1} : \mu \in W(B\lambda - A), \mu \neq 0\} \cup \{0\}$ $(W(P(\lambda)) = \{\mu^{-1} : \mu \in W((-B)\lambda - (-A)), \mu \neq 0\} \cup \{0\})$. *In this case,* $W(P(\lambda))$ *is the union of two disjoint unbounded intervals and* $0 \in W(P(\lambda))$.

(e) *Both* A *and* B *are indefinite and* $W(P(\lambda)) = \mathbb{R}$.

Proof. Case (a) is clear. If $s = 0$, then

$$W(P(\lambda)) = \{\frac{x^*B_1x + z^*B_3z}{x^*x} : x \in \mathbb{C}^r, z \in \mathbb{C}^t, x^*x \neq 0\},$$
$$= \begin{cases} W(B_1) + [0, \infty) & \text{if } B_3 \text{ is positive definite,} \\ W(B_1) + (-\infty, 0] & \text{if } B_3 \text{ is negative definite.} \end{cases}$$

Thus, case (b) holds.

To obtain (c) and (d), consider $W(B\lambda - A)$ and use Proposition 2.1(c).

For (e), notice that if we take $u = e_1 + e_{r+1}$, then $u^*Bu \neq 0$. Now consider the sequence $u_k = e_1 + ke_{r+1}/(k+1)$ and the sequence $v_k = ke_1/(k+1) + e_{r+1}$, $k = 1, 2, \ldots$. One sees that $\{u_k^*Bu_k/u_k^*Au_k\}$ and $\{v_k^*Bv_k/v_k^*Av_k\}$ go to $\pm\infty$ in opposite directions. Thus $W(P(\lambda))$ is unbounded above and below. Now apply similar arguments to $W(B\lambda - A)$ and use Proposition 2.1(c), to see that $W(P(\lambda))$ contains all positive and negative numbers. Clearly, $0 \in W(P(\lambda))$. Thus $W(P(\lambda)) = \mathbb{R}$, as asserted. □

5. Polynomials with a degenerate numerical range. From Theorem 4.1, one sees that if the numerical range of a linear matrix polynomial $P(\lambda)$ with hermitian coefficients is given, then one has quite a lot of information about the coefficient matrices. It is interesting to study the more general case, but the analysis will undoubtedly be more involved. In this section, we study the cases in which the numerical range is a singleton or lies on the real axis.

THEOREM 5.1. *Let* $P(\lambda) = A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_0$, *where* $A_m \neq 0$. *Then* $W(P(\lambda)) = \{\alpha\}$ *with* $\alpha \in \mathbb{C}$ *if and only if* $0 \notin W(A_m)$ *and* $P(\lambda) = A_m(\lambda I - \alpha I)^m$.

Proof. The sufficiency part is clear. To prove the necessity part, observe first that $0 \notin W(A_m)$ by Theorem 2.3. Furthermore, if $W(P(\lambda)) = \{\alpha\}$ then $x^*P(\lambda)x$ must be of the form $\mu(\lambda - \alpha)^m$ for all $x \in \mathcal{S}$. In particular, for every $j = 0, \ldots, m-1$, $x^*A_jx/x^*A_mx = (-\alpha)^{m-j}$ and hence $x^*(A_j - (-\alpha)^{m-j}A_m)x = 0$ for all $x \in \mathcal{S}$. It follows that $A_j = (-\alpha)^{m-j}A_m$ for all j and the result follows. □

Now assume that $W(P(\lambda)) \subseteq \mathbb{R}$, so that all the coefficients of $P(\lambda)$ are hermitian matrices. It is clear (even in the scalar case) that this condition is not sufficient to ensure that $W(P(\lambda)) \subseteq \mathbb{R}$. In general, a convenient criterion for $W(P(\lambda)) \subseteq \mathbb{R}$ is not available. However, under additional hypotheses such a criterion can be provided. For example, the following result holds in the quadratic case.

THEOREM 5.2. *Let $P(\lambda) = A_2\lambda^2 + A_1\lambda + A_0$, where A_j is $n \times n$ hermitian for $j = 0, 1, 2$, and $A_2 \neq 0$. Then the following are equivalent:*

(a) $W(P(\lambda)) \subseteq \mathbb{R}$.

(b) $4(x^*A_0x)(x^*A_2x) \leq (x^*A_1x)^2$ *for all* $x \in \mathcal{S}$.

(c) *For every* $x \in \mathcal{S}$ *there exists* $\lambda = \lambda(x) \in \mathbb{R}$ *such that* $x^*P(\lambda)x \leq 0$.

Moreover, if A_2 is positive definite, then conditions (a)–(c) *are equivalent to the following*:

(d) *There exists* $\lambda \in \mathbb{R}$ *such that* $-P(\lambda)$ *is positive semidefinite.*

Proof. The equivalence of (a)–(c) follows easily from the formula for solving the quadratic equation $\lambda^2(x^*A_2x) + \lambda(x^*A_1x) + (x^*A_0x) = 0$.

Clearly, (d) implies (c). We claim that the converse is also valid if A_2 is positive definite. Consider the following condition.

(e) For every $x \in \mathcal{S}$ there exists $\lambda = \lambda(x) \in \mathbb{R}$ such that $x^*P(\lambda)x < 0$.

It is known (e.g., see [2, Thm. 13.1]) that (e) implies (d). Now if (c) holds, then (e) holds for the matrix polynomial $A_2\lambda^2 + A_1\lambda + A_0 - \varepsilon I$, for every $\varepsilon > 0$. Therefore, for every $\varepsilon > 0$, there exists $\lambda(\varepsilon) \in \mathbb{R}$ such that $-(A_2\lambda^2 + A_1\lambda + A_0 - \varepsilon I)$ is positive semidefinite. Since A_2 is positive definite, the set $\{\lambda(\varepsilon) : 0 < \varepsilon \leq 1\}$ is bounded. Taking a convergent subsequence of the sequence $\{\lambda(1/p)\}_{p=1}^{\infty}$ and passing to the limit when $p \to \infty$, we obtain condition (d). □

Note added in proof. Professor A.S. Markus, Ben Gurion University, Beer-Sheva, Israel, has informed us that the answer to Problem 2 is affirmative for $m = 2$ and $m = 3$. For $m = 2$, this follows from Theorem 1 in [14]. In case $m = 3$, the polynomial $P(\lambda)$ can be first factored in the from $P(\lambda) = (\lambda I - Y_1(\pi))Q(\lambda)$, where $\sigma(Y_1(\pi)) \subseteq \Lambda_{\pi(1)}$ and $Q(\lambda)$ is a matrix polynomial of second degree (the existence of such factorization follows from Theorem 2 in [15]). Using this factorization of $P(\lambda)$, the proof of the above-mentioned Theorem 1 yields the desired result. Problem 2 remains open for $m \geq 4$.

REFERENCES

[1] R.J. DUFFIN, *A minimax theory for overdamped networks*, J. Rat. Mech. Anal., 4 (1955), pp. 221–233.

[2] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrix Polynomials*, Academic Press, New York, 1982.

[3] ———, *Invariant Subspaces of Matrices*, J. Wiley & Sons, New York, 1986.

[4] R.A. HORN AND C.R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, London, 1991.

[5] M.G. KREIN AND H. LANGER, *On some mathematical principles in the linear theory of damped oscillations of continua*, I, II, Integral Equations Operator Theory, 1 (1978), pp. 364–399 and pp. 539–566. (Translation from Russian original, 1964.)

[6] P. LANCASTER, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, 1966.

[7] H. LANGER, *Über eine Klasse polynomialer Scharen selbstadjungierter Operatoren im Hilbertraum*, J. Funct. Anal., 12 (1973), pp. 13–29.

[8] H. LANGER, *Über eine Klasse polynomialer Scharen selbstadjungierter Operatoren im Hilbertraum* II, J. Funct. Anal., 16 (1974), pp. 221–234.

[9] A. S. MARKUS, *Introduction to Spectral Theory of Polynomial Operator Pencils*, Stiinca, Kishinev, 1986. (In Russian.) AMS Transl. Math. Monographs, 71, 1988. (In English.)

[10] A. S. MARKUS AND V. I. MATSAEV, *On spectral factorization of holomorphic operator functions*, Matem. Issled., 47 (1978), pp. 71–100. (In Russian.)

[11] A. S. MARKUS AND V. I. MEREUTSA, *On some properties of λ-matrices*, Matem. Issled., 3 (1975), pp. 207–213. (In Russian.)

[12] R. C. THOMPSON, *The characteristic polynomial of a principal subpencil of a Hermitian matrix pencil*, Linear Algebra Appl., 14 (1976), pp. 135–177.

[13] ———, *Pencils of complex and real symmetric and skew matrices*, Linear Algebra Appl., 147 (1991), pp. 323–371.

[14] V. I. KABAK AND A. S. MARKUS, *On factorization of polynomial bundles into linear factors*, Uspehi Mat. Nauk., 30 (1975), pp. 245–246. (In Russian.)

[15] A.S. MARKUS AND V. I. MATSAEV, *On spectral theory of holomorphic operator functions in Hilbert space,* Funct. Analysis, 9 (1975), pp. 76–77. (In Russian.)

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1266–1276, October 1994

014

A STABLE AND EFFICIENT ALGORITHM FOR THE RANK-ONE MODIFICATION OF THE SYMMETRIC EIGENPROBLEM*

MING GU† AND STANLEY C. EISENSTAT‡

Abstract. An algorithm is presented for computing the eigendecomposition of a symmetric rank-one modification of a symmetric matrix whose eigendecomposition is known. Previous algorithms for this problem suffer a potential loss of orthogonality among the computed eigenvectors, unless extended precision arithmetic is used. This algorithm is based on a novel, stable method for computing the eigenvectors. It does not require extended precision and is as efficient as previous approaches.

Key words. symmetric eigenproblem, rank-one modification

AMS subject classification. 65F15

1. Introduction.

Given a real scalar ρ, a real n-vector u, and the eigendecomposition of a real $n \times n$ symmetric matrix B, the rank-one modification of the symmetric eigenproblem is to find the eigendecomposition of the matrix $B + \rho uu^T$.

This is an important problem in numerical linear algebra. Applications include divide-and-conquer algorithms for the symmetric tridiagonal eigenproblem [8], [9], [14], [16], [18], the bidiagonal singular value decomposition (SVD) [2]–[4], [14], [17], [20], [21], and the unitary and orthogonal eigenproblems [1], [4], [12]; updating the SVD [6], [14], [15]; and various stationary value problems [10].

The problem can easily be reduced to the following special case (see [7]). Given a diagonal matrix $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ with $d_1 < d_2 < \cdots < d_n$, and a real vector $z = (z_1, z_2, \ldots, z_n)^T$ with $z_j > 0$, find the eigendecomposition

$$D + zz^T = Q\Lambda Q^T$$

of $A \equiv D + zz^T$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_1 < \lambda_2 < \cdots < \lambda_n$, and Q is orthogonal. The diagonal elements of Λ are the eigenvalues of A and the columns of Q are the corresponding eigenvectors. From now on we focus on this reduced problem, yet still refer to it as the rank-one modification problem.

Since error is associated with computation, a *numerical eigendecomposition* of $D + zz^T$ is usually defined as a decomposition of the form

$$D + zz^T = \tilde{Q}\tilde{\Lambda}\tilde{Q}^T + O(\epsilon(\|D\|_2 + \|z\|_2^2)), \tag{1}$$

where ϵ is the machine precision, $\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n)$ with $\tilde{\lambda}_1 < \tilde{\lambda}_2 < \cdots < \tilde{\lambda}_n$, and $\tilde{Q}$ is *numerically orthogonal.* An algorithm that produces such a decomposition is said to be *stable.*

While the eigenvalues of A are always well conditioned with respect to a symmetric perturbation, the eigenvectors can be extremely sensitive to such perturbations [11], [26], [28], [29]. That is, $\tilde{\Lambda}$ is guaranteed to be close to Λ, but $\tilde{Q}$ can be very different from Q. Thus one is usually content with a stable algorithm.

* Received by the editors October 19, 1992; accepted for publication July 2, 1993. This research was supported in part by U.S. Army Research Office contract DAAL03-91-G-0032.

† Department of Mathematics and Lawrence Berkeley Laboratory, University of California, Berkeley, California 94720 (minggu@math.berkeley.edu).

‡ Department of Computer Science, Yale University, P. O. Box 208285, New Haven, Connecticut 06520-8285 (eisenstat-stan@cs.yale.edu).

The problem can be further simplified in light of (1). Given any rank-one modification matrix $D + zz^T$, we can use the deflation procedure in [9] to reduce the eigenproblem to one that satisfies

$$d_{j+1} - d_j \geq \tau(\|D\|_2 + \|z\|_2^2) \quad \text{and} \quad z_j \geq \tau\sqrt{\|D\|_2 + \|z\|_2^2},$$

where τ is a small multiple of ϵ to be specified later.

The basic tool for the rank-one modification problem is an algorithm developed by Bunch, Nielsen, and Sorensen [7] and inspired by earlier work of Golub [10]. Dongarra and Sorensen [9] propose a more liberal deflation process to make the algorithm more efficient and more stable. We refer to the algorithm in [9] as *Algorithm* I.

While Algorithm I always computes the eigenvalues to high absolute accuracy, in the presence of close eigenvalues it can have difficulties in computing numerically orthogonal eigenvectors [7]–[9]. This instability affects all algorithms using rank-one modification techniques.

To overcome this instability, Kahan [22] proposes using extended precision arithmetic to compute some key quantities more accurately. Independently, Sorensen and Tang [27] develop a new version of Algorithm I that uses simulated extended precision and they show that it is stable. The problem with extended precision is that it results in machine-dependent software [5], [27].

In this paper we present a new algorithm for solving the rank-one modification problem. Since Algorithm I works well for finding the eigenvalues, the new algorithm uses a similar approach. But it uses a completely different approach to finding the eigenvectors; one that is stable. The amount of work for the stable approach is roughly the same as for Algorithm I, yet the new algorithm does not require the use or simulation of extended precision arithmetic. We refer to this new algorithm as *Algorithm* II.

Section 2 introduces Algorithm I and points out how it can fail. Section 3 introduces Algorithm II. Section 4 proves the numerical stability of Algorithm II. Section 5 reviews previous results on the stability of Algorithm I and shows why these results require more accuracy than necessary. Section 6 presents numerical results and §7 points out extensions.

Throughout the paper we assume that the elements of D and z are given in floating-point representation. We take the usual model of arithmetic[1]

$$fl(x \circ y) = (x \circ y)(1 + \xi),$$

where x and y are floating-point numbers; $\circ$ is one of $+, -, \times$, and $\div$; $fl(x \circ y)$ is the floating-point result of the operation $\circ$; and $|\xi| \leq \epsilon$. We also require that

$$fl(\sqrt{x}) = \sqrt{x}(1 + \xi)$$

for any positive floating-point number x. For simplicity, we ignore the possibility of overflow and underflow.

2. How Algorithm I can fail. The following lemma characterizes the eigenvalues and eigenvectors of $D + zz^T$.

[1] This model excludes machines like the CRAY and CDC Cyber that do not have a guard digit. Algorithm II can easily be modified for such machines.

LEMMA 2.1 (Bunch, Nielsen, and Sorensen [7]). *Assume that* $d_1 < d_2 < \cdots < d_n$ *and that* $z_j > 0$. *Then the eigenvalues* $\{\lambda_i\}_{i=1}^n$ *of* $D + zz^T$ *satisfy the* interlacing property

$$d_1 < \lambda_1 < d_2 < \lambda_2 < \cdots < d_n < \lambda_n$$

and are the roots of the secular equation

$$f(\lambda) \equiv 1 + \sum_{j=1}^{n} \frac{z_j^2}{d_j - \lambda} = 0.$$

For each eigenvalue λ_i, *the corresponding eigenvector is given by*

$$q_i = \left(\frac{z_1}{d_1 - \lambda_i}, \ldots, \frac{z_n}{d_n - \lambda_i}\right)^T \Bigg/ \sqrt{\sum_{j=1}^{n} \frac{z_j^2}{(d_j - \lambda_i)^2}}. \tag{2}$$

Algorithm I uses a rational interpolation strategy to solve for $\{\lambda_i\}_{i=1}^n$ (see [7]). For each eigenvalue λ_i, it finds a numerical approximation $\tilde{\lambda}_i$ and approximates q_i by

$$\tilde{q}_i^I = \left(\frac{z_1}{d_1 - \tilde{\lambda}_i}, \ldots, \frac{z_n}{d_n - \tilde{\lambda}_i}\right)^T \Bigg/ \sqrt{\sum_{j=1}^{n} \frac{z_j^2}{(d_j - \tilde{\lambda}_i)^2}}.$$

In other words, the *exact* λ_i is replaced by the *approximation* $\tilde{\lambda}_i$ in (2).

In pathological cases, even though $\tilde{\lambda}_i$ is very close to λ_i, the approximate ratio $z_j/(d_j - \tilde{\lambda}_i)$ is very different from the exact ratio $z_j/(d_j - \lambda_i)$, resulting in a computed eigenvector very different from the true eigenvector. More importantly, when all the eigenvectors are computed, the resulting eigenvector matrix is far from orthogonal.

3. Algorithm II.

3.1. Computing the eigenvectors. For each eigenvalue λ_i, the corresponding eigenvector is given by

$$q_i = \left(\frac{z_1}{d_1 - \lambda_i}, \ldots, \frac{z_n}{d_n - \lambda_i}\right)^T \Bigg/ \sqrt{\sum_{j=1}^{n} \left(\frac{z_j}{d_j - \lambda_i}\right)^2}$$

(see Lemma 2.1). Observe that if λ_i was given *exactly*, then we could compute each difference, each ratio, each product, and each sum in this formula to high relative accuracy, and thus compute q_i to componentwise high relative accuracy.

In practice we can only hope to compute an approximation $\tilde{\lambda}_i$ to λ_i. But suppose that we could find a $\tilde{z}$ such that $\{\tilde{\lambda}_i\}_{i=1}^n$ are the *exact* eigenvalues of the *new* rank-one modification matrix $\tilde{A} = D + \tilde{z}\tilde{z}^T$. (We rederive Löwner's [24] solution to this *inverse eigenvalue problem* below.) Since

$$\begin{aligned} A &= D + zz^T \\ &= D + \tilde{z}\tilde{z}^T + zz^T - \tilde{z}\tilde{z}^T \\ &= \tilde{A} + (z - \tilde{z})z^T + z(z - \tilde{z})^T - (z - \tilde{z})(z - \tilde{z})^T, \end{aligned}$$

$\tilde{A}$ will be close to A as long as $\tilde{z}$ is close to z. Moreover, the formula

$$\tilde{q}_i^{II} = \left(\frac{\tilde{z}_1}{d_1 - \tilde{\lambda}_i}, \ldots, \frac{\tilde{z}_n}{d_n - \tilde{\lambda}_i}\right)^T \Bigg/ \sqrt{\sum_{j=1}^{n} \frac{\tilde{z}_j^2}{(d_j - \tilde{\lambda}_i)^2}} \tag{3}$$

gives the *exact* eigenvector corresponding to the eigenvalue $\tilde{\lambda}_i$ of $\tilde{A}$. As we observed before, $\tilde{q}_i^{II}$ can be computed to componentwise high relative accuracy. Thus, when all the eigenvectors of $\tilde{A}$ are computed, the resulting eigenvector matrix will be numerically orthogonal.

We now show why such a $\tilde{z}$ exists (cf. [5], [24]). By definition,

$$\det(\tilde{A} - \lambda I) = \prod_{j=1}^{n} (\tilde{\lambda}_j - \lambda).$$

On the other hand,

$$\det(\tilde{A} - \lambda I) = \det(D + \tilde{z}\tilde{z}^T - \lambda I) = \left(1 + \sum_{j=1}^{n} \frac{\tilde{z}_j^2}{d_j - \lambda}\right) \prod_{j=1}^{n} (d_j - \lambda).$$

Combining these relations,

$$\prod_{j=1}^{n} (\tilde{\lambda}_j - \lambda) = \left(1 + \sum_{j=1}^{n} \frac{\tilde{z}_j^2}{d_j - \lambda}\right) \prod_{j=1}^{n} (d_j - \lambda).$$

Setting $\lambda = d_i$, we get

$$\tilde{z}_i^2 = \frac{\prod_j (\tilde{\lambda}_j - d_i)}{\prod_{j \neq i} (d_j - d_i)}. \tag{4}$$

If the computed eigenvalues $\{\tilde{\lambda}_i\}_{i=1}^n$ satisfy the interlacing property[2]

$$d_1 < \tilde{\lambda}_1 < d_2 < \tilde{\lambda}_2 < \ldots < d_n < \tilde{\lambda}_n,$$

then the expression on the right-hand side of (4) is positive and

$$\tilde{z}_i = \sqrt{\prod_{j=1}^{i-1} \frac{(\tilde{\lambda}_j - d_i)}{(d_j - d_i)} \cdot \prod_{j=i}^{n-1} \frac{(\tilde{\lambda}_j - d_i)}{(d_{j+1} - d_i)} \cdot (\tilde{\lambda}_n - d_i)}. \tag{5}$$

Working backward, if $\tilde{z}$ is given by (5), then the eigenvalues of $D + \tilde{z}\tilde{z}^T$ are $\{\tilde{\lambda}_i\}_{i=1}^n$.

Each difference, each ratio, and each product in (5) can be computed to high relative accuracy. As a result, $\tilde{z}$ can be computed to componentwise high relative accuracy. Substituting the computed $\tilde{z}$ into (3), $\tilde{q}_i^{II}$ can also be computed to componentwise high relative accuracy. Consequently, when all the eigenvectors are computed, the resulting eigenvector matrix $\tilde{Q} = (\tilde{q}_1^{II}, \ldots, \tilde{q}_n^{II})$ will be numerically orthogonal.

To show that $\tilde{Q}\tilde{\Lambda}\tilde{Q}^T$ is a numerical eigendecomposition of A, we must show that $\tilde{z}$ is close to z. We do so in §4.

[2] Since the exact eigenvalues satisfy the same interlacing property (see Lemma 2.1), this is only an accuracy requirement on the computed eigenvalues and is not an additional restriction on A.

3.2. Computing the eigenvalues. To guarantee that $\tilde{z}$ is close to z, we must ensure that the approximations $\{\tilde{\lambda}_i\}_{i=1}^n$ to the eigenvalues are sufficiently accurate. The key is the stopping criterion for the root-finder, which requires a slight reformulation of the secular equation (cf. [7]).

Consider the eigenvalue $\lambda_i \in (d_i, d_{i+1})$, where $1 \le i \le n-1$; we consider the case $i = n$ later. λ_i is a root of the secular equation

$$f(\lambda) \equiv 1 + \sum_{j=1}^{n} \frac{z_j^2}{d_j - \lambda} = 0.$$

We first assume that $\lambda_i \in (d_i, \frac{d_i+d_{i+1}}{2})$. This can easily be checked by computing $f(\frac{d_i+d_{i+1}}{2})$. If $f(\frac{d_i+d_{i+1}}{2}) > 0$, then $\lambda_i \in (d_i, \frac{d_i+d_{i+1}}{2})$, otherwise $\lambda_i \in [\frac{d_i+d_{i+1}}{2}, d_{i+1})$. Let $\delta_j = d_j - d_i$ and let

$$\psi(\mu) \equiv \sum_{j=1}^{i} \frac{z_j^2}{\delta_j - \mu} \quad \text{and} \quad \phi(\mu) \equiv \sum_{j=i+1}^{n} \frac{z_j^2}{\delta_j - \mu}.$$

Since

$$f(\mu + d_i) = 1 + \psi(\mu) + \phi(\mu) \equiv g(\mu),$$

we seek the root $\mu_i = \lambda_i - d_i \in (0, \delta_{i+1}/2)$ of $g(\mu) = 0$.

An important property of $g(\mu)$ is that each difference $\delta_j - \mu$ can be evaluated to high relative accuracy for any $\mu \in (0, \delta_{i+1}/2)$. Indeed, since $\delta_i = 0$, we have $fl(\delta_i - \mu) = -fl(\mu)$; since $fl(\delta_{i+1}) = fl(d_{i+1} - d_i)$ and $0 < \mu < (d_{i+1} - d_i)/2$, we can compute $fl(\delta_{i+1} - \mu)$ as $fl(fl(d_{i+1} - d_i) - fl(\mu))$; and similarly, we can compute $\delta_j - \mu$ to high relative accuracy for any $j \neq i, i+1$.

Because of this property, each ratio $z_j^2/(\delta_j - \mu)$ in $g(\mu)$ can be evaluated to high relative accuracy for any $\mu \in (0, \delta_{i+1}/2)$. And, since both $\psi(\mu)$ and $\phi(\mu)$ are sums of terms with the same sign, we can bound the error in computing $g(\mu)$ by

$$\eta n(1 + |\psi(\mu)| + |\phi(\mu)|),$$

where η is a small multiple of ϵ that is independent of n and μ.

We now assume that $\lambda_i \in [\frac{d_i+d_{i+1}}{2}, d_{i+1})$. Let $\delta_j = d_j - d_{i+1}$ and let

$$\psi(\mu) \equiv \sum_{j=1}^{i} \frac{z_j^2}{\delta_j - \mu} \quad \text{and} \quad \phi(\mu) \equiv \sum_{j=i+1}^{n} \frac{z_j^2}{\delta_j - \mu}.$$

We seek the root $\mu_i = \lambda_i - d_{i+1} \in [\delta_i/2, 0)$ of the equation

$$g(\mu) \equiv f(\mu + d_{i+1}) = 1 + \psi(\mu) + \phi(\mu) = 0.$$

For any $\mu \in [\delta_i/2, 0)$, each difference $\delta_j - \mu$ can again be computed to high relative accuracy, as can each ratio $z_j^2/(\delta_j - \mu)$; and we can bound the error in computing $g(\mu)$ as before.

Finally we consider the case $i = n$. Let $\delta_j = d_j - d_n$ and let

$$\psi(\mu) \equiv \sum_{j=1}^{n} \frac{z_j^2}{\delta_j - \mu} \quad \text{and} \quad \phi(\mu) \equiv 0.$$

We seek the root $\mu_n = \lambda_n - d_n \in (0, \|z\|_2^2)$ of the equation

$$g(\mu) \equiv f(\mu + d_n) = 1 + \psi(\mu) + \phi(\mu) = 0.$$

Again, for any $\mu \in (0, \|z\|_2^2)$, each ratio $z_j^2/(\delta_j - \mu)$ can be computed to high relative accuracy, and we can bound the error in computing $g(\mu)$ as before.

In practice the root-finder cannot make any progress at a point μ where it is impossible to determine the sign of $g(\mu)$ numerically. Thus we propose the stopping criterion

$$|g(\mu)| \le \eta n \left(1 + |\psi(\mu)| + |\phi(\mu)|\right), \tag{6}$$

where, as before, $\eta n(1 + |\psi(\mu)| + |\phi(\mu)|)$ is an upper bound on the roundoff error in computing $g(\mu)$. Note that for each i, there is at least one floating-point number that satisfies this stopping criterion numerically, namely, $fl(\mu_i)$.

We have not specified the scheme used to find the root of $g(\mu)$. We used the rational interpolation strategy in [7] for the numerical experiments, but bisection or the improved rational interpolation strategies in [13] and [23] would also work. What is most important is the stopping criterion and the fact that, with the reformulation of the secular equation given above, we can find a μ that satisfies it.

3.3. Efficiency. The only additional work in Algorithm II is the evaluation of $\tilde{z}$ using (5). This is roughly equivalent to 1–2 extra evaluations of the secular equation; however, Li [23] reports a comparable savings from using our stopping criterion (6).

4. Numerical stability of Algorithm II. In this section we show that Algorithm II computes the eigenvalues to high absolute accuracy and that $\tilde{z}$ is indeed close to z.

Since $f(\lambda_i) = 0$, we have

$$1 = -\sum_{j=1}^{n} \frac{z_j^2}{d_j - \lambda_i} \le \sum_{j=1}^{n} \frac{z_j^2}{|d_j - \lambda_i|},$$

and the stopping criterion (6) implies that the computed eigenvalue $\tilde{\lambda}_i$ satisfies

$$|f(\tilde{\lambda}_i)| \le \eta n \left(\sum_{j=1}^{n} \frac{z_j^2}{|d_j - \lambda_i|} + \sum_{j=1}^{n} \frac{z_j^2}{|d_j - \tilde{\lambda}_i|} \right).$$

Since

$$f(\tilde{\lambda}_i) = f(\tilde{\lambda}_i) - f(\lambda_i) = (\tilde{\lambda}_i - \lambda_i) \sum_{j=1}^{n} \frac{z_j^2}{(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)},$$

it follows that

$$|\tilde{\lambda}_i - \lambda_i| \sum_{j=1}^{n} \frac{z_j^2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|} \le \eta n \left(\sum_{j=1}^{n} \frac{z_j^2}{|d_j - \tilde{\lambda}_i|} + \sum_{j=1}^{n} \frac{z_j^2}{|d_j - \lambda_i|} \right). \tag{7}$$

Thus

$$\begin{aligned} &|\tilde{\lambda}_i - \lambda_i| \sum_{j=1}^{n} \frac{z_j^2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|} \\ &\quad \le \eta n \max_{1 \le k \le n} \left(|d_k - \tilde{\lambda}_i| + |d_k - \lambda_i| \right) \sum_{j=1}^{n} \frac{z_j^2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|} \end{aligned}$$

or

$$|\tilde{\lambda}_i - \lambda_i| \le \eta n \max_{1\le k\le n} \left(|d_k - \tilde{\lambda}_i| + |d_k - \lambda_i|\right) \le \frac{2\eta n}{1-\eta n} \max_{1\le k\le n} |d_k - \lambda_i|,$$

i.e., all the eigenvalues are computed to high absolute accuracy.

To show that $\tilde{z}$ is close to z, we note that for any j,

$$\frac{1}{|d_j - \tilde{\lambda}_i|} + \frac{1}{|d_j - \lambda_i|} \le \frac{2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|^{\frac{1}{2}}} + \frac{|\tilde{\lambda}_i - \lambda_i|}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|}.$$

Substituting this into (7), we get

$$\begin{aligned} |\tilde{\lambda}_i - \lambda_i| \sum_{j=1}^{n} \frac{z_j^2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|} &\le \frac{2\eta n}{1-\eta n} \sum_{j=1}^{n} \frac{z_j^2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|^{\frac{1}{2}}} \\ &\le \frac{2\eta n}{1-\eta n} \|z\|_2 \sqrt{\sum_{j=1}^{n} \frac{z_j^2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|}} \end{aligned}$$

or

$$\begin{aligned} |\tilde{\lambda}_i - \lambda_i| &\le \frac{2\eta n}{1-\eta n} \|z\|_2 \Bigg/ \sqrt{\sum_{j=1}^{n} \frac{z_j^2}{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|}} \\ &\le \frac{2\eta n \|z\|_2}{(1-\eta n) z_j} \sqrt{|(d_j - \tilde{\lambda}_i)(d_j - \lambda_i)|} \\ &\le \frac{2\eta n \|z\|_2}{(1-\eta n) z_j} \left(|d_j - \lambda_i| + \frac{1}{2}|\tilde{\lambda}_i - \lambda_i|\right). \end{aligned}$$

Letting $\beta_j = 2\eta n \|z\|_2 / ((1-\eta n) z_j)$, this implies that

$$|\tilde{\lambda}_i - \lambda_i| \le \frac{\beta_j}{1 - \frac{1}{2}\beta_j} |d_j - \lambda_i| \tag{8}$$

for every $1 \le j \le n$, provided that $\beta_j < 2$.

Let $\tilde{\lambda}_i - \lambda_i = \alpha_{ij}(d_j - \lambda_i)/z_j$ for all i and j. Suppose that we pick $\tau = 2\eta n^2$ in the deflation procedure of §1. Then $z_j \ge 2\eta n^2 \|z\|_2$. Assume further that $\eta n < 1/100$. Then $\beta_j \le 2/3$, and (8) implies that $|\alpha_{ij}| \le \alpha \equiv 4\eta n \|z\|_2$ for all i and j. Thus

$$\tilde{z}_i = \sqrt{\frac{\prod_j (\tilde{\lambda}_j - d_i)}{\prod_{j\ne i}(d_j - d_i)}} = \sqrt{\frac{\prod_j (\lambda_j - d_i)(1 + \alpha_{ji}/z_i)}{\prod_{j\ne i}(d_j - d_i)}} = z_i \sqrt{\prod_{j=1}^{n} \left(1 + \frac{\alpha_{ji}}{z_i}\right)}$$

and

$$\begin{aligned} |\tilde{z}_i - z_i| = z_i \left| \sqrt{\prod_{j=1}^{n} \left(1 + \frac{\alpha_{ji}}{z_i}\right)} - 1 \right| &\le z_i \left(\left(1 + \frac{\alpha}{z_i}\right)^{n/2} - 1 \right) \\ &\le z_i \left(\exp\left(\frac{\alpha n}{2 z_i}\right) - 1 \right) \le (e-1)\alpha n/2 \\ &\le 4\eta n^2 \|z\|_2, \end{aligned} \tag{9}$$

where we have used the fact that $\alpha n/(2z_i) \leq 1$ and that $(e^x - 1)/x \leq e - 1$ for $0 < x \leq 1$.

One factor of n in τ and (9) comes from the stopping criterion (6). This is quite conservative and could be reduced to $\log_2 n$ by using a binary tree structure for summing up the terms in $\psi(\mu)$ and $\phi(\mu)$. The other factor of n comes from the upper bound for $\prod_j (1 + \alpha_{ji}/z_i)$. This also seems quite conservative. Thus we might expect the factor of n^2 in τ and (9) to be more like $O(n)$ in practice.

5. Another view of numerical stability. In this section we review previous results on the numerical stability of the eigenvector computation and show why they impose unnecessary requirements on the accuracy to which the eigenvalues are computed.

The following lemma bounds the lack of numerical orthogonality in the eigenvectors computed by Algorithm I.

LEMMA 5.1 (Dongarra and Sorensen [9]). *Let λ_k and λ_ℓ be distinct eigenvalues of $D + zz^T$. Let $\hat{q}_k = \hat{u}_k/\|\hat{u}_k\|_2$ and $\hat{q}_\ell = \hat{u}_\ell/\|\hat{u}_\ell\|_2$, where*

$$\hat{u}_k = \left(\frac{z_1}{\tilde{\delta}_{k1}}, \frac{z_2}{\tilde{\delta}_{k2}}, \ldots, \frac{z_n}{\tilde{\delta}_{kn}}\right)^T$$

and

$$\hat{u}_\ell = \left(\frac{z_1}{\tilde{\delta}_{\ell 1}}, \frac{z_2}{\tilde{\delta}_{\ell 2}}, \ldots, \frac{z_n}{\tilde{\delta}_{\ell n}}\right)^T$$

are the computed eigenvectors corresponding to the exact eigenvectors q_k and q_ℓ. If

$$\tilde{\delta}_{ij} = (d_j - \lambda_i)(1 + \eta_{ij}),$$

where $|\eta_{ij}| \leq \gamma \ll 1$ for all i and j, then

$$\hat{q}_k^T \hat{q}_\ell \leq \gamma(2 + \gamma)\left(\frac{1+\gamma}{1-\gamma}\right)^2.$$

Thus numerical orthogonality can be assured for Algorithm I whenever it is possible to compute all of the differences $d_j - \lambda_i$ to high relative accuracy. Sorensen and Tang [27] show that in pathological cases one encounters enormous difficulties meeting this condition, and thus they advocate the use of extended precision arithmetic.

But what does this condition imply about Algorithm II? Recall that

$$z_i = \sqrt{\prod_{j=1}^{i-1} \frac{(\lambda_j - d_i)}{(d_j - d_i)} \cdot \prod_{j=i}^{n-1} \frac{(\lambda_j - d_i)}{(d_{j+1} - d_i)} \cdot (\lambda_n - d_i)}$$

and

$$\tilde{z}_i = \sqrt{\prod_{j=1}^{i-1} \frac{(\tilde{\lambda}_j - d_i)}{(d_j - d_i)} \cdot \prod_{j=i}^{n-1} \frac{(\tilde{\lambda}_j - d_i)}{(d_{j+1} - d_i)} \cdot (\tilde{\lambda}_n - d_i)}.$$

Thus if we compute all of the differences $\lambda_j - d_i$ to high relative accuracy, then $\tilde{z}_i$ will be close to z_i to high *relative* accuracy. In contrast we have shown only that the stopping criterion guarantees that $\tilde{z}_i$ is close to z_i to high *absolute* accuracy, but this is enough for Algorithm II to be stable.

6. Numerical results. In this section we present some numerical results for Algorithms I and II. The tests were run on a SPARCstation 1 in double precision arithmetic. The machine precision is $\epsilon \approx 1.1 \times 10^{-16}$. τ was taken small enough that no deflation occurs.

We define the scaled orthogonality and residual measures

$$\mathcal{O} = \max_{1 \le i \le n} \frac{\|\tilde{Q}^T \tilde{q}_i - e_i\|_2}{n\epsilon} \quad \text{and} \quad \mathcal{R} = \max_{1 \le i \le n} \frac{\|A\tilde{q}_i - \tilde{\lambda}_i \tilde{q}_i\|_2}{n\epsilon \|A\|_2}.$$

$\mathcal{O}_I$ and $\mathcal{O}_{II}$ are the orthogonality measures for Algorithms I and II, respectively; $\mathcal{R}_I$ and $\mathcal{R}_{II}$ are the residual measures for Algorithms I and II, respectively. We also define

$$Z = \max_{1 \le i \le n} \frac{|\tilde{z}_i - z_i|}{n\epsilon \|z\|_2},$$

which measures the scaled absolute error in replacing z by $\tilde{z}$.

We use the following three sets of test problems.

TEST 1 [27]. These problems arise in applying Cuppen's divide-and-conquer algorithm [8], [9] to the matrix

$$T = \begin{pmatrix} W_{21} & \beta & & & & \\ \beta & W_{21} & \beta & & & \\ & \beta & W_{21} & & & \\ & & & \ddots & & \\ & & & & \beta & \\ & & & & \beta & W_{21} \end{pmatrix},$$

where W_{21} is the symmetric tridiagonal matrix of order 21 with diagonal elements $10, 9, \ldots, 1, 0, 1, 2, \ldots, 10$ and off-diagonal elements all 1. As β becomes smaller and the number of copies of W_{21} becomes larger, more and more difficult rank-one modification problems arise.

TEST 2 [27]. $D = \mathrm{diag}(1, 2 - \beta, 2 + \beta, 10/3)$ and $z = (2, \beta, \beta, 2)^T$. This example illustrates how badly Algorithm I can fail in computing the eigenvectors.

TEST 3. $n = 202; d_1 = 1, d_n = 10/3$, and the rest of the d_i's are of the form $2 \pm j\beta$, for $j = 1, 2, \ldots, 100$; and $z = (2, \beta, \ldots, \beta, 2)^T$. As $\beta > 0$ becomes very small, this matrix has $10\frac{1}{3}$ as an isolated eigenvalue; the remaining eigenvalues are clustered around 2. This generalization of TEST 2 is designed to show the stability of Algorithm II in the presence of an extremely large and tight cluster.

Tables 1, 2, and 3 summarize the results. For Algorithm II, the orthogonality and residual are small, and the vectors z and $\tilde{z}$ agree to machine precision. For Algorithm I, the residual is small, but the orthogonality can be very poor.

7. Some extensions. In this paper we have presented a novel, stable algorithm for solving the rank-one modification of the symmetric eigenproblem. The techniques developed have also been extended to the following:

- Cuppen's method for the symmetric tridiagonal eigenproblem [8], [9], [14], [18].
- An algorithm for the eigendecomposition of symmetric arrowhead matrices [25]; this is in turn the basic tool in the arrowhead divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem [3], [14], [16], [18].

TABLE 1
Results for TEST 1.

n	20	20	20	48	36
$-\log_{10}\beta$	7	7	7	7	7
$\mathcal{O}_I$	1.1×10^{4}	1.9×10^{6}	3.2×10^{2}	2.0×10^{7}	8.3×10^{9}
$\mathcal{R}_I$	4.2×10^{-2}	3.8×10^{-2}	3.8×10^{-2}	1.8×10^{-2}	1.6×10^{-2}
Z	8.8×10^{-4}	1.8×10^{-3}	1.8×10^{-3}	1.5×10^{-4}	5.5×10^{-4}
$\mathcal{O}_{II}$	1.1×10^{-1}	1.3×10^{-1}	9.3×10^{-2}	8.1×10^{-2}	6.9×10^{-2}
$\mathcal{R}_{II}$	7.5×10^{-2}	4.0×10^{-2}	4.8×10^{-2}	2.1×10^{-2}	5.2×10^{-2}

TABLE 2
Results for TEST 2.

n	4	4	4	4	4
$-\log_{10}\beta$	1	4	7	10	13
$\mathcal{O}_I$	1.0×10^{0}	1.7×10^{3}	1.4×10^{5}	9.6×10^{8}	1.7×10^{12}
$\mathcal{R}_I$	3.5×10^{-2}	2.2×10^{-1}	8.9×10^{-2}	2.0×10^{-1}	2.0×10^{-1}
Z	1.2×10^{-2}	4.4×10^{-2}	4.4×10^{-2}	4.4×10^{-2}	4.4×10^{-2}
$\mathcal{O}_{II}$	2.6×10^{-1}	5.2×10^{-1}	4.2×10^{-1}	4.2×10^{-1}	3.2×10^{-1}
$\mathcal{R}_{II}$	1.0×10^{-1}	2.3×10^{-1}	2.0×10^{-1}	1.6×10^{-1}	2.2×10^{-1}

TABLE 3
Results for TEST 3.

n	202	202	202
$-\log_{10}\beta$	3	8	15
Z	6.9×10^{-5}	6.1×10^{-5}	1.6×10^{-4}
$\mathcal{O}_{II}$	3.7×10^{-2}	2.5×10^{-2}	4.5×10^{-2}
$\mathcal{R}_{II}$	1.4×10^{-2}	3.6×10^{-3}	1.7×10^{-2}

- Algorithms for updating the SVD [6], [14], [15].
- A divide-and-conquer algorithm for computing the bidiagonal SVD [2], [4], [14], [17], [21].
- Algorithms for downdating the SVD [6], [14], [19].

Moreover it should be easy to apply these techniques to the divide-and-conquer algorithms for the unitary and orthogonal eigenproblems developed in [1], [4], and [12].

Acknowledgments. The authors thank Peter Tang of Argonne National Laboratory for providing them with the test matrices used in [27]; Ren-Cang Li of the University of California at Berkeley for sending them a copy of [23]; and Shivkumar Chandrasekaran and Ilse Ipsen of Yale University for some helpful discussions.

REFERENCES

[1] G. S. AMMAR, L. REICHEL, AND D. C. SORENSEN, *An implementation of a divide and conquer algorithm for the unitary eigenproblem*, ACM Trans. Math. Software, 18 (1992), pp. 292–307.

[2] P. ARBENZ, *Divide-and-conquer algorithms for the computation of the* SVD *of bidiagonal matrices*, in Vector and Parallel Computing, J. Dongarra, I. Duff, P. Gaffney, and S. McKee, eds., Ellis Horwood, Chichester, 1989, pp. 1–10.

[3] ———, *Divide and conquer algorithms for the bandsymmetric eigenvalue problem*, Parallel Comput., 18 (1992), pp. 1105–1128.

[4] P. ARBENZ AND G. H. GOLUB, *On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[5] J. L. BARLOW, *Error analysis of update methods for the symmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 598–618.

[6] J. R. BUNCH AND C. P. NIELSEN, *Updating the singular value decomposition*, Numer. Math., 31 (1978), pp. 111–129.

[7] J. R. BUNCH, C. P. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[8] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

[9] J. J. DONGARRA AND D. C. SORENSEN, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. s139–s154.

[10] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.

[11] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[12] W. B. GRAGG AND L. REICHEL, *A divide and conquer method for the unitary and orthogonal eigenproblems*, Numer. Math., 57 (1990), pp. 203–229.

[13] W. B. GRAGG, J. R. THORNTON, AND D. D. WARNER, *Parallel divide and conquer algorithms for the symmetric tridiagonal eigenproblem and bidiagonal singular value problem*, in Proc. 23rd Annual Pittsburgh Conference, University of Pittsburgh School of Engineering, Vol. 23, Modelling and Simulation, 1992.

[14] M. GU, *Studies in Numerical Linear Algebra*, Ph.D. thesis, Department of Computer Science, Yale University, New Haven, CT, 1993.

[15] M. GU AND S. C. EISENSTAT, *A fast algorithm for updating the singular value decomposition*, manuscript.

[16] ———, *A fast divide-and-conquer method for the symmetric tridiagonal eigenproblem*, 4th SIAM Conference on Applied Linear Algebra, Minneapolis, MN, Sept. 1991.

[17] ———, *A divide-and-conquer algorithm for the bidiagonal* SVD, SIAM J. Matrix Anal. Appl., 16 (1995).

[18] ———, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995).

[19] ———, *Downdating the singular value decomposition*, SIAM J. Matrix Anal. Appl., to appear.

[20] E. R. JESSUP, *Parallel Solution of the Symmetric Tridiagonal Eigenproblem*, Ph.D. thesis, Department of Computer Science, Yale University, New Haven, CT, 1989.

[21] E. R. JESSUP AND D. C. SORENSEN, *A parallel algorithm for computing the singular value decomposition of a matrix*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 530–548.

[22] W. KAHAN, *Rank-1 Perturbed Diagonal's Eigensystem*, manuscript, July 1989.

[23] R.-C. LI, *Solving secular equations stably and efficiently*, manuscript, Oct. 1992.

[24] K. LÖWNER, *Über monotone matrixfunktionen*, Math. Z., 38 (1934), pp. 177–216.

[25] D. P. O'LEARY AND G. W. STEWART, *Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices*, J. Comput. Phys., 90 (1990), pp. 497–505.

[26] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[27] D. C. SORENSEN AND P. T. P. TANG, *On the orthogonality of eigenvectors computed by divide-and-conquer techniques*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.

[28] G. W. STEWART, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[29] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1277–1289, October 1994

015

FAST SOLUTION OF CONFLUENT VANDERMONDE LINEAR SYSTEMS*

HAO LU†

Abstract. It is shown that the solution of confluent Vandermonde linear systems can be obtained by the Hermite evaluation of rational functions, which can actually be converted to the Hermite evaluation of two polynomials. Based on this result, divide and conquer methods are used to construct a fast algorithm for confluent Vandermonde linear systems. If fast polynomial multiplication and division (fast Fourier transform (FFT)) are used, the algorithm needs only $O(n \log n \log p)$ operations.

Key words. confluent Vandermonde linear system, divide and conquer, FFT

AMS subject classifications. 65F05, 65Y05, 68C25

1. Introduction.

Let $a_0, a_1, \ldots, a_p$ be $p+1$ numbers, $n_0, n_1, \ldots, n_p$ be $p+1$ positive integers and $p(x) = (1, x, \ldots, x^{n-1})^T$, where $n = \sum_{i=0}^{p} n_i$. The confluent Vandermonde matrix, denoted by $V(a_0, \ldots, a_p, n_0, \ldots, n_p)$ or, briefly V_c, is given by

$$V_c = (p(a_0), p'(a_0), \ldots, p^{(n_0-1)}(a_0), \ldots, p(a_p), p'(a_p), \ldots, p^{(n_p-1)}(a_p)) \tag{1.1}$$

(see [4], [8]). In the case of $n_0 = n_1 = \cdots = n_p = 1$, V_c yields the well-known Vandermonde matrix. If $p(x) = (p_1(x), p_2(x), \ldots, p_n(x))^T$, where $p_i(x)$ is a polynomial of degree $i-1$, (1.1) defines the confluent Vandermonde-like matrix V_c^l [13] and the Vandermonde-like matrix [9], [12] if $n_0 = n_1 = \cdots = n_p = 1$. Consider confluent Vandermonde linear systems

$$V_c u = b. \tag{1.2}$$

These systems are associated with the construction of quadrature formulae [1], [11], [15], [17] and the approximation of linear functionals [2], [21].

If Gaussian elimination for solving dense systems of linear equations is applied to (1.2), it requires $O(n^3)$ operations. Fortunately, it is shown that operations of solving some special linear systems such as Vandermonde systems [5], [16], confluent Vandermonde systems [4], Vandermonde-like systems [12], and confluent Vandermonde-like systems [13] can be further reduced. Based on forward and backward vector recursion, Björck, Elfving, and Pereyra [4], [5] presented some fast algorithms for both Vandermonde and confluent Vandermonde linear systems in the early 1970s. Their algorithms require $O(n^2)$ operations. In 1988 and 1990, Higham [12], [13] considered the generalization of Vandermonde linear systems and confluent Vandermonde linear systems, i.e., Vandermonde-like systems and confluent Vandermonde-like systems. He derived some $O(n^2)$ fast algorithms for both systems for the case where the polynomials $p_1(x)$, $p_1(x)$, $\ldots$, $p_n(x)$ satisfy a three-term recurrence relation that generalizes the early ones for $p_k(x) = x^{k-1}$. Recently, the author [16] showed that the solution of Vandermonde linear systems can be obtained by the evaluation of certain polynomials. We state the result here for use later in the paper. For convenience, quot$(A(x), B(x))$

* Received by the editors February 4, 1993; accepted for publication (in revised form) July 7, 1993. This work was partly supported by The Netherlands Organization for Pure Research grant 611-302-025.

† Department of Mathematics, Faculty of Mathematics and Informatics, Catholic University of Nijmegen, Toernooiveld, 6525 ED Nijmegen, The Netherlands (na.hlu@na-net.ornl.gov).

denotes the quotient of polynomial division $A(x)/B(x)$, i.e., ignoring the remainder $r(x)$: $A(x) = B(x)\text{quot}(A(x), B(x)) + r(x)$, throughout the paper.

THEOREM 1.1. *Let $V = V(a_1, a_2, \ldots, a_n)$ be the Vandermonde matrix of order n, $a_i \neq a_j$, $i \neq j, i, j = 1, 2, \ldots, n$, and*

$$L(x) = (x - a_1)(x - a_2) \cdots (x - a_n),$$

$$L_i(x) = \frac{L(x)}{x - a_i}, \qquad i = 1, 2, \ldots, n.$$

Then the solution of the Vandermonde linear system

$$Vu = b \tag{1.3}$$

is given by

$$u_i = \frac{g(a_i)}{L_i(a_i)}, \qquad i = 1, 2, \ldots, n, \tag{1.4}$$

where

$$g(x) = \text{quot}(L(x)b(x), x^n),$$

$$b(x) = b_1 x^{n-1} + b_2 x^{n-2} + \cdots + b_{n-1}x + b_n.$$

Based on this result, the author [16] gave a new fast algorithm for solving Vandermonde linear systems with $O(n \log^2 n)$ operations by using fast polynomial multiplication and division.

Let $r(x)$ be a rational function and $a_0, a_1, \ldots, a_p$ be $p + 1$ numbers, where p is a nonnegative integer. The Hermite evaluation of $r(x)$ is to compute

$$r^{(k)}(a_i), \quad k = 0, 1, \ldots, n_i - 1, \quad i = 0, 1, \ldots, p,$$

where $n_0, n_1, \ldots, n_p$ are positive integers.

The purpose of this paper is to extend Theorem 1.1 to confluent Vandermonde systems and to construct an asymptotically fast algorithm with $O(n \log n \log p)$ operations for the linear systems (1.2) by using the divide and conquer method. To the best of our knowledge, the method presented here is the fastest method known for solving confluent Vandermonde systems. In §2, we focus our attention on the proof that the solution of confluent Vandermonde linear systems can be obtained by the Hermite evaluation of rational functions, and we derive the inverse of confluent Vandermonde matrices as well as the determinant of the matrices. Based on the result given in §2, we convert the Hermite evaluation of rational functions to the Hermite evaluation of polynomials, which, as we will see, can be done only by the Hermite evaluation of two polynomials in §3. To obtain a fast algorithm for solving confluent Vandermonde linear systems, we use the divide and conquer method. In §4, we analyze the computational complexity of the algorithm presented in §3. It is shown that if fast polynomial multiplication and division are used in the algorithm, the algorithm needs only $O(n \log n \log p)$ operations.

2. Solution of confluent Vandermonde linear systems. As we have seen, the solution of Vandermonde linear systems can be obtained by evaluating certain polynomials so that a stable algorithm for the equations can be constructed easily. The question arises naturally: Does a similar result for confluent Vandermonde linear systems exist? The answer to this question is the purpose of this section.

Let $p(x) = (p_1(x), \ldots, p_n(x))^T$, where $p_k(x)$ is a polynomial of degree $k-1$ and denote

$$p[x] = p(x),$$

$$p[x_0, \ldots, x_k] = (p_1[x_0, \ldots, x_k], \ldots, p_n[x_0, \ldots, x_k])^T,$$

where $p_i[x_0, \ldots, x_k]$ is the kth divided difference of $p_i(x)$ at $x_0, \ldots, x_k$ defined by

$$p_i[x_0, \ldots, x_k] = \frac{p_i[x_1, \ldots, x_k] - g_i[x_0, \ldots, x_{k-1}]}{x_k - x_0}.$$

If $x_{i+1} - x_i = x_i - x_{i-1} = h(\neq 0)$, induction shows that

$$p[x_0, x_1, \ldots, x_k] = \frac{1}{k!\, h^k} \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} p(x_i) \tag{2.1}$$

and

$$\lim_{h \to 0} p[x_0, x_1, \ldots, x_k] = \frac{1}{k!} (\underbrace{0, \ldots, 0}_{k}, p_{k+1}^{(k)}(x_0), \ldots, p_n^{(k)}(x_0)). \tag{2.2}$$

Partitioning the confluent Vandermonde-like matrix V_c^l as

$$V_c^l = (B_0, B_1, \ldots, B_p),$$

where B_k is an $n \times n_k$ matrix with (i,j) entry $p_i^{(j-1)}(a_k)$, we immediately have the following limit equality concerning B_k from (2.2):

$$B_k = \lim_{h \to 0} Q_k \operatorname{diag}(1, 1, 2!, \ldots, (n_k - 1)!), \tag{2.3}$$

where Q_k is an $n \times n_k$ matrix with (i,j) entry

$$p_i[a_k, a_k + h, \ldots, a_k + (j-1)h].$$

Now we investigate the relation between Q_k and an $n \times n_k$ Vandermonde-like matrix $M_k(h)$ with (i,j) entry $p_i(a_k + (j-1)h)$. Equality (2.1) acts as the key for relating these two matrices. In fact, from (2.1) Q_k can be decomposed as the product of $M_k(h)$ and an $n_k \times n_k$ upper triangular matrix, i.e.,

$$Q_k = M_k(h) U_k, \tag{2.4}$$

where

$$U_k = \begin{pmatrix} 1 & -1 & 1 & \cdots & (-1)^{n_k-1} \\ & 1 & -2 & \cdots & (-1)^{n_k-2} \binom{n_k-1}{1} \\ & & 1 & \cdots & (-1)^{n_k-3} \binom{n_k-1}{2} \\ & & & \ddots & \vdots \\ & & & & 1 \end{pmatrix}$$

$$\operatorname{diag}\left(1, h^{-1}, \frac{1}{2!}h^{-2}, \ldots, \frac{1}{(n_k-1)!}h^{1-n_k}\right).$$

Therefore,

$$(Q_0, \ldots, Q_p) = (M_0(h), \ldots, M_p(h))\text{blockdiag}(U_0, \ldots, U_p). \tag{2.5}$$

In particular, (2.3) and (2.5) imply that

$$V_c^l = \lim_{h\to 0}(M_0(h), \ldots, M_p(h))\text{blockdiag}(\tilde{U}_0, \ldots, \tilde{U}_p). \tag{2.6}$$

where $\tilde{U}_i = U_i \operatorname{diag}(1, 1, 2!, \ldots, (n_i-1)!)$. Assume that $p_k(x) = x^{k-1} + a_{k,k-1}x^{k-2} + \cdots + a_{k0}$. A simple computation shows the following result.

PROPOSITION 2.1. *Let*

$$V_c^l = (p(a_0), p'(a_0), \ldots, p^{(n_0-1)}(a_0), \ldots, p(a_p), p'(a_p), \ldots, p^{(n_p-1)}(a_p))$$

be a confluent Vandermonde-like matrix, where $p(x) = (p_1(x), p_2(x), \ldots, p_n(x))^T$ *and* $p_k(x) = x^{k-1} + a_{k,k-1}x^{k-2} + \cdots + a_{k0}$. *Then*

$$\det(V_c^l) = \left(\prod_{i=0}^{p}\prod_{k=0}^{n_i-1} k!\right)\prod_{p\geq i>j\geq 0}(a_i - a_j)^{n_i n_j}. \tag{2.7}$$

In the case of $p_k(x) = x^{k-1}$, $(B_0, \ldots, B_p)$ and $(M_0(h), \ldots, M_p(h))$ yield a confluent Vandermonde matrix and a Vandermonde matrix, respectively. Our next theorem shows that the solution of confluent Vandermonde linear systems can be computed by the Hermite evaluation of rational functions.

THEOREM 2.2. *Let* $V_c u = b$ *be the confluent Vandermonde linear system given by* (1.2) *and*

$$\begin{aligned} l(x) &= \prod_{i=0}^{p}(x - a_i)^{n_i}, \\ l_i(x) &= \frac{l(x)}{(x-a_i)^{n_i}}, \qquad i = 0, 1, \ldots, p, \\ b(x) &= b_n + b_{n-1}x + \cdots + b_1 x^{n-1}, \\ q(x) &= \operatorname{quot}(l(x)b(x), x^n), \end{aligned}$$

where $a_i \neq a_j$, $i \neq j$, $i, j = 0, 1, \ldots, p$, *and* $n = \sum_{t=0}^{p} n_t$. *Then, the solution of the system is given by*

$$u_i = \frac{1}{(k-1)!(n_j-k)!}\left(\frac{q(x)}{l_j(x)}\right)^{(n_j-k)}\Bigg|_{x=a_j}, \tag{2.8}$$

$$i = m_j + k, \quad 0 \leq j \leq p, \quad 1 \leq k \leq n_j, \quad m_0 = 0, \quad m_j = \sum_{t=0}^{j-1} n_t,$$

Proof. Set

$$d_i = a_j + (k-1)h, \quad i = m_j + k, \quad 0 \le j \le p, \quad 1 \le k \le n_j,$$
$$\bar{l}(x) = \prod_{i=1}^{n}(x - d_i),$$
$$\bar{l}_i(x) = \frac{\bar{l}(x)}{x - d_i},$$
$$\bar{q}(x) = \operatorname{quot}(\bar{l}(x)b(x), x^n).$$

Consider a linear system

$$M(h)\operatorname{blockdiag}(\tilde{U}_0, \ldots, \tilde{U}_p)u(h) = b \tag{2.9}$$

where $M(h) = (M_0(h), \ldots, M_p(h))$, and denote

$$v(h) = (v_1(h), \ldots, v_n(h))^T = \operatorname{blockdiag}(\tilde{U}_0, \ldots, \tilde{U}_p)u(h). \tag{2.10}$$

Under the assumptions of Theorem 2.2, $M(h)$ becomes a nonsingular Vandermonde matrix if h is sufficiently small. Theorem 1.1 shows that

$$v_i(h) = \frac{\bar{q}(d_i)}{\bar{l}_i(d_i)}, \qquad i = 1, 2, \ldots, n. \tag{2.11}$$

On the other hand, a computation shows that the inverse of $\tilde{U}_j$ is of the form

$$\tilde{U}_j^{-1} = \operatorname{diag}(1, h, h^2, \ldots, h^{n_j-1}) \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ & 1 & 2 & \cdots & \binom{n_j-1}{1} \\ & & 1 & \cdots & \binom{n_j-1}{2} \\ & & & \ddots & \vdots \\ & & & & 1 \end{pmatrix}.$$

Hence,

$$\begin{aligned} u_i(h) &= h^{k-1} \sum_{t=k}^{n_j} \binom{t-1}{k-1} \frac{\bar{q}(d_t)}{\bar{l}_i(d_t)} \\ &= \frac{1}{h^{n_j-k}} \sum_{t=k}^{n_j} \frac{(-1)^{n_j-t}}{(t-1)!(n_j-t)!} \binom{t-1}{k-1} \frac{\bar{q}(d_t)}{\tilde{l}_j(d_t)} \\ &= \frac{1}{h^{n_j-k}} \frac{1}{(k-1)!(n_j-k)!} \sum_{t=k}^{n_j} \binom{n_j-k}{t-k} \frac{\bar{q}(d_t)}{\tilde{l}_j(d_t)} \\ &= \frac{1}{(k-1)!(n_j-k)!} \left(\frac{\bar{q}(x)}{\tilde{l}_j(x)} \right)^{(n_j-k)} \Bigg|_{x=\xi_{kj}}, \end{aligned}$$

where $a_j + (k-1)h \le \xi_{kj} \le a_j + (n_j-1)h$ and

$$\tilde{l}_j(x) = \frac{\bar{l}(x)}{\prod_{i=m_j+1}^{m_j+n_j}(x - d_i)}, \qquad j = 0, 1, \ldots, p.$$

Since $a_i \neq a_j$, $i \neq j$, $i,j = 0,1,\ldots,p$, Proposition 2.1 shows that V_c is nonsingular. Furthermore, because the set of all nonsingular matrices is an open set, we have

$$\begin{aligned} u_i &= \lim_{h\to 0} u_i(h) \\ &= \lim_{h\to 0} \frac{1}{(k-1)!(n_j-k)!} \left(\frac{\bar{q}(x)}{\tilde{l}_j(x)} \right)^{(n_j-k)} \Bigg|_{x=\xi_{kj}} \\ &= \frac{1}{(k-1)!(n_j-k)!} \left(\frac{q(x)}{l_j(x)} \right)^{(n_j-k)} \Bigg|_{x=a_j}. \end{aligned} \qquad \square$$

Let

$$e_i = (\underbrace{0,\ldots,0,1}_{i},0,\ldots,0), \qquad i = 1,2,\ldots,n,$$

and $V_c^{-1} = (b_{ij})$ if V_c is nonsingular. Clearly, $v_j = (b_{1j}, b_{2j}, \ldots, b_{nj})^T$ is the solution of $V_c v_j = e_j$. Proposition 2.1 and Theorem 2.2 show the following result on the inverse of confluent Vandermonde matrices.

COROLLARY 2.3. *The confluent Vandermonde matrix*

$$V_c = V(a_0,\ldots,a_p,n_0,\ldots,n_p)$$

defined by (1.1) *is nonsingular if and only if* $a_i \neq a_j$, $i \neq j$, $i,j = 0,1,\ldots,p$. *Let* $V_c^{-1} = (b_{ij})$ *if* V_c *is nonsingular. Then*

$$b_{ij} = \frac{1}{(k-1)!(n_t-k)!} \left(\frac{q_j(x)}{l_t(x)} \right)^{(n_t-k)} \Bigg|_{x=a_t}, \tag{2.12}$$

$$i = \sum_{r=0}^{t-1} n_r + k, \quad 1 \leq k \leq n_t, \quad 0 \leq t \leq p, \quad j = 1,2,\ldots,n,$$

where $q_j(x) = \mathrm{quot}(l(x), x^j)$, $j = 1,2,\ldots,n$, $l(x)$ *and* $l_i(x)$, $i = 1,2,\ldots,n$, *are given in Theorem* 2.2.

3. Algorithm. Based on the result given in the previous section, divide and conquer methods are used to construct a fast algorithm for solving confluent Vandermonde linear systems in this section. To this end, we denote

$$R_j(x) = \frac{q(x)}{l_j(x)}, \qquad j = 0,1,\ldots,p \tag{3.1}$$

and expand $R_j(x)$, $q(x)$, and $l_j(x)$ in Taylor series at a_j, i.e.,

$$R_j(x) = \sum_{t=0}^{n_j-1} r_{jt}(x-a_j)^t + O((x-a_j)^{n_j}), \tag{3.2a}$$

$$q(x) = \sum_{t=0}^{n_j-1} q_{jt}(x-a_j)^t + O((x-a_j)^{n_j}), \tag{3.2b}$$

$$l_j(x) = \sum_{t=0}^{n_j-1} l_{jt}(x-a_j)^t + O((x-a_j)^{n_j}). \tag{3.2c}$$

It follows from Theorem 2.2 that the solution of confluent Vandermonde linear systems is given by

$$u_i = \frac{r_{j,n_j-k}}{(k-1)!}, \quad i = \sum_{t=0}^{j-1} n_t + k, \quad k = 1,2,\ldots,n_j. \tag{3.3}$$

Comparing the coefficients of equality $R_j(x)l_j(x) = q(x)$ shows that the vector $r_j = (r_{j0}, r_{j1}, \ldots, r_{j,n_j-1})^T$ is the solution of the triangular Toeplitz linear system

$$T_j r_j = q_j, \tag{3.4}$$

where T_j is the triangular Toeplitz matrix of the form

$$T_j = \begin{pmatrix} l_{j0} & & & \\ l_{j1} & l_{jo} & & \\ \ddots & \ddots & \ddots & \\ l_{j,n_j-1} & \ddots & l_{j1} & l_{j0} \end{pmatrix},$$

or simply $\mathrm{tri}(l_{j0}, \ldots, l_{j,n_j-1})$ for convenience, $q_j = (q_{j0}, q_{j1}, \ldots, q_{j,n_j-1})^T$, and

$$q_{jk} = \frac{1}{k!} q^{(k)}(a_j), \quad k = 0, \ldots, n_j - 1, \quad j = 0, \ldots, p. \tag{3.5}$$

Equality $l(x) = (x-a_j)^{n_j} l_j(x)$ implies

$$l_{jk} = \frac{l_j^{(k)}(a_j)}{k!} = \frac{l^{(n_j+k)}(a_j)}{(n_j+k)!}, \quad k = 0, \ldots, n_j - 1, \quad j = 0, \ldots, p. \tag{3.6}$$

To construct a fast algorithm for confluent Vandermonde linear systems, we need a fast algorithm for the Hermite evaluation of polynomials such that the algorithm can easily be adjusted to evaluate two polynomials $q(x)$ and $l(x)$ by adding minimum operations. To do this, consider first the Hermite evaluation of polynomials.

Let $q(x)$ be a polynomial of degree at most $n-1$, $a_0, a_1, \ldots, a_p$ be $p+1$ distinct numbers, and $n_0, n_1, \ldots, n_p$ be $p+1$ positive integers. Consider the Hermite evaluation of $q(x)$

$$q^{(k)}(a_i), \quad k = 0, 1, \ldots, n_i - 1, \quad i = 0, 1, \ldots, p.$$

Without loss of generality, assume $n = \sum_{t=0}^{p} n_t = 2^m$ for some positive integer m. Denote

$$c_{m_i+j} = a_i, \quad m_i = \sum_{k=0}^{i-1} n_k, \quad j = 1, \ldots, n_i, \quad i = 0, \ldots, p,$$

$$T_{0i}(x) = x - c_i, \qquad i = 1, 2, \ldots, n,$$

$$T_{ji}(x) = T_{j-1,2i-1}(x) T_{j-1,2i}(x), \quad i = 1, \ldots, 2^{m-j}, \quad j = 1, 2, \ldots, m. \tag{3.7}$$

Dividing $q(x)$ by $T_{m-1,1}(x)$ and $T_{m-1,2}(x)$ and denoting the remainders by $r_{m-1,1}(x)$ and $r_{m-1,2}(x)$, respectively, i.e.,

$$q(x) = q_{m-1,1}(x)T_{m-1,1}(x) + r_{m-1,1}(x), \tag{3.8a}$$

$$q(x) = q_{m-1,2}(x)T_{m-1,2}(x) + r_{m-1,2}(x), \tag{3.8b}$$

one finds that

$$q^{(j)}(a_i) = r^{(j)}_{m-1,1}(a_i), \quad j = 0, 1, \ldots, n_i - 1, \quad i < t, \tag{3.9a}$$

$$q^{(j)}(a_i) = r^{(j)}_{m-1,2}(a_i), \quad j = 0, 1, \ldots, n_i - 1, \quad i > t, \tag{3.9b}$$

where t is the nonnegative integer such that $c_{2^{m-1}} = a_t$. We compute directly $q^{(j)}(a_t)$, $j = 0, 1, \ldots, n_t - 1$. Thus, repeating the procedure to $r_{m-1,1}(x)$ and $r_{m-1,2}(x)$ finishes the computation $q^{(j)}(a_i)$, $j = 0, 1, \ldots, n_i - 1$, $i \neq t$.

To obtain $r_{m-1,1}(x)$ and $r_{m-1,2}(x)$ from (3.8a) and (3.8b), we need two polynomial divisions of polynomials. It is somewhat expensive, though some $O(n \log n)$ algorithms exist for polynomial division [3]. Fortunately, preprocessing can be used to reduce operations in cases where all divisors satisfy (3.7). Given a polynomial $B(x)$ of degree n, by preprocessing $B(x)$, we define the computation of the quotient of x^{2n-1} divided by $B(x)$ (see [18]). Assume that $Q_{ji}(x)$ is the quotient of $x^{2^{j+1}-1}$ divided by $T_{ji}(x)$. It is not hard to see from (3.7) that

$$Q_{j-1,2i-1}(x) = \mathrm{quot}(T_{j-1,2i}(x)Q_{ji}(x), x^{2^j}), \tag{3.10a}$$

$$Q_{j-1,2i}(x) = \mathrm{quot}(T_{j-1,2i-1}(x)Q_{ji}(x), x^{2^j}) \tag{3.10b}$$

are the quotient of x^{2^j-1} divided by $T_{j-1,2i-1}(x)$ and x^{2^j-1} divided by $T_{j-1,2i}(x)$, respectively. After preprocessing, the polynomial division can be computed by using the following result (see [18] for details).

PROPOSITION 3.1. *Let*

$$A(x) = \sum_{i=0}^{\bar{n}} a_i x^i, \quad B(x) = \sum_{i=0}^{n} b_i x^i, \quad (a_{\bar{n}} \neq 0, b_n \neq 0, \bar{n} \geq n),$$

$D(x)$ be the result of preprocessing $B(x)$ and $K(x) = \mathrm{quot}(A(x), x^n)$. Then

$$Q(x) = \mathrm{quot}(D(x)K(x), x^{n-1}), \tag{3.11}$$

$$R(x) = A(x) - Q(x)B(x), \tag{3.12}$$

are the quotient and the remainder of division $A(x)/B(x)$, respectively.

Similarly, we can compute $l^{(n_i+j)}(a_i)$, $j = 0, 1, \ldots, n_i - 1$, $i = 0, 1, \ldots, p$ in the following way. Dividing $l(x)$ by $T^2_{m-1,1}(x)$ and $T^2_{m-1,2}(x)$, we have

$$l(x) = \tilde{q}_{m-1,1}(x)T^2_{m-1,1}(x) + \tilde{r}_{m-1,1}(x), \tag{3.13a}$$

$$l(x) = \tilde{q}_{m-1,2}(x)T^2_{m-1,2}(x) + \tilde{r}_{m-1,2}(x), \tag{3.13b}$$

where $\tilde{r}_{m-1,1}(x)$ and $\tilde{r}_{m-1,2}(x)$ are the remainders. Hence,

$$l^{(n_i+j)}(a_i) = \tilde{r}^{(n_i+j)}_{m-1,1}(a_i), \quad j = 0, 1, \ldots, n_i - 1, \quad i < t, \tag{3.14a}$$

$$l^{(n_i+j)}(a_i) = \tilde{r}^{(n_i+j)}_{m-1,1}(a_i), \quad j = 0, 1, \ldots, n_i - 1, \quad i > t, \tag{3.14b}$$

where t is the same as in (3.9). We then compute for t

$$l^{(n_t+j)}(a_t), \qquad j = 0, 1, \ldots n_t - 1,$$

in the same way as in the computation of $q^{(j)}(a_t)$. It is not necessary to preprocess $T^2_{ji}(x)$ after preprocessing $T_{ji}(x)$. In fact, let $\tilde{A}(x)$ and $\tilde{B}(x)$ be two polynomials of degree $\bar{n}$ and let n ($\bar{n} \geq 2n$), respectively, and let $\tilde{D}(x)$ be the result of preprocessing $\tilde{B}(x)$. Using Proposition 3.1 to $\tilde{A}(x)$ and $\tilde{B}(x)$, we obtain $\tilde{Q}(x)$ the quotient of $\tilde{A}(x)/\tilde{B}(x)$. $\bar{Q}(x)$ the quotient of $\tilde{Q}(x)/\tilde{B}(x)$ is clearly the quotient of $\tilde{A}(x)/\tilde{B}^2(x)$. Therefore, the corresponding remainder can be computed by

$$\tilde{R}(x) = \tilde{A}(x) - \bar{Q}(x)\tilde{B}^2(x). \tag{3.15}$$

Based on our discussion, we now give an algorithm for confluent Vandermonde linear systems.

ALGORITHM 3.2. The algorithm is divided into two stages. At stage I, the algorithm converts all polynomials defined by (3.7) into the form $\sum h_i x^i$, computes $r_{m1}(x)$, i.e., $q(x)$ given in Theorem 2.2, and preprocesses $T_{m1}(x)$. At stage II, we continue to preprocess $T_{ji}(x)$ for $j \leq m-1$ by using (3.10). For clarity, functions **Div1** and **Div2**, which are based on Proposition 3.1 and our later discussion, are used to compute the remainders $r_{ji}(x)$ and $\tilde{r}_{ji}(x)$, respectively. For the sake of reducing the operations as explained in §4, the remainders $\tilde{r}_{m-1,i}(x)$, $i = 1, 2$, are computed directly by polynomial division. After finding a proper a_l, the algorithm calls the algorithm **Solution** to solve the corresponding triangular Toeplitz linear system (3.4) and obtain u_i ($m_l + 1 \leq i \leq m_l + n_l$), the solution of the confluent Vandermonde linear system (1.2). Again for clarity, we delete the details for the algorithm **Solution** here, which is given in the next section.

Stage I: $c_{m_i+j} = a_i$, $m_0 = 0$, $m_i = \sum_{k=0}^{i-1} n_k$,
$j = 1, \ldots, n_i$, $i = 0, \ldots, p$,
$T_{0i} = x - c_i$, $i = 1, 2, \ldots, n$, $b = b_1 x^{n-1} + \cdots + b_{n-1}x + b_n$,
$S_{m1} = \{0, 1, \ldots, p\}$
For $j = 1 : 1 : m$
  For $i = 1 : 1 : 2^{m-j}$
    $T_{ji} = T_{j-1,2i-1} T_{j-1,2i}$
  endfor i
endfor j
$r_{m1} = \text{quot}(T_{m1}b, x^n)$
$\tilde{r}_{m1} = T_{m1}$
$Q_{m1} = \text{quot}(x^{2^{m+1}-1}, T_{m1})$
Stage II: For $j = m : -1 : m - \lceil \log(p+1) \rceil + 1$
  For $i = 1 : 1 : 2^{m-j}$

if $S_{ji} = \{l\}$ then
call **Solution**$(r_{ji}, \tilde{r}_{ji}, a_l, n_l)$
For $k = 1 : 1 : n_l$
$u_{m_l+k} = v_{n_l-k}/(k-1)!$
endfor k
$S_{ji} = \phi$
elseif $S_{ji} \neq \phi$ then
$Q_{j-1,2i-1} = \text{quot}(T_{j-1,2i}Q_{ji}, x^{2^j})$
$Q_{j-1,2i} = \text{quot}(T_{j-1,2i-1}Q_{ji}, x^{2^j})$
For $k = 2i-1, 2i$
$r_{j-1,k} = \mathbf{Div1}(r_{ji}, T_{j-1,k}, Q_{j-1,k}, 2^{j-1})$
if $j = m$ then
$\tilde{r}_{j-1,k} \equiv \tilde{r}_{ji} \pmod{T^2_{j-1,k}}$
else
$\tilde{r}_{j-1,k} = \mathbf{Div2}(\tilde{r}_{ji}, T_{j-1,k}, Q_{j-1,k}, 2^{j-1})$
endif
endfor k
if $c_{(2i-1)2^{j-1}} = a_l$ and $l \in S$ then
call **Solution**$(r_{ji}, \tilde{r}_{ji}, a_l, n_l)$
For $k = 1 : 1 : n_l$
$u_{m_l+k} = v_{n_l-k}/(k-1)!$
endfor k
$S_{j-1,2i-1} = \{t\colon\ t \in S_{ji} \text{ and } t < l\}$
$S_{j-1,2i} = \{t\colon\ t \in S_{ji} \text{ and } t > l\}$
endif
endif
endfor i
endfor j
Function **Div1**$(A(x), B(x), Q(x), n)$
$K(x) = \text{quot}(A(x), x^n)$
$P(x) = \text{quot}(K(x)Q(x), x^{n-1})$
$R(x) = A(x) - P(x)B(x)$
return $R(x)$
end
Function **Div2**$(A(x), B(x), Q(x), n)$
$K(x) = \text{quot}(A(x), x^n)$
$P_1(x) = \text{quot}(K(x)Q(x), x^{2n-1})$
$P(x) = \text{quot}(P_1(x)Q(x), x^{n-1})$
$R(x) = A(x) - P(x)B^2(x)$
return $R(x)$
end
Algorithm **Solution**$(A(x), B(x), n, a)$
for $k = 0, 1, \ldots, n-1$
Compute $a_k = \frac{1}{(n+k)!}A^{(n+k)}(a)$, $b_k = \frac{1}{k!}B^{(k)}(a)$
Solve triangular Toeplitz linear system
$\text{tri}(a_0, a_1, \ldots, a_{n-1})v = (b_0, b_1, \ldots, b_{n-1})^T$
end

where $\lceil x \rceil$ is the integer ceiling function of x.

4. Computational complexity. In this section, we analyze the computational complexity of Algorithm 3.2. The following two propositions estimate the operations needed by the algorithm if the FFT is used to compute polynomial multiplications and polynomial divisions in the algorithm.

PROPOSITION 4.1. *If fast polynomial multiplication and division are used, Stage* I *of Algorithm* 3.2 *needs at most* $O(n \log n \log p)$ *operations.*

Proof. Recall our algorithm, where $r_{m1}(x)$ and $Q_{m1}(x)$ can be computed by using directly fast polynomial multiplication and division, respectively. Both need $O(n \log n)$ operations.

Let $A(x)$ and $B(x)$ be two polynomials of degree $\bar{n}$ and $\bar{m}$ ($\bar{n} \geq \bar{m} \geq 2$), respectively. If the FFT is used to compute $A(x)B(x)$, it is well known that there exists a constant C_1 independent of $\bar{n}$ and $\bar{m}$ such that

$$C(\bar{n}, \bar{m}) \leq C_1 \bar{n} \log \bar{m}.$$

Denote the operations needed to compute all $T_{ji}(x)$ at Stage I by $T(n, p)$. In the case of $p = 0$,

$$T_{ji}(x) = (x-a)^{2^j} = \sum_{t=0}^{2^j} (-1)^t a^t \begin{pmatrix} 2^j \\ t \end{pmatrix} x^{2^j - t},$$

which can clearly be computed by using $C_2'(2^j + 1)$ operations. Thus the computation of all $T_{ji}(x)$ in this case needs only $C_2 n$ operations, where C_2' and C_2 are constants independent of n. We claim generally that

$$T(n, p) \leq C(n \log n \log(p+1) + n), \tag{4.1}$$

where $C = \max(C_1, C_2)$.

Note first that (4.1) holds for $p = 0$ implies that (4.1) is true if $n + p = 1$. We prove (4.1) by induction on $n + p$.

Assume all $T_{ji}(x)$, $j \leq m-1$ have been finished and let

$$T_{m-1,1}(x) = (x-a_0)^{n_0} \dots (x-a_{t-1})^{n_{t-1}} (x-a_t)^{k_t},$$

$$T_{m-1,2}(x) = (x-a_t)^{\bar{k}_t} (x-a_{t+1})^{n_{t+1}} \dots (x-a_p)^{n_p},$$

where $0 \leq k_t \leq n_t - 1$ and $\bar{k}_t = n_i - k_t$. Clearly, $T_{m-1,1}(x) = (x-a_0)^{\frac{n}{2}}$ if $t = 0$, or $t = 1$ and $k_1 = 0$, and $T_{m-1,2} = (x-a_p)^{\frac{n}{2}}$ if $t = p$. In the case of $t = 0$ or $t = 1$ and $k_1 = 0$, all $T_{ji}(x)$, $1 \leq j \leq m-1, 1 \leq i \leq 2^{m-j-1}$ can be computed by using at most $C\frac{n}{2}$ operations. Hence

$$\begin{aligned} T(n,p) &\leq T\left(\frac{n}{2}, p\right) + C\frac{n}{2} + C\frac{n}{2}\log\frac{n}{2} \\ &\leq C\frac{n}{2}\log\frac{n}{2}\log(p+1) + C\frac{n}{2} + C\frac{n}{2}\log\frac{n}{2} \\ &\leq C(n \log n \log(p+1) + n). \end{aligned}$$

If $t = p$, (4.1) is derived in the same way. Otherwise,

$$T(n,p) \leq T\left(\frac{n}{2}, p_1\right) + T\left(\frac{n}{2}, p_2\right) + C\frac{n}{2}\log\frac{n}{2}$$

$$\leq C\frac{n}{2}\log\frac{n}{2}\log(p_1+1)+C\frac{n}{2}\log\frac{n}{2}\log(p_2+1)+C\frac{n}{2}\log\frac{n}{2}+Cn$$

$$= Cn\log\frac{n}{2}\log(2(p_1+1)(p_2+1))^{\frac{1}{2}}+Cn$$

$$\leq Cn\log n\log(2(p_1+1)(p_2+1))^{\frac{1}{2}}+Cn,$$

where $p_1 \leq t$, $p_2 = p-t$. Since $p_1 \geq 1$, $p_2 \geq 1$, and $p_1+p_2 \leq p$, it is easy to check that $(2(p_1+1)(p_2+1))^{\frac{1}{2}} \leq p+1$. Equation (4.1) follows immediately. The overall computational cost of Stage I is $T(n,p)+O(n\log n) = O(n\log n\log p)$. □

PROPOSITION 4.2. *If fast polynomial multiplication and division are used, Stage* II *of Algorithm* 3.2 *needs at most* $O(n\log n\log p)$ *operations.*

Proof. Denote the operations needed for fixed j,i at Stage II by $\bar{T}(j,i)$. The algorithm computes $\tilde{r}_{m-1,i}(x)$, $i=1,2$ by

$$r_{m1}(x)(\bmod T^2_{m-1,i}(x)), \qquad i=1,2.$$

Since $\deg(T^2_{m-1,i}(x)) = \deg(r_{m1}(x))$, after the multiplications $T^2_{m-1,i}(x)$, $i=1,2$, which need $O(n\log n)$ operations, such computation can be done by using $O(n)$ operations. This is the reason we compute $\tilde{r}_{m-1,1}(x)$ and $\tilde{r}_{m-1,2}(x)$ by using polynomial division directly. As for the algorithm **Solution**$(r_{ji},\tilde{r}_{ji},a_l,n_l)$, when Algorithm 3.2 calls **Solution**$(r_{ji},\tilde{r}_{ji},a_l,n_l)$, $2^j \geq n_l$. The total number of operations of performing **Solution** is at most $O(2^j j)$. To see this, let $A(x) = \sum_{i=0}^{\bar{n}} \bar{a}_i x^i$ be a polynomial of degree at most $\bar{n}$. Consider the computation

$$\begin{aligned}
&A(a) = \bar{a}_0+\bar{a}_1 a+\cdots+\bar{a}_{\bar{n}}a^{\bar{n}},\\
&A'(a) = \bar{a}_1+2\bar{a}_2 a+\cdots+\bar{n}\bar{a}_{\bar{n}}a^{\bar{n}-1},\\
&\cdots\cdots,\\
&A^{(\bar{m})}(a) = \bar{m}!\bar{a}_{\bar{m}}+\cdots+\bar{n}\ldots(\bar{n}-\bar{m}+1)a_{\bar{n}}a^{\bar{n}-\bar{m}},
\end{aligned}$$

which can be written as a product of an $(\bar{m}+1)\times(\bar{n}+1)$ Toeplitz matrix with a vector as follows.

$$\begin{pmatrix} A(a)\\ A'(a)\\ \vdots\\ A^{(\bar{m})}(a)\end{pmatrix} = \begin{pmatrix} \bar{n}!\bar{a}_{\bar{n}} & \cdots & \bar{m}!\bar{a}_{\bar{m}} & \cdots & \bar{a}_1 & \bar{a}_0\\ 0 & \bar{n}!\bar{a}_{\bar{n}} & \cdots & \bar{m}!\bar{a}_{\bar{m}} & \cdots & \bar{a}_1\\ \vdots & \ddots & \ddots & & \ddots & \vdots\\ 0 & \cdots & 0 & \bar{n}!\bar{a}_{\bar{n}} & \cdots & \bar{m}!\bar{a}_{\bar{m}}\end{pmatrix}\begin{pmatrix} \frac{a^{\bar{n}}}{\bar{n}!}\\ \vdots\\ a\\ 1\end{pmatrix}.$$

Hence, polynomial multiplication can be applied to such computation. The cost is $O(\bar{n}\log\bar{m})$. Since $r_{ji}(x)$ and $\tilde{r}_{ji}(x)$ are polynomials of degree at most 2^j-1 and $2^{2j}-1$, respectively, the computation of $r^{(t)}_{ji}(a_l)$ and $\tilde{r}^{(t)}_{ji}(a_l)$, $t=0,1,\ldots,n_l-1$ needs at most $O(2^j\log n_l) \leq O(2^j j)$ operations. Triangular Toeplitz linear systems can actually be solved by using polynomial division (see [3] for details). Therefore, solving the linear system included in the algorithm **Solution** needs $O(n_l\log n_l) \leq O(2^j j)$ operations. The rest of the computation at Stage II is multiplication of polynomials of degree at most 2^{2j}. We thus have $\bar{T}(j,i) \leq O(2^j j)$. The overall operation at Stage II is bounded by

$$\sum_{j=m-\lceil\log(p+1)\rceil+1}^{m}\ \sum_{i=0}^{2^{m-j}} O(2^j j) = O(n\log n\log p),$$

which completes the proof. □

Propositions 4.1 and 4.2 show that Algorithm 3.2 needs only $O(n \log n \log p)$ operations if fast polynomial multiplication and division are used.

Acknowledgment. I am grateful to Dr. Nicholas J. Higham for valuable comments on the manuscript and for offering me the reference [13] and the Bibliography on Vandermonde Matrices, and to Professor Gene H. Golub for pointing out [10].

REFERENCES

[1] C. T. H. Baker and M. S. Derakhshan, *Fast generation of quadrature rules with some special properties*, in Numerical Integration: Recent Developments, Software and Applications, P. Keast and G. Fairweather, eds., D. Reidel Publishing Company, Dordrecht, Holland, 1987, pp. 53–60.

[2] C. Ballester and V. Pereyra, *On the construction of discrete approximations to linear differential expressions*, Math. Comp., 21 (1967), pp. 297–302.

[3] D. Bini and V. Pan, *Polynomial division and its computational complexity*, J. Complexity, 2 (1986), pp. 179–203.

[4] Å. Björck and T. Elfving, *Algorithms for confluent Vandermonde systems*, Numer. Math., 21 (1973), pp. 130–137.

[5] Å. Björck and V. Pereyra, *Solution of Vandermonde systems of equations*, Math. Comp., 24 (1970), pp. 893–903.

[6] C. J. Demeure, *Fast QR factorization of Vandermonde matrices*, Linear Algebra Appl., 122/3/4 (1989), pp. 165–194.

[7] G. Galimberti and V. Pereyra, *Solving confluent Vandermonde systems of Hermite type*, Numer. Math., 18 (1971), pp. 44–60.

[8] W. Gautschi, *On inverse of Vandermonde and confluent Vandermonde matrices*, Numer. Math., 4 (1962), pp. 117–123.

[9] ———, *The condition of Vandermonde-like matrices involving orthogonal polynomials*, Linear Algebra Appl., 52/53 (1983), pp. 293–300.

[10] S.-Å. Gustafson, *Control and estimation of computational errors in the evaluation of interpolation formulae and quadrature rules*, Math. Comp., 24 (1970), pp. 847–854.

[11] N.J. Higham, *Error analysis of the Björck–Pereyra algorithm for solving Vandermonde systems*, Numer. Math., 50 (1987), pp. 613–632.

[12] ———, *Fast solution of Vandermonde-like systems involving orthogonal polynomials*, IMA J. Numer. Anal., 8 (1988), pp. 473–486.

[13] ———, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 23–41.

[14] ———, *Iterative refinement enhances the stability of QR factorization methods for solving linear equations*, BIT, 31 (1991), pp. 447–468.

[15] J. Kautsky and S. Elhay, *Calculation of weights of interpolatory quadratures*, Numer. Math., 40 (1982), pp. 407–422.

[16] H. Lu, *Computational complexity of Vandermonde linear systems*, Science Bulletin of China, 9 (1990), pp. 654–656.

[17] J.N. Lyness, *Some quadrature rules for finite trigonometric and related integrals*, in Numerical Integration: Recent Developments, Software and Applications, P. Keast and G. Fairweather, eds., D. Reidel Publishing Company, Dordrecht, Holland, 1987, pp. 17–33.

[18] R. Moenck and A. B. Borodin, *Fast modular transforms via division*, J. Comput. System Sci., 8 (1974), pp. 366–386.

[19] L. Reichel and G. Opper, *Chebyshev-Vandermonde systems*, Math. Comp., 57 (1991), pp. 703–721.

[20] W.P. Tang and G.H. Golub, *The block decomposition of a Vandermonde and its applications*, BIT, 21 (1981), pp. 505–517.

[21] J.F. Traub, *Associated polynomials and uniform methods for the solution of linear problems*, SIAM Rev., 8 (1966), pp. 277–301.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1290–1309, October 1994

016

CONDITION AND ACCURACY OF ALGORITHMS FOR COMPUTING SCHUR COEFFICIENTS OF TOEPLITZ MATRICES*

I. GOHBERG†, I. KOLTRACHT‡, AND D. XIAO§

Abstract. A formula for the condition number of Schur coefficients of a positive definite Toeplitz matrix is obtained and an efficient algorithm for computing the condition number is given. New bounds of backward roundoff errors in Schur and Levinson algorithms for computing Schur coefficients are presented. These bounds, together with the condition number, provide a posteriori estimate of the error in computed Schur coefficients. Numerical comparison of Schur and Levinson algorithms with the LDL^T algorithm also indicates their forward stability.

Key words. Toeplitz matrix, Schur coefficients, condition of the problem, stability of algorithms

AMS subject classifications. 65F05, 65F30, 65G05, 15A12

1. Introduction.

Associated with an $(n+1)\times(n+1)$ positive definite Toeplitz matrix

$$A_n = \{a_{|i-j|}\}_{i,j=0}^{n},$$

there are n parameters $c_1, c_2, \ldots, c_n$ called Schur, or reflection, or partial autocorrelation coefficients (for definition see §2). They appear in a wide variety of applications in science and engineering, such as theory of analytic functions, geophysics, speech processing, statistics, transmission lines and others (see, for example, [1], [3], [9], [11], [13], [15], and [16]). The Schur coefficients are called here S-coefficients and are denoted by $c = (c_1, \ldots, c_n)$.

In the present paper we analyze accuracy of computing S-coefficients by two standard algorithms, the Schur [16] and the Levinson [12] algorithms that are described in §2. In fact, these algorithms can be used to define S-coefficients. We also show in §2 that S-coefficients can be determined by the LDL^T decomposition of a certain positive definite matrix defined by $a_0, \ldots, a_n$. The Schur and Levinson algorithms are known as fast algorithms since they give the S-coefficients in $O(n^2)$ arithmetic operations. The algorithm based on LDL^T factorization is of $O(n^3)$.

To analyze the accuracy of computed S-coefficients, it is useful to consider the condition number of the corresponding map

$$F : (a_0, \ldots, a_n) \to (c_1, \ldots, c_n),$$

at the point $a = (a_0, \ldots, a_n)$, which is given by the expression

$$k(F, a) = \frac{\|F'(a)\|\|a\|}{\|c\|}. \tag{1.1}$$

* Received by the editors May 28, 1992; accepted for publication (in revised form) July 20, 1993.

† Department of Mathematics, Tel Aviv University, Ramat Aviv, 69978, Tel Aviv, Israel (gohberg@taurus.bitnet). The research of this author was supported in part by National Science Foundation grant DMS-9007030.

‡ Department of Mathematics, University of Connecticut, Storrs, Connecticut, 06269 (koltrach@uconnvm.bitnet). The research of this author was supported in part by National Science Foundation grant DMS-9007030.

§ Department of Mathematics, University of Connecticut, Storrs, Connecticut, 06269 (dxiao@uconnvm.bitnet). The research of this author was supported in part by National Science Foundation grants DMS-9007030 and DMS-8901860.

Here $F'(a)$ is the derivative of F at a and $\|F'(a)\|$ is an operator norm of a linear map induced by vector norms of a and Fa. If x is a perturbation of a such that

$$\|x - a\| \leq \epsilon \|a\|,$$

then

$$\frac{\|Fx - Fa\|}{\|Fa\|} \leq k(F, a)\epsilon + o(\epsilon), \tag{1.2}$$

(see, for example, Belitzkii and Lyubich [20]). In §2 we obtain a formula for the derivative of the map of S-coefficients and we give an efficient algorithm for computing the condition number $k(F, a)$. We also find lower and upper bounds for $k(F, a)$ that differ, at most, by a factor of n from lower and upper bounds for the usual condition number of $A_{n-1} = \{a_{|i-j|}\}_{i,j=0}^{n-1}$, $k(A_{n-1}) = \|A_{n-1}\|\|A_{n-1}^{-1}\|$, given by Cybenko [4]. Extensive numerical experiments in §5 show that the ratio of $k(A_{n-1})$ and $k(F, a)$ is, in fact, of order unity; therefore, we presume that there is little difference between $k(A_{n-1})$ and $k(F, a)$. (Similar results for the condition numbers of the inversion of a positive definite Toeplitz matrix and the solution of Yule–Walker equations can be found in Gohberg and Koltracht [5] and Golhberg, Koltracht, and Xiao [7], respectively.)

In §4, we present the backward roundoff error analysis for Schur and Levinson algorithms. The obtained bounds improve previously known results by Bultheel [2]. The essential factor of the backward error bound in the Schur algorithm is $\prod_{j=1}^{n-1}(1 + |\bar{c}_j|)$; in the Levinson algorithm it is $(\prod_{j=1}^{n-1}(1 + |\bar{c}_j|))^2$, where $\bar{c}_1, \ldots, \bar{c}_n$ are the computed S-coefficients. Since the essential factor is squared in the error bound for the Levinson algorithm, we conclude that the Schur algorithm might be more trustworthy than the Levinson algorithm. (For the general definition of trustworthiness, see Stoer and Bulirsch [17, §1.3].) This conclusion is supported by numerical experiments in §5 that consistently show higher accuracy in the Schur algorithm. The obtained error bounds do not imply that either of the algorithms is backward stable on the class of all positive definite Toeplitz matrices; indeed, the attainable maximum for $\prod_{j=1}^{n-1}(1 + |\bar{c}_j|)$ is 2^{n-1}. The obtained backward error bounds have, however, the following practical significance: If the computed S-coefficients (say, by the Schur algorithm) satisfy the inequality

$$4n^2 \prod_{j=1}^{n-1}(1 + |\bar{c}_j|)k(F, a)u \leq 10^{-p}, \tag{1.3}$$

where u is the unit roundoff error, then at least p correct figures can be guaranteed in the computed results.

Further insight into numerical behaviour of Schur and Levinson algorithms can be obtained by comparing them with the numerically stable LDL^T algorithm of Martin, Peters, and Wilkinson [14]. Under the presumption that $k(F, a) \approx k(A_{n-1})$, this algorithm is forward stable for computing S-coefficients. Namely, the error in the computed vector of S-coefficients, $\bar{c}$, is given by

$$\frac{\|c - \bar{c}\|}{\|c\|} \leq Kn^{3/2}k(A_{n-1})u + o(u), \tag{1.4}$$

where K is a constant of order unity. Numerical comparisons in §5 suggest that Schur and Levinson algorithms are also forward stable although, in the majority of the

experiments, the LDL^T type algorithm is more accurate than the Schur algorithm, which in turn is more accurate than the Levinson algorithm.

The effects of finite precision arithmetic on the computation of S-coefficients from underlying time series rather than from the autocorrelation sequence and in fixed point rather than floating point arithmetic are studied by Alexander and Rhee [18] and Rialan and Scharf [19].

2. S-coefficients. Let $A_n = \{a_{|i-j|}\}_{i,j=0}^n$ be a real symmetric positive definite Toeplitz matrix. The S-coefficients of A_n are defined recursively. One such recursion is the Schur algorithm [17].

ALGORITHM 1

1. Start with

$$p_0(0) = q_0(0) = 1, \qquad E_0 = a_0,$$

$$p_0(i) = q_0(i) = -a_i, \qquad i = 1, \dots, n.$$

2. For $k = 0, 1, \dots, n-2$ let

$$c_{k+1} = p_k(k+1)/E_k,$$

$$E_{k+1} = (1 - c_{k+1}^2)E_k,$$

$$\begin{bmatrix} p_{k+1}(k+2) & q_{k+1}(k+2) \\ \vdots & \vdots \\ p_{k+1}(n) & q_{k+1}(n) \end{bmatrix} = \begin{bmatrix} p_k(k+2) & q_k(k+1) \\ \vdots & \vdots \\ p_k(n) & q_k(n-1) \end{bmatrix} \begin{bmatrix} 1 & c_{k+1} \\ c_{k+1} & 1 \end{bmatrix}.$$

3. $c_n = p_{n-1}(n)E_{n-1}^{-1}$.

The numbers $c_1, \dots, c_n$ are the S-coefficients of A_n. The S-coefficients can also be calculated (see, for example, Gohberg, el al. [6]) by the Levinson [1] algorithm.

ALGORITHM 2

1. $q_0(0) = 1, \quad E_0 = a_0$.
2. For $k = 1, 2, \dots, n$

$$c_k = -E_{k-1}^{-1} \sum_{j=0}^{k-1} a_{j+1} q_{k-1}(j),$$

$$\begin{bmatrix} q_k(0) \\ \vdots \\ q_k(k) \end{bmatrix} = \begin{bmatrix} 0 \\ q_{k-1}(0) \\ \vdots \\ q_{k-1}(k-1) \end{bmatrix} + c_k \begin{bmatrix} q_{k-1}(k-1) \\ \vdots \\ q_{k-1}(0) \\ 0 \end{bmatrix},$$

$$E_k = E_{k-1} * (1 - c_k^2).$$

We use the same notation $q_k(0),\dots,q_k(k)$ in Algorithm 2 and $q_k(k+1),\dots,q_k(n)$ in Algorithm 1 because, if we put them together, these numbers satisfy the equation

$$\begin{bmatrix} A_k & 0 \\ A_k^c & I \end{bmatrix} \begin{bmatrix} q_k(0) \\ \vdots \\ q_k(k) \\ \vdots \\ q_k(n) \end{bmatrix} = E_k \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $A_k = \{a_{|i-j|}\}_{i,j=0}^{k}$, $A_k^c = \{a_{|i-j|}\}_{i=k+1,\dots,n}^{j=0,\dots,k}$, and 1 in the right-hand side is in position $k+1$. The above equation can be used to obtain the LDL^T factorization of A_n and the L^TDL factorization of A_n^{-1} in terms of the quantities computed in Algorithms 1 and 2, respectively. (See, for example, [6].) In fact, it holds that

$$A_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -q_0(1)/E_0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ -q_0(n)/E_0 & -q_1(n)/E_1 & \dots & 1 \end{bmatrix} \begin{bmatrix} E_0 & 0 & \dots & 0 \\ & E_1 & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ 0 & 0 & \dots & E_n \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & -q_0(1)/E_0 & \dots & -q_0(n)/E_0 \\ 0 & 1 & \dots & -q_1(n)/E_1 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \tag{2.1}$$

and

$$A_n^{-1} = \begin{bmatrix} 1 & q_1(0) & \dots & q_n(0) \\ 0 & 1 & \dots & q_n(1) \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} E_0^{-1} & 0 & \dots & 0 \\ & E_1^{-1} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & E_n^{-1} \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & \dots & 0 \\ q_1(0) & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ q_n(0) & q_n(1) & \dots & 1 \end{bmatrix}, \tag{2.2}$$

which are used later in the paper. Note that the following inequality holds:

$$0 < E_k = a_0 \prod_{j=1}^{k} (1 - c_j^2), \qquad k = 1,\dots,n;$$

hence, $|c_j| < 1$, $j = 1,\dots,n$. It follows from the Levinson algorithm that the S-coefficients do not change when the matrix A_n is multiplied by a positive real number; therefore, it is often assumed that $a_0 = 1$. In fact, there is a one-to-one correspondence between all n-tuples $(c_1,\dots,c_n)$ such that $|c_j| < 1$, $j = 1,\dots,n$, and all

n-tuples $(a_1, \ldots, a_n)$ such that $A = \{a_{|i-j|}\}_{i,j=0}^{n}$, $a_0 = 1$ is positive definite (see, for example, Koltracht and Lancaster [10].) Algorithms 1 and 2 are two standard methods for computing S-coefficients. They are known as fast algorithms, since they give S-coefficients in $O(n^2)$ arithmetic operations. Another representation of S-coefficients can be obtained as follows.

THEOREM 1. *Let $A_n = \{a_{|i-j|}\}_{i,j=0}^{n}$ be a positive definite Toeplitz matrix. Let*

$$W = \begin{bmatrix} & & & a_1 \\ & A_{n-1} & & \vdots \\ & & & a_n \\ a_1 & \ldots & a_n & a_0 \end{bmatrix}$$

and let $W = LDL^T$, where $L = \{L_{jk}\}_{j,k=0}^{n}$ is a lower triangular matrix with unit diagonal and where D is a diagonal matrix. Then the S-coefficients of A_n admit the following representation:

$$c_k = -L_{n,k-1}, \qquad k = 1, \ldots, n.$$

Proof. Let $A_{n-1} = L_{n-1}D_{n-1}L_{n-1}^T$. Using the representation of S-coefficients in Algorithm 2 and comparing with (2.2), we get

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = - \begin{bmatrix} E_0^{-1} & 0 & \ldots & 0 \\ & E_1^{-1} & \ldots & 0 \\ \vdots & \vdots & \ldots & 0 \\ 0 & 0 & \ldots & E_{n-1}^{-1} \end{bmatrix} \begin{bmatrix} 1 & 0 & \ldots & 0 \\ q_1(0) & 1 & \ldots & 0 \\ \vdots & \vdots & \ldots & 0 \\ q_{n-1}(0) & q_{n-1}(1) & \ldots & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

$$= -D_{n-1}^{-1}L_{n-1}^{-1} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}.$$

It is straightforward to check that

$$L = \begin{bmatrix} L_{n-1} & 0 \\ b^T & 1 \end{bmatrix},$$

where

$$b^T = [a_1, \ldots, a_n]D_{n-1}^{-1}L_{n-1}^{-T}.$$

Transposing the last equation we see that $b = -c$; hence, the theorem is proved. □

Note that

$$W = \begin{bmatrix} J_{n-1} & 0 \\ 0 & 1 \end{bmatrix} A_n \begin{bmatrix} J_{n-1} & 0 \\ 0 & 1 \end{bmatrix},$$

where

$$J_{n-1} = \begin{bmatrix} 0 & \ldots & 0 & 1 \\ 0 & \ldots & 1 & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & 0 & \ldots & 0 \end{bmatrix}. \tag{2.3}$$

This shows that W is positive definite if and only if A_n is positive definite. Theorem 1 implies the following algorithm for computing S-coefficients.

ALGORITHM 3

1. Compute the LDL^T factorization of W using the standard algorithm of Martin, Peters, and Wilkinson [14].
2. Set $c_k = -L_{n,k-1}$, $k = 1, \ldots, n$.

This algorithm requires $n^3/6 + O(n^2)$ multiplications and is, of course, slower than Algorithms 1 and 2. However, it can be used for an experimental comparison of numerical properties of Algorithms 1 and 2 with properties of Algorithm 3. It is interesting to note that

$$L_{n-1}D_{n-1}c = -[a_1, \ldots, a_n]^T,$$

and hence the vector of S-coefficients is an intermediate result in computing the solution of the Yule–Walker system of equations

$$A_{n-1}x = L_{n-1}D_{n-1}L_{n-1}^T x = -[a_1, \ldots, a_n]^T,$$

by the LDL^T algorithm.

3. Condition numbers. It follows from Algorithm 1 that c_k is a rational function of $a_0, \ldots, a_k$; therefore the map $F : a \to c$ is differentiable. Thus for $c \neq 0$, i.e., when A_n is different from the identity matrix, the formula (1.1) for the condition number $k(F, a)$ can be used, where the norm is always the infinity norm. Since it is clear that c_k depends on $a_0, \ldots, a_k$ only, we have

$$F'(a) = \left(\frac{\partial F_k}{\partial a_j}\right)_{k=1,j=0}^{n},$$

where $F_k : (a_0, \ldots, a_k) \to c_k, k = 1, \ldots, n$, and $\partial F_k/\partial a_j = 0$ for $k < j$.

THEOREM 2. *Let $E_k, q_k(j)$, $k = 0, \ldots, n$, $j = 0, \ldots, k$ be defined as in Algorithm 2. Then for $k = 1, \ldots, n$,*

$$\begin{bmatrix} 2\frac{\partial F_k}{\partial a_0} \\ \frac{\partial F_k}{\partial a_1} \\ \vdots \\ \frac{\partial F_k}{\partial a_k} \end{bmatrix} = -E_{k-1}^{-1}\left(\begin{bmatrix} q_k(k) & \ldots & q_k(1) & q_k(0) \\ 0 & q_k(k) & \ldots & q_k(1) \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & q_k(k) \end{bmatrix}\begin{bmatrix} 0 \\ q_{k-1}(0) \\ \vdots \\ q_{k-1}(k-1) \end{bmatrix}\right. \tag{3.1}$$
$$\left. + \begin{bmatrix} q_k(0) & \ldots & q_k(k-1) & q_k(k) \\ 0 & q_k(0) & \ldots & q_k(k-1) \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & q_k(0) \end{bmatrix}\begin{bmatrix} q_{k-1}(k-1) \\ \vdots \\ q_{k-1}(0) \\ 0 \end{bmatrix}\right)$$

and

$$k(F, a) = \frac{a_0}{\|c\|} \max_{k=1,\ldots,n} \sum_{j=0}^{k} \left|\frac{\partial F_k}{\partial a_j}\right|. \tag{3.2}$$

Proof. We will show (3.1) for $k = n$. For $k = 1, \ldots, n-1$, the proof is exactly the same. Let $\gamma_n(j) = E_n^{-1}q_n(j)$, $j = 0, 1, \ldots, n$ and $\gamma_{n-1}(j) = E_{n-1}^{-1}q_{n-1}(j)$, $j =$

$0, 1, \ldots, n-1$. It follows from (2.2) that $\gamma_n(j) = (A^{-1})_{n,j}$, $j = 0, \ldots, n$ and $\gamma_{n-1}(j) = (A_{n-1}^{-1})_{n-1,j}$, $j = 0, \ldots, n-1$, where $(A^{-1})_{r,s}$ denotes the (r, s) element of A^{-1}. From Algorithm 2, we have

$$c_n = -[\gamma_{n-1}(0)a_1 + \cdots + \gamma_{n-1}(n-1)a_n]. \tag{3.3}$$

Since $\gamma_{n-1}(j)$ do not depend on a_n, differentiation of (3.3) with respect to a_n yields

$$\frac{\partial F_n}{\partial a_n} = -\gamma_{n-1}(n-1) = -E_{n-1}^{-1}.$$

Differentiating (3.3) with respect to a_j, $j = 1, \ldots, n-1$, we get

$$\frac{\partial F_n}{\partial a_j} = -\sum_{m=0}^{n-1} a_{m+1} \frac{\partial \gamma_{n-1}(m)}{\partial a_j} - \gamma_{n-1}(j-1). \tag{3.4}$$

It is known (see, for example, Gohberg and Koltracht [5]) that for an arbitrary invertible matrix $A = (a_{i,j})_{i,j=0}^n$,

$$\frac{\partial (A^{-1})_{r,s}}{\partial a_{i,j}} = -e_r^T A^{-1} e_i e_j^T A^{-1} e_s.$$

Therefore, if A_n is Toeplitz and symmetric, then

$$\frac{\partial (A_n^{-1})_{r,s}}{\partial a_j} = -e_r^T A_n^{-1} T_j A_n^{-1} e_s,$$

where T_j is a Toeplitz matrix of the form

$$T_j = \begin{bmatrix} 0 & \ldots & 1 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & 0 & \ddots & 0 & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 1 & \ldots & 0 \end{bmatrix},$$

with 1's in positions of a_j's in A_n. Thus

$$\frac{\partial \gamma_{n-1}(m)}{\partial a_j} = \frac{\partial (A_{n-1}^{-1})_{n-1,m}}{\partial a_j}$$

$$= -e_{n-1}^T A_{n-1}^{-1} T_j A_{n-1}^{-1} e_m, \qquad m = 0, \ldots, n-1.$$

Substituting into (3.4), we get

$$\frac{\partial F_n}{\partial a_j} = -e_{n-1}^T A_{n-1}^{-1} T_j A_{n-1}^{-1} \sum_{m=0}^{n-1} a_{m+1} e_m - \gamma_{n-1}(j-1).$$

It is clear that

$$\sum_{m=0}^{n-1} a_{m+1} e_m = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}.$$

Since $A\gamma_n = e_n$, it is not hard to obtain now the well-known identity

$$A_{n-1}^{-1} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = - \begin{bmatrix} q_n(n-1) \\ \vdots \\ q_n(0) \end{bmatrix}.$$

Since

$$e_{n-1}^T A_{n-1}^{-1} = E_{n-1}^{-1}[q_{n-1}(0) \dots q_{n-1}(n-1)]^T,$$

we get for $j = 1, \dots, n-1$,

$$\frac{\partial F_n}{\partial a_j} = -E_{n-1}^{-1}(q_{n-1}^T T_j q_n + q_{n-1}(j-1)),$$

where

$$q_{n-1}^T = [q_{n-1}(0), \dots, q_{n-1}(n-1)].$$

For $j = 0$ this identity holds with $T_j = I$ and $q_{n-1}(-1) = 0$. It is straightforward to check that this system of equations is equivalent to (3.1). The formula (3.2) follows immediately from (1.1) and from the fact that $a_0 > |a_j|$, $j = 1, \dots, n$. The theorem is proved. □

The main reason for expressing $F'(a)$ in the form (3.1) is that in this form $F'(a)$ can be computed by the fast Fourier transform (FFT). Indeed, in this form the computation of F_k' reduces to two convolutions that can be obtained at the cost of $O(k \log k)$ arithmetic operations. Thus, we get the following recursive algorithm for computing the condition number of the map of S-coefficients.

ALGORITHM 4

1. Set $t_0 = 0, f_0 = 0$.
2. For $k = 1, \dots, n$
   2.1. Compute c_k, E_k and q_k via Algorithm 2.
   2.2. Compute $(\partial F_k / \partial a_j)_{j=0}^k$ via (3.1) using the FFT.
   2.3. Let

$$t_k = \max(|c_k|, t_{k-1}).$$

   2.4. Compute

$$f_k = \max\left(\sum_{j=0}^{k} \left| \frac{\partial F_k}{\partial a_j} \right|, f_{k-1} \right).$$

3. Set

$$k(F, a) = \frac{a_0 \cdot f_n}{t_n}.$$

The complexity of this algorithm is dominated by the step 2.2. The cost of step 2.2, for $k = 1, \dots, n$, is bounded from above by $O(n^2 \log n)$. Note that the condition number of an individual S-coefficient, $c_k \neq 0$, is given by

$$k(F_k, (a_1, \dots, a_k)) = \frac{a_0}{|c_k|} \sum_{j=0}^{k} \left| \frac{\partial F_k}{\partial a_j} \right|,$$

which can be computed at the cost of $O(k \log k)$ only.

It will be seen in the next section that the Schur algorithm is numerically more trustworthy than the Levinson algorithm for computing S-coefficients. Therefore, for better accuracy of the above condition estimator, it is advisable to compute the S-coefficients via the Schur algorithm first; and then to use them in the Levinson algorithm for computing vectors q_k, $k = 1, \ldots, n$.

Next, we derive upper and lower bounds for the condition number $k(F, a)$ in terms of S-coefficients, and obtain a precise and simple formula if all S-coefficients are nonnegative.

THEOREM 3. *Let $F : (a_0, \ldots, a_n) \to (c_1, \ldots, c_n)$ be the map of S-coefficients. Then the following bounds for the condition number $k(F, a)$ hold,*

$$\frac{1}{\|c\| \prod_{j=1}^{n-1}(1 - c_j^2)} \leq k(F, a) \leq \frac{2(1 + |c_n|)}{\|c\|} \prod_{j=1}^{n-1} \frac{1 + |c_j|}{1 - |c_j|}. \tag{3.5}$$

Moreover, if $c_j \geq 0$, $j = 1, \ldots, n$, then the condition number can be expressed as follows:

$$k(F, a) = \frac{(1 + c_n)}{\|c\|} \prod_{j=1}^{n-1} \frac{1 + c_j}{1 - c_j}. \tag{3.6}$$

Proof. Since

$$\frac{\partial F_n}{\partial a_n} = E_{n-1}^{-1} = \frac{1}{a_0 \prod_{j=1}^{n-1}(1 - c_j^2)},$$

the left inequality follows immediately from (3.2). From (3.1) we have

$$\|F'(a)\| \leq 2 \max_{k=1,\ldots,n} (E_{k-1}^{-1} \|q_k\|_1 \|q_{k-1}\|_1).$$

It is shown by Cybenko [4] that

$$\|q_k\|_1 \leq \prod_{j=1}^{k} (1 + |c_j|) \leq \prod_{j=1}^{n} (1 + |c_j|);$$

hence,

$$\|F'(a)\| \leq 2 \left(E_{n-1}^{-1} \prod_{j=1}^{n-1} (1 + |c_j|) \prod_{j=1}^{n} (1 + |c_j|) \right) = \frac{2(1 + |c_n|)}{\|c\|} \prod_{j=1}^{n-1} \frac{1 + |c_j|}{1 - |c_j|}.$$

Thus the right inequality follows. Suppose now that $c_1, \ldots, c_n \geq 0$. Then it follows from Algorithm 2 that $q_k(j) \geq 0$, $k = 0, \ldots, n$, $j = 0, \ldots, k$. Since all $\partial F_k / \partial a_j$ are of the same sign, it follows from (3.1) that

$$\begin{aligned} 2\left(\frac{\partial F_k}{\partial a_0}\right) + \sum_{j=1}^{k} \left(\frac{\partial F_k}{\partial a_j}\right) &= -E_{k-1}^{-1} \left(\sum_{j=0}^{k} q_k(j) \right) \left(\sum_{j=0}^{k-1} q_{k-1}(j) \right) + \frac{\partial F_k}{\partial a_0} \\ &= -E_{k-1}^{-1} \|q_k\|_1 \|q_{k-1}\|_1 + \frac{\partial F_k}{\partial a_0}. \end{aligned}$$

It is also shown by Cybenko [4] that in this case

$$\|q_k\|_1 = \sum_{j=0}^{k} q_k(j) = \prod_{j=1}^{k}(1+c_j).$$

Thus

$$\|F'(a)\| = \frac{(1+c_n)}{a_0}\prod_{j=1}^{n-1}\frac{1+c_j}{1-c_j}.$$

The theorem is proved. □

For the discussion of numerical properties of Algorithm 3 in the next section, we remark that bounds of Theorem 3 are similar to the bounds for the usual condition number of A_{n-1}, $k(A_{n-1}) = \|A_{n-1}^{-1}\|\|A_{n-1}\|$. Indeed, it is shown by Cybenko [4] that

$$\frac{1}{\prod_{j=1}^{n-1}(1-c_j^2)} \le k(A_{n-1}) \le n\prod_{j=1}^{n-1}\frac{1+|c_j|}{1-|c_j|}. \tag{3.7}$$

In particular, if $c_1, \ldots, c_n \ge 0$, then clearly

$$k(A_{n-1}) \le nk(F,a). \tag{3.8}$$

The comparison of bounds in (3.5) and (3.7) and the inequality (3.8) indicate that there should be little difference between the condition numbers $k(A_{n-1})$ and $k(F,a)$. Our numerical experiments in §5 support this conclusion.

4. Analysis of algorithms. To analyze the propagation of roundoff errors in Algorithms 1 and 2, we need the bounds for the size of entries of vectors p_k and q_k appearing in these algorithms, which will be used later in the Appendix. For the quantities q_k in Algorithm 2 we have the bound already used in Theorem 3; namely,

$$\sum_{j=0}^{k}|q_k(j)| \le \prod_{j=1}^{k}(1+|c_j|),\ k=1,\ldots,n, \tag{4.1}$$

with the equality holding in the case when $c_1, \ldots, c_n \ge 0$. Quantities $q_k(k+1), \ldots, q_k(n)$ in Algorithm 1 can be bounded from above using (2.1). (See the proof of Theorem 4.) To find an upper bound for $p_k(k+1), \ldots, p_k(n)$, we need the following result.

LEMMA 1. *Let* $W_k = Z_k A_n Z_k$, $k = 1, \ldots, n-1$, *where*

$$Z_k = \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix}$$

is a permutation matrix with the leading block J *defined in* (2.3) *and of the size* $(k+1)\times(k+1)$. *Let* $W_k = L_k D_k L_k^T$ *be the* LDL^T *factorization of* W_k; *then*

$$E_j = (D_k)_{j,j}, \qquad j = 0, \ldots, n$$

and

$$p_k(j) = -E_k(L_k)_{j,k}, \qquad j = k+1, \ldots, n.$$

Proof. Let k be fixed. Let J_m be of the form (2.3) and have the size $(m+1)\times(m+1)$. Using the fact that $J_m A_m J_m = A_m$, $m = 0,1,\ldots,n$, it is easy to see that the determinants of the principal leading submatrices of A_n and W_k of the same size are equal. Since diagonal entries in the LDL^T factorization are ratios of corresponding determinants, it follows that they are the same for A_n and W_k; hence, $(D_k)_{m,m} = E_m$, $m = 0,\ldots,n$.

It is shown in Gohberg et al. [6] that for the matrix

$$A_k^c = \{a_{|i-j|}\}_{j=1,\ldots,k}^{i=k+1,\ldots,n},$$

the following identity holds:

$$A_k^c \begin{bmatrix} q_k(k) \\ \vdots \\ q_k(0) \end{bmatrix} = - \begin{bmatrix} p_k(k+1) \\ \vdots \\ p_k(n) \end{bmatrix}. \tag{4.2}$$

Observing that

$$W_k = \begin{bmatrix} A_k & * \\ A_k^c J_k & * \end{bmatrix}$$

and comparing the factorization identities (2.1) and $W_k = L_k D_k L_k^T$, it is not hard to see that

$$W_k \begin{bmatrix} q_k(0) \\ \vdots \\ q_k(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = E_k \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ (L_k)_{k+1,k} \\ \vdots \\ (L_k)_{n,k} \end{bmatrix}.$$

Thus,

$$A_k^c J_k \begin{bmatrix} q_k(0) \\ \vdots \\ q_k(k) \end{bmatrix} = E_k \begin{bmatrix} (L_k)_{k+1,k} \\ \vdots \\ (L_k)_{n,k} \end{bmatrix}.$$

Comparing the last equality with (4.2) we get

$$p_k(j) = -E_k(L_k)_{j,k}, \qquad j = k+1,\ldots,n.$$

The lemma is proved. □

THEOREM 4. *Let $E_k, q_k(j), p_k(j)$, $k = 1,\ldots,n-1$, $j = k+1,\ldots,n$ be defined by Algorithm* 1. *Then*

$$|q_k(j)|,\ |p_k(j)| \le (E_k(a_0 - E_j))^{1/2} \le a_0. \tag{4.3}$$

Proof. It follows from (2.1) that

$$\sum_{k=0}^{j-1} q_k^2(j) E_{k-1}^{-1} + E_j = a_0;$$

hence, $q_k^2(j) \le E_k(a_0 - E_j)$. Similarly from $W_k = L_k D_k L_k^T$, we have

$$\sum_{i=0}^{j-1} (L_k)_{j,i}^2 (D_k)_{i,i} + (D_k)_{j,j} = a_0.$$

By Lemma 1 we have $p_k(j) = -E_k(L_k)_{j,k}$ and $(D_k)_{i,i} = E_i$; hence, $p_k^2(j)E_k^{-1} \le (a_0 - E_j)$. The theorem is proved. □

Remark. The first part of this theorem is well known; to the best of our knowledge the second part, i.e., the bound for $p_k(j)$, is new.

Next, we state the results of the backward round off error analysis for Algorithms 1 and 2. The proofs, which are rather technical, are given in the Appendix. We assume the standard model of floating point arithmetic with a guard digit

$$fl(x \otimes y) = (x \otimes y)(1 + \epsilon),$$

where $\otimes$ stands for $+, -, *, /$, and ϵ is bounded by the unit roundoff error $|\epsilon| \le u$. We also assume for simplicity that $a_0 = 1$.

THEOREM 5. *Let $\bar{c}_k, \bar{p}_k(j), \bar{E}_k, \bar{q}_k(j)$ be the values of $c_k, p_k(j), E_k, q_k(j)$ computed by Algorithm 1 in floating point arithmetic with a unit roundoff error u. Suppose that $|\bar{c}_k|, |\bar{p}_k(j)|, |\bar{q}_k(j)| < 1$. Then there exist $\epsilon_1, \dots, \epsilon_n$ such that $\bar{c}_1, \dots, \bar{c}_n$ are the exact S-coefficients of the positive definite Toeplitz matrix defined by $a_0 = 1, a_1 + \epsilon_1, \dots, a_n + \epsilon_n$ and*

$$|\epsilon_i| \le 4i \left(2 + \sum_{j=1}^{i} |\bar{c}_j| \prod_{t=1}^{j-1} (1 + |\bar{c}_t|) \right) u + o(u), \qquad i = 1, \dots, n.$$

Note that $\sum_{j=1}^{i} |\bar{c}_j| \le i$; hence, we also have

$$|\epsilon_i| \le 4i \left(2 + i \prod_{t=1}^{i-1} (1 + |\bar{c}_t|) \right) u + o(u), \qquad i = 1, \dots, n. \tag{4.4}$$

THEOREM 6. *Let $\bar{c}_k, \bar{E}_k, \bar{q}_k(j)$ be the values of $c_k, E_k, q_k(j)$ computed by Algorithm 2 in floating point arithmetic with a unit roundoff error u. Suppose that $|\bar{c}_k| < 1$, $k = 1, \dots, n$. Then there exist $\epsilon_1, \dots, \epsilon_n$ such that $\bar{c}_1, \dots, \bar{c}_n$ are the exact S-coefficients of the positive definite Toeplitz matrix defined by $a_0 = 1, a_1 + \epsilon_1, \dots, a_n + \epsilon_n$ and*

$$|\epsilon_i| \le 2i^2 \prod_{j=1}^{i-1} (1 + |\bar{c}_j|)^2 u + o(u), \qquad i = 1, \dots, n. \tag{4.5}$$

The assumption made in Theorems 5 and 6 that $|\bar{c}_k| < 1$, $k = 1, \dots, n$ is necessary because if for some k, $|\bar{c}_k| \ge 1$, then there is no positive definite Toeplitz matrix having $\bar{c}_k$ as its S-coefficients. In view of Theorem 4, the same applies to the assumption in Theorem 5 that $|\bar{p}_k(j)|$, $|\bar{q}_k(j)| \le 1$.

Moreover, if say, $(a_1, \dots, a_n)$ represents the response of a layered piecewise homogeneous elastic medium to a unit probing signal applied to the surface, then $c_1, \dots, c_n$ are the reflection coefficients of this stratified medium, and by definition they must be less than one in magnitude. The quantities $p_k(j)$ and $q_k(j)$ represent observed signals in the kth layer and also must be less than one in magnitude. (See, for example,

Koltracht and Lancaster [11].) Therefore, the assumptions of Theorems 5 and 6 also mean that the computed quantities are physically meaningful.

Theorems 5 and 6 lead us to following observations.

Observation 1. The fact that the essential factor of the backward error bound for the Schur algorithm, namely, $\prod_{j=1}^{n-1}(1+|\bar{c}_j|)$, is squared in the error bound for the Levinson algorithm, indicates that the Schur algorithm might be more trustworthy. This conclusion is supported by numerical evidence in the next section.

Observation 2. The error bound (4.4) has the following practical significance: If the S-coefficients computed by the Schur algorithm satisfy the inequality

$$4n^2 \prod_{j=1}^{n-1}(1+|\bar{c}_j|)k(F,a)u \leq 10^{-p}, \tag{4.6}$$

and the conditions of Theorem 5 are met, then at least p correct figures can be guaranteed in the computed S-coefficients.

For S-coefficients computed by Algorithm 3, the following inequality holds:

$$\frac{\|\bar{c}-c\|}{\|c\|} \leq Kn^{3/2}k(A_{n-1})u + o(u), \tag{4.7}$$

where K is a constant of order unity, and where in practice it is rare for the bound $Kn^{3/4}k(A_{n-1})u$ to be exceeded. This follows immediately from the results of Martin, Peters, and Wilkinson [14], because the computation of $\bar{c}_1, \ldots, \bar{c}_n$ amounts to the following:

1. Computing L_{n-1} and D_{n-1} by the algorithm from Martin, Peters, and Wilkinson [14].

2. Solving by substitution

$$L_{n-1}\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}.$$

3. Setting

$$c_i = -(D_{n-1})_{i,i}^{-1} y_i, \qquad i = 1, \ldots, n.$$

Following Gohberg and Koltracht [5], we could claim that Algorithm 3 is forward stable for computing S-coefficients of a positive definite Toeplitz matrix, if we could show that $k(A_{n-1})$ is of the same order of magnitude as $k(F,a)$, the condition number of the map of S-coefficients. It follows from Theorem 3 and (3.7) that both upper and lower bounds for $k(F,a)$ are comparable with those for $k(A_{n-1})$. In the case of $c_1, \ldots, c_n \geq 0$, it follows from (3.8) that we can replace $k(A_{n-1})$ by $nk(F,a)$ in (4.7). Extensive numerical evidence in the following section shows that there is indeed little difference between the two condition numbers. Therefore we claim that Algorithm 3 is forward stable for computing S-coefficients.

5. Numerical experiments. Numerical experiments in this section are designed to test the forward stability of Algorithms 1, 2, and 3 and to compare the condition number of the map of S-coefficients $k(F,a)$, with the usual condition number of A_{n-1}, $k(A_{n-1}) = \|A_{n-1}^{-1}\|\|A_{n-1}\|$. In each experiment, we start with a set of S-coefficients $c_1, \ldots, c_n$ chosen in $(-1,1)$ and generate the corresponding positive

definite Toeplitz matrix A_n, using an algorithm from Koltracht and Lancaster [10], in double precision. Then we compute S-coefficients of this matrix A_n using Algorithms 1, 2, and 3 in single precision and compare the computed S-coefficients $\bar{c}_1, \ldots, \bar{c}_n$ with the original set. The condition numbers $k(F, a)$ and $k(A_{n-1})$ are computed in double precision via Algorithm 4 and the Gohberg–Semencul formula (see, for example, Gohberg and Leiterer [8]), respectively. All computations are performed on the IBM ES-9000 computer with the unit roundoff error $u \approx 10^{-7}$.

For a fixed value of n (e.g., $n = 20, 30, 40$), we generate n uniformly distributed random numbers in a given subinterval of $(-1, 1)$ as the original S-coefficients. We recompute the S-coefficients by agorithms 1, 2, and 3 and denote them by $\bar{c}^i, i = 1, 2, 3$, respectively. This procedure is repeated 20 times for every given subinterval. In the following table the worst relative error

$$\frac{\|\bar{c}^i - c\|}{\|c\|}$$

among the 20 trials is given for each algorithm and for different subintervals. For each subinterval we also compute the largest value of the condition number, $K(F, a)$, over the 20 trials and the largest ratio of $k(A_{n-1}) \equiv k_1$ versus $k(F, a) \equiv k_2$, and k_2 versus k_1.

TABLE 1
$n = 20$.

Sign	Subin. of c	Worst case errors			$k(F, a)$	k_1/k_2	k_2/k_1
		Levinson	Schur	LDL^T			
−	[−.25, 0]	.00007	.00004	.00002	135.18	1.23	4.31
	[−.35, 0]	.00031	.00015	.00007	8×10^2	1.77	3.59
	[−.45, 0]	.00242	.001278	.00055	6×10^3	1.97	3.56
+/−	[−.25, .25]	.00002	.00002	.00001	59.88	1.20	2.87
	[−.45, .45]	.00010	.00010	.00004	4×10^2	2.17	1.78
	[−.65, .65]	.00127	.00130	.00045	9×10^3	4.13	1.36
+	[0,.25]	.00005	.00013	.00018	2×10^3	0.22	10.1
	[0,.35]	.00053	.00060	.00067	8×10^4	0.36	8.96
	[0,.45]	.0176	.0157	.01349	2×10^5	0.55	7.03

TABLE 2
$n = 30$.

Sign	Subin. of c	Worst case errors			$k(F, a)$	k_1/k_2	k_2/k_1
		Levinson	Schur	LDL^T			
−	[−.25, 0]	.00079	.00034	.00014	1689	0.85	2.76
	[−.35, 0]	.022	.007	.009	1×10^4	1.14	2.21
	[−.40, 0]	.124	.039	.020	5×10^5	1.32	1.97
+/−	[−.25, .25]	.000029	.000025	.000012	106	1.32	2.66
	[−.45, .45]	.00030	.00023	.0001	1×10^3	3.0	1.56
	[−.65, .65]	.01578	.01114	.00615	1×10^5	6.71	0.86
+	[0,.25]	.00054	.00068	.0020	3×10^4	.25	8.7
	[0,.35]	.050	.056	.073	9×10^5	0.35	5.98
	[0,.40]	.387	.193	.272	5×10^6	0.43	5.01

TABLE 3
$n = 40$.

Sign	Subin. of c	worst case errors			$k(F,a)$	k_1/k_2	k_2/k_1
		Levinson	Schur	LDL^T			
−	$[-.20, 0]$	.00173	.00081	.00037	2×10^3	0.586	4.45
	$[-.25, 0]$	.00692	.00312	.00121	1×10^4	0.77	3.52
	$[-.30, 0]$	.059	.021	.0074	8×10^4	0.85	2.80
+/−	$[-.25, .25]$	.00011	.00011	.00006	482.91	1.63	2.52
	$[-.45, .45]$	.00362	.00311	.00098	3×10^4	4.30	1.14
	$[-.55, .55]$	.061	.062	.014	4×10^5	5.52	1.07
+	$[0, .20]$	.00287	.00462	.0086	3×10^5	0.16	11.5
	$[0, .25]$	.01195	.01576	.04223	3×10^5	0.21	9.44
	$[0, .30]$	.5315	.5637	.4819	6×10^6	0.31	6.54

We see from Tables 1, 2, and 3 that the ratio k_1/k_2 is, indeed, of order unity and that the forward error in all these algorithms is consistent with the condition of the problem; namely,

$$\frac{\|\bar{c}^i - c\|}{\|c\|} \approx k(F,a)u.$$

Note that in most of the cases the worst error in the LDL^T algorithm is smaller than that in the Schur algorithm which is in turn smaller than that in the Levinson algorithm. For matrices that correspond to nonnegative S-coefficients, the worst error is smallest for the Levinson algorithm. However, a closer examination of each of the 20 cases represented by one row of a table (even for rows corresponding to nonnegative S-coefficients), shows that the error in Levinson algorithm is larger than that in Schur algorithm which is in turn larger than that in the LDL^T algorithm in the vast majority of the individual experiments. Therefore, our practical recommendation is to use the Schur algorithm (or the LDL^T algorithm if the cost is no object) for computing S-coefficients. Note also that condition numbers are visibly smaller when S-coefficients vary in sign as compared to the constant sign pattern, that agrees with the results obtained in Koltracht and Lancaster [10]. When the intervals are larger, like $(-.85, .85)$, or $(-.65, 0)$, or when n becomes larger, all three algorithms fail in the worst case because the condition numbers become too large.

Appendix. It is convenient to start with the proof of Theorem 6 which is somewhat easier.

Proof of Theorem 6. It is clear that for $k = 1$ the theorem is true. Let us consider q_k, c_k, E_k as functions of $a_1, \ldots, a_k$ and denote them as $c_k = c_k(a_1, \ldots, a_k), q_k = q_k(a_1, \ldots, a_k)$ and $E_k = E_k(a_1, \ldots, a_k)$. Let us denote $T_k \equiv (a_1, \ldots, a_k)$. Assume that the claim is true for $s = 1, \ldots, k$. For ϵ_s, $s = 1, 2, \ldots, k$, define

$$q_s^{'} = q_s(a_1 + \epsilon_1, \ldots, a_s + \epsilon_s),$$

$$E_s^{'} = E_s(a_1 + \epsilon_1, \ldots, a_s + \epsilon_s).$$

Let us now find ϵ_{k+1} such that $\bar{c}_{k+1} = c_{k+1}(\rho_1 + \epsilon_1, \ldots, \rho_{k+1} + \epsilon_{k+1})$, where ϵ_{k+1} satisfies (4.5). It follows from Algorithm 2 that $\bar{c}_{k+1} = fl(\bar{s}_{k+1} * \bar{E}_k^{-1})$, which can be written as follows.

$$\bar{c}_{k+1} = -(T_k \cdot \bar{q}_k + \Delta_k)(1 + \delta_k)/\bar{E}_k, \tag{A.1}$$

where $|\delta_k| \le u, |\Delta_k| \le |T_k| \cdot |\bar{q_k}|u$, and $a \cdot b$ denotes the dot product of two vectors. Let us define ϵ_{k+1} from the equation:

$$\bar{c}_{k+1} = -\left(a_{k+1} + \epsilon_{k+1} + \sum_{i=0}^{k-1} a_{i+1} q_k^{'}(i)\right) / E_k^{'}. \tag{A.2}$$

It follows from (6.1) that

$$(T_k \cdot \bar{q_k} + \Delta_k)(1 + \delta_k) E_k^{'} = \left(a_{k+1} + \epsilon_{k+1} + \sum_{i=0}^{k-1} a_{i+1} q_k^{'}(i)\right) \bar{E}_k.$$

Solving for ϵ_{k+1}, we get

$$\begin{aligned}\epsilon_{k+1} = {} & (E_k^{'}/\bar{E}_k - 1)(T_k \cdot \bar{q_k} + \Delta_k) + \sum_{i=0}^{k} a_{i+1}(\bar{q_k}(i) - q_k^{'}(i)) \\ & + E_k^{'}/\bar{E}_k \, (T_k \cdot \bar{q_k} + \Delta_k) \, \delta_k + \Delta_k.\end{aligned}$$

Since $\bar{q_k}(k) = q_k^{'}(k) = 1$, we have

$$\begin{aligned}\epsilon_{k+1} = {} & (E_k^{'}/\bar{E}_k - 1)(T_k \cdot \bar{q_k} + \Delta_k) + \sum_{i=0}^{k-1} a_{i+1}(\bar{q_k}(i) - q_k^{'}(i)) \\ & + E_k^{'}/\bar{E}_k \, T_k \cdot \bar{q_k} \, \delta_k + \Delta_k + o(u).\end{aligned} \tag{A.3}$$

Assuming that $(1 - c_k) * (1 + c_k)$ is used to compute $1 - c_k^2$, we get $\bar{E}_k = \bar{E}_{k-1}(1 - \bar{c_k}^2)/(1 + \gamma_k)$ where $\gamma_k \le 4u + o(u)$. Therefore, $E_k^{'}/\bar{E}_k = \prod_{i=1}^{k}(1 + \gamma_i)$ or

$$|E_k^{'}/\bar{E}_k - 1| \le 4ku + o(u). \tag{A.4}$$

Let us now find the bound for $\|\bar{q_k} - q_k^{'}\|_1$. From the definition, we have

$$\bar{q_k} = \begin{bmatrix} 0 \\ \bar{q}_{k-1} \end{bmatrix} + \bar{c}_k \begin{bmatrix} J_{k-1}\bar{q}_{k-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \eta_k \\ 0 \end{bmatrix},$$

where η_k is a (k+1)-dimensional vector and

$$|\eta_k(i)| \le (|\bar{q}_{k-1}(i)| + |\bar{c}_k||\bar{q}_{k-1}(i-1)|)2u.$$

From (4.1), we get

$$|\eta_k(i)| \le (1 + |\bar{c}_k|)\|\bar{q}_{k-1}\|_1 2u \le \prod_{j=1}^{k}(1 + |\bar{c}_j|)2u. \tag{A.5}$$

Also from the definition, it follows that

$$q_k^{'} = \begin{bmatrix} 0 \\ q_{k-1}^{'} \end{bmatrix} + \bar{c}_k \begin{bmatrix} J_{k-1} q_{k-1}^{'} \\ 0 \end{bmatrix}.$$

Thus for $s = 1, \ldots, k$

$$\|\bar{q}_s - q_s^{'}\|_1 \le (1 + |\bar{c}_s|)\|\bar{q}_{s-1} - q_{s-1}^{'}\|_1 + \|\eta_s\|_1.$$

Starting with $\eta_0 = 0$, we get by induction and (A.5),

$$\begin{aligned}\|\bar{q}_s - \bar{q}_s^{'}\|_1 &\leq \sum_{i=1}^{s-1} \prod_{j=i+1}^{s} (1 + |\bar{c}_j|)\|\eta_i\|_1 \\ &\leq \sum_{i=1}^{s-1} \prod_{j=i+1}^{s} (1 + |\bar{c}_j|)k \prod_{j=1}^{i}(1 + |\bar{c}_j|)2u,\end{aligned}$$

for $s = 1, \ldots, k$, which implies that

$$\|\bar{q}_k - \bar{q}_k^{'}\|_1 \leq k^2 \prod_{i=1}^{k}(1 + |\bar{c}_i|)^2 2u. \tag{A.6}$$

Using (A.4) and (A.6), we obtain from (A.3) that

$$|\epsilon_{k+1}| \leq 4ku \prod_{i=1}^{k}(1 + |\bar{c}_i|) + k^2 2u \prod_{i=1}^{k}(1 + |\bar{c}_i|)^2 + u \prod_{i=1}^{k}(1 + |\bar{c}_i|)^2 + o(u).$$

The theorem is proved. □

Proof of Theorem 5. The claim is clearly true for $k = 1$. Suppose that (4.4) holds for $s = 1, \ldots, k$. For these ϵ_s, $s = 1, \ldots, k$, define

$$p_s^{'} = p_s(\rho_1 + \epsilon_1, \ldots, \rho_s + \epsilon_s),$$

$$q_s^{'} = q_s(\rho_1 + \epsilon_1, \ldots, \rho_s + \epsilon_s),$$

$$E_s^{'} = E_s(\rho_1 + \epsilon_1, \ldots, \rho_s + \epsilon_s).$$

Let us find now ϵ_{k+1} which satisfies (4.4). It follows from Algorithm 1 that

$$\bar{c}_{k+1} = fl(\bar{p}_k(k+1) * \bar{E}_k^{-1}) = (\bar{p}_k(k+1) * \bar{E}_k^{-1})(1 + \eta_k),$$

where $|\eta_k| \leq u$. We also have

$$\bar{p}_k(k+1) = \bar{p}_{k-1}(k+1) + \bar{c}_k \bar{q}_{k-1}(k) + \delta_{k-1};$$

hence,

$$\bar{p}_k(k+1) = p_0(k+1) + \sum_{i=1}^{k}(\bar{c}_i \bar{q}_{i-1}(k) + \delta_{i-1}), \tag{A.7}$$

where for $i = 0, 1, \ldots, k-1$,

$$|\delta_i| \leq |\bar{p}_i(k+1)|u + |\bar{c}_{i+1}||\bar{q}_i(k)|2u \leq (1 + |\bar{c}_{i+1}|)2u. \tag{A.8}$$

Let us define now ϵ_{k+1} from the equation

$$\bar{c}_{k+1} = \left[p_0(k+1) - \epsilon_{k+1} + \sum_{i=1}^{k} \bar{c}_i q_{i-1}^{'}(k)\right] / E_k^{'}. \tag{A.9}$$

Then it follows from (A.7) and (A.8) that

$$\left[p_0(k+1)+\sum_{i=1}^{k}(\bar{c}_i\bar{q}_{i-1}(k)+\delta_{i-1})\right]E_k^{'}(1+\eta_k)=$$

$$\left[p_0(k+1)-\epsilon_{k+1}+\sum_{i=1}^{k}\bar{c}_i q_{i-1}^{'}(k)\right]\cdot\bar{E}_k.$$

Solving for ϵ_{k+1}, we get

$$\epsilon_{k+1}=\bar{p}_k(k+1)(1-E_k^{'}/\bar{E}_k)+\left(\bar{p}_k(k+1)\eta_k+\sum_{i=1}^{k}\delta_{i-1}\right)$$

$$+\sum_{i=1}^{k}\bar{c}_i(q_{i-1}^{'}(k)-\bar{q}_{i-1}(k)). \tag{A.10}$$

As in the proof of Theorem 6, we have

$$|E_k^{'}/\bar{E}_k-1|\leq 4ku+o(u). \tag{A.11}$$

It remains to find the bound of $q_{i-1}^{'}(k)-\bar{q}_{i-1}(k)$. It follows from Algorithm 1 that

$$\bar{q}_i(j)=\bar{p}_{i-1}(j)\bar{c}_i+\bar{q}_{i-1}(j-1)+\delta_j^i \tag{A.12}$$

and

$$\bar{p}_i(j)=\bar{p}_{i-1}(j)+\bar{c}_i\bar{q}_{i-1}(j-1)+\Delta_j^i, \tag{A.13}$$

where for $i=1,\ldots,k$ and $j=i+1,\ldots,n$,

$$|\delta_j^i|\leq(1+|\bar{c}_i|)2u, \tag{A.14}$$

$$|\Delta_j^i|\leq(1+|\bar{c}_i|)2u. \tag{A.15}$$

Let us define

$$d_i=[\bar{q}_i(i+1),\ldots,\bar{q}_i(n)]^T-[q_i^{'}(i+1),\ldots,q_i^{'}(n)]^T,$$

$$D_i=[\bar{p}_i(i+1),\ldots,\bar{p}_i(n)]^T-[p_i^{'}(i+1),\ldots,p_i^{'}(n)]^T,$$

and

$$\lambda_i=\max(\|d_i\|_\infty,\|D_i\|_\infty),$$

$$\mu_i=\max_{i\leq j\leq n}(|\delta_j^i|,|\Delta_j^i|).$$

Then it follows from (A.12), (A.13) and the definition of $p_i^{'}, q_i^{'}$ that

$$\|d_i\|_\infty\leq(1+|\bar{c}_i|)\lambda_{i-1}+\mu_i,\|D_i\|_\infty\leq(1+|\bar{c}_i|)\lambda_{i-1}+\mu_i;$$

hence, $\lambda_i \leq (1 + |\bar{c}_i|)\lambda_{i-1} + \mu_i$. Since $\lambda_0 = 0$, we obtain

$$\lambda_i \leq \sum_{t=2}^{i+1} \prod_{s=t}^{i} (1 + |\bar{c}_s|)\mu_{t-1}; \tag{A.16}$$

hence, for $i = 1, \ldots, k$,

$$\lambda_i \leq i \prod_{j=1}^{i} (1 + |\bar{c}_j|)2u + o(u). \tag{A.17}$$

Returning to (6.10), we see that

$$\begin{aligned} |\epsilon_{k+1}| &\leq 4ku + u + \sum_{i=1}^{k} (1 + |\bar{c}_i|)2u \\ &\quad + \sum_{i=1}^{k} |\bar{c}_i| i \prod_{j=1}^{k} (1 + |\bar{c}_j|)2u + o(u) \qquad \text{(A.18)} \\ &\leq 4k \left(2 + \sum_{i=1}^{k} |\bar{c}_i| \prod_{j=1}^{k} (1 + |\bar{c}_j|) \right) u + o(u). \qquad \text{(A.19)} \end{aligned}$$

The theorem is proved. □

Acknowledgments. We would like to thank Professor James Bunch and the referees for their useful comments.

REFERENCES

[1] A. M. BRUCKSTEIN AND T. KAILATH, *Inverse scattering for discrete transmission-line models*, SIAM Rev., 29 (1987), pp. 359–389.

[2] A. BULTHEEL, *Error analysis of incoming and outgoing schemes for the trigonometric moment problem*, in Padé Approximation and its Applications, Amsterdam, 1980; Lecture Notes in Math. 888, Springer-Verlag, Berlin, 1981, pp. 100–109,

[3] I. F. CLAERBOUT, *Imaging the Earth Interior*, Blackwell Scientific Publications, Inc., Cambridge, MA, 1985.

[4] G. CYBENKO, *The numerical stability of the Levinson–Durbin algorithm for Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 3 (1980), pp. 303–319.

[5] I. GOHBERG AND I. KOLTRACHT, *Mixed componentwise, and structured condition numbers*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 688–704.

[6] I. GOHBERG, T. KAILATH, I. KOLTRACHT, AND P. LANCASTER, *Linear complexity parallel algorithms for linear systems of equations with recursive structure*, Linear Algebra Applications, 88/89 (1957), pp. 271–315.

[7] I. GOHBERG, I. KOLTRACHT, AND D. XIAO, *On the solution of the Yule–Walker equations*, Proc. SPIE Conference On Advanced Algorithms and Architectures for Signal processing IV, Vol. 1566, July 1991, pp. 14–22.

[8] I. GOHBERG AND J. LEITERER, *General theorems on canonical factorization of operator functions relative to a contour*, MAT. Issled. 3 (1972), pp. 87–134. (In Russian.)

[9] T. KAILATH, *A Theorem of I. Schur and its impact on modern signal processing*, in I. Schur Methods in Operator Theory and Signal Processing, Oper. Theory, Adv. Appl., I. Gohberg, ed., Vol. 18, Birkhauser, Basel, 1986, pp. 9–30.

[10] I. KOLTRACHT AND P. LANCASTER, *Condition numbers of Toeplitz and block-Toeplitz matrices*, Oper. Theory, Advances Applications, 18 (1986), pp. 231–323.

[11] ———, *Threshold algorithms for the prediction of reflection coefficients in a layered medium*, Geophysics, 53 (1987), pp. 908–919.

[12] N. LEVINSON, *The Wiener* RMS *(root mean square) error criterion in filter design and prediction*, J. Math. Phys., 25 (1947), pp. 261–278.

[13] J. D. MARKEL AND A. H. GRAY, JR., *Linear Prediction of Speech*, Springer-Verlag, New York, 1978.

[14] R. S. MARTIN, G. PETERS, AND J. H. WILKINSON, *Symmetric decomposition of a positive definite matrix*, Numer. Math., 7 (1965), pp. 362–383.

[15] F. RAMSEY, *Characterisation of the partial autocorrelation function*, Ann. Stat., 2 (1974), pp. 1296–1301.

[16] I. SCHUR, *Ueber Potenzreihen die im Inneren des Einheitskreises Beschrankt Sind*, J. Reine Angew. Math., 147 (1917), pp. 205–232.

[17] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

[18] S. T. ALEXANDER AND Z. M. RHEE, *Analytical finite precision results for Burg's algorithm and the autocorrelation method for linear prediction*, IEEE Trans. ASSP-35 (1987), pp. 626–635.

[19] C. P. RIALAN AND L. L. SCHARF, *Fixed-point error analysis of the lattice and the Schur algorithms for the autocorrelation method of linear prediction*, IEEE Trans. ASSP-37 (1989), pp. 1950–1987.

[20] G. R. BELITSKII AND Y. I. LYUBICH, *Matrix Norms and their Applications*, OT36, Birkhauser, Basel, 1989.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1310–1318, October 1994

017

EIGENVALUES OF BLOCK MATRICES ARISING FROM PROBLEMS IN FLUID MECHANICS*

K. A. CLIFFE†, T. J. GARRATT‡, AND A. SPENCE§

Abstract. Block matrices with a special structure arise from mixed finite element discretizations of incompressible flow problems. This paper is concerned with an analysis of the eigenvalue problem for such matrices and the derivation of two shifted eigenvalue problems that are more suited to numerical solution by iterative algorithms like simultaneous iteration and Arnoldi's method. The application of the shifted eigenvalue problems to the determination of the eigenvalue of smallest real part is discussed and a numerical example arising from a stability analysis of double-diffusive convection is described.

Key words. block matrices, eigenvalues, finite elements, Navier–Stokes

AMS subject classifications. 15A18, 65F15, 65F50, 76M10

1. Introduction.

Let A and B be $N \times N$ real matrices with the block structure

$$A = \begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}, \tag{1.1}$$

where $N = n+m$, $n > m$, K is $n \times n$, C is $n \times m$ of rank m, and M is $n \times n$ symmetric positive definite. The paper is concerned with the theory of the generalised eigenvalue problem

$$Aw = \mu Bw \tag{1.2}$$

called EVP1 and three related eigenvalue problems called EVP2, EVP3, and EVP4 that are introduced in §3. Since the matrices are typically large and sparse, numerical techniques based on transformation methods like the QZ algorithm will be very expensive. The reason for introducing the related eigenvalue problems EVP2, EVP3, and EVP4 is that they should be amenable to iterative techniques commonly used to find selected eigenvalues of large sparse matrices ([2, §I-5]). In the applications we have in mind K, C, and M arise from mixed finite element discretizations of the "velocity-pressure" formulation of the Navier–Stokes equations for incompressible flow problems [3], [9], and the eigenvalue problem (1.2) arises in the determination of the stability of steady flows [5]. The problem is to find the eigenvalues of (1.2) with smallest real part.

As is standard, the finite values $\mu \in C$ such that $\det(A - \mu B) = 0$ are known as finite eigenvalues, though we usually drop the term "finite." Since B is singular there are also infinite eigenvalues, which are defined to be zero eigenvalues of $\nu Aw = Bw$, with corresponding eigenvectors that are null vectors of B. The theory for (1.2) is

* Received by the editors June 24, 1992; accepted for publication (in revised form) July 20, 1993.
† Theoretical Studies Department, B424.4, Harwell Laboratory, Didcot, OX11 0RA, United Kingdom. This author's research forms a part of the Underlying Corporate Research Programme of AEA Technology, Harwell.

‡ AspenTech UK Ltd, Sheraton House, Castle Park, Cambridge, CB3 0AX, United Kingdom. This author's research was supported by a United Kingdom Science and Engineering Research Council of Canada CASE studentship at Bath University and AEA Technology, Harwell.

§ School of Mathematical Sciences, University of Bath, Bath, BA2 7AY, United Kingdom (as@uk.ac.bath.maths). This author's research was supported by the United Kingdom Science and Engineering Research Council and National Science Foundation cooperative agreement CCR-8809615.

more complicated than for the standard eigenvalue problem $Aw = \mu w$ [21]. However, the assumptions made in this paper on C and M allow very precise statements to be made about the number of eigenvalues of (1.2) and make possible the introduction of related eigenvalue problems that are better suited to solution by iterative algorithms.

Eigenvalue problems of the form (1.2) with block structure (1.1) arise in applications involving constraints. For example, Malkus [14] discusses the case when K is symmetric in an analysis of the discrete Ladyzhenskaya–Babuska–Brezzi (LBB) stability condition for incompressible finite elements arising in linear elasticity or Stokes flow. For symmetric K, the results on the eigenvalues of EVP1 in Theorem 2.1 and Lemma 2.2 are contained in [14, Thm. 3] though our method of proof produces the results in a more direct manner. In addition, the case of K symmetric and $M = I$ is discussed by Golub in [10].

The plan of the paper is as follows. In §2 the basic theory for the eigenvalue problem (1.1), (1.2) is presented. Section 3 contains an analysis of some related eigenvalue problems that provide some shift strategies for the eigenvalues. In §4 some practical aspects are considered. First there is a discussion relating to the execution of certain matrix-vector operations and second the estimation of the accuracy of computed eigensolutions is examined. Section 5 contains a discussion of strategies that could be used to determine the eigenvalues of smallest real part of (1.1), (1.2). These are illustrated with reference to a matrix problem arising from a finite element discretization of two-dimensional double-diffusive convection in a box.

2. Theory for the eigenvalue problem. This section contains some results about the eigenvalue problem $Aw = \mu Bw$ which, for convenience, we rewrite (and rename) as

$$\text{(EVP1)} \qquad \begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \mu \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix},$$

where, in analogy with our applications arising from the discretization of the Navier–Stokes equations, we use the notation $w = [u, p]$, $u \in \boldsymbol{R}^n$, $p \in \boldsymbol{R}^m$, where u and p correspond to velocity and pressure degrees of freedom, respectively.

First we note that since C is full rank, the QR factorisation of C has the form (ignoring possible permutations that play no role here)

$$C = QR = [Q_1 \; Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \; (= Q_1 R_1), \tag{2.1}$$

where R is $n \times m$, R_1 is $m \times m$ nonsingular and upper triangular, Q is $n \times n$ orthogonal, Q_1 is $n \times m$ and provides an orthonormal basis for range(C), and Q_2 is $n \times (n-m)$ and provides an orthonormal basis for $C^{\perp}$. For future use note that C^TC is $m \times m$ positive definite and has the Cholesky decomposition $C^TC = R_1^T R_1$ [11, p. 217]. Also, the matrix

$$\pi := I - C(C^TC)^{-1}C^T \tag{2.2}$$

is a projection from $\boldsymbol{R}^n$ onto $C^{\perp}$ along range(C).

Now we state a fundamental result on the number of eigenvalues of EVP1.

THEOREM 2.1. (a) *The eigenvalue problem* EVP1 *has precisely* $n-m$ *eigenvalues, that are those of the reduced eigenvalue problem of dimension* $(n-m)$

$$\text{(REVP1)} \qquad Q_2^T(K - \mu M)Q_2 z = 0.$$

(b) *If* (μ, z), $z \in \boldsymbol{R}^{n-m}$ *is an eigensolution of* REVP1 *then* (μ, u, p) *is a corresponding eigensolution of* EVP1 *where*

$$u = Q_2 z, \qquad p = -R_1^{-1} Q_1^T (K - \mu M) Q_2 z.$$

(c) *If* (μ, u, p), $u \in \boldsymbol{R}^n$, $p \in \boldsymbol{R}^m$ *is an eigensolution of* EVP1 *then* (μ, z) *is a corresponding eigensolution of* REVP1 *where* $z = Q_2^T u$.

Proof. (a) Introduce $Z = \left[\begin{smallmatrix} Q & 0 \\ 0 & I_m \end{smallmatrix}\right]$, where Q is defined in (2.1), and $y := Z^T w = (Q^T u, p) = (Q_1^T u, Q_2^T u, p) =: (u_1, u_2, p)$. Now EVP1 is equivalent to $Z^T A Z y = \mu Z^T B Z y$, that in block form becomes

$$\begin{bmatrix} K_{11} & K_{12} & R_1 \\ K_{21} & K_{22} & 0 \\ R_1^T & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ p \end{bmatrix} = \mu \begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ p \end{bmatrix}, \tag{2.3}$$

where $K_{ij} = Q_i^T K Q_j$, $M_{ij} = Q_i^T M Q_j$, $i, j = 1, 2$. Simple manipulation shows that $u_1 = 0$, $(K_{22} - \mu M_{22}) u_2 = 0$, $p = -R_1^{-1}(K_{12} - \mu M_{12}) u_2$. Since M and M_{22} are symmetric positive definite, (a) is immediate. Results (b) and (c) now follow. □

Remark 1. An equivalent proof in the style of Golub [10] uses the projection π defined by (2.2). Since $\pi u = u$ for $u \in C^{\perp}$ we may write the first row of EVP1 as $K\pi u + Cp = \mu M \pi u$. Premultiplication by π gives

$$\pi K \pi u = \mu \pi M \pi u \tag{2.4}$$

that has the same eigenvalues as (REVP1) plus m zero eigenvalues corresponding to eigenvectors lying in range(C) that have no relevance for EVP1.

For future analysis it is convenient to exclude the possibility that $\mu = 0$ is an eigenvalue, and so we assume the following:

$$\mu = 0 \text{ is } \textit{not} \text{ an eigenvalue of EVP1.} \tag{2.5}$$

This assumption is not a severe restriction in the applications we have in mind, since though zero eigenvalues are important, corresponding to "steady-state" bifurcations in some nonlinear problem, they are usually detected readily. For example, if a direct solver for A is feasible then one can check the determinant of A. If a zero eigenvalue of EVP1 has been found, an eigenvalue problem satisfying (2.5) may be obtained by considering a "shifted" eigenvalue problem with the same structure as EVP1, but with K replaced by $K - \gamma M$ for some appropriately chosen shift $\gamma \in \boldsymbol{R}$.

Under assumption (2.5), K_{22} is nonsingular and so (2.3) may be rewritten as

$$\nu w = \begin{bmatrix} 0 & 0 & R_1^{-T} \\ 0 & K_{22}^{-1} & * \\ R_1^{-1} & * & * \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} w$$

or

$$\nu w = \begin{bmatrix} 0 & 0 & 0 \\ * & K_{22}^{-1} M_{22} & 0 \\ * & * & 0 \end{bmatrix} w, \tag{2.6}$$

where $*$ denotes unique submatrices that do not affect the analysis. Clearly (2.6) has a zero eigenvalue of algebraic multiplicity $2m$. Thus we have proved the following lemma.

LEMMA 2.2. *Under* (2.5), EVP1 *has an infinite eigenvalue of multiplicity* $2m$.

Similar results apply to the following generalization of EVP1,

$$\begin{bmatrix} K & C_1 \\ C_2^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \mu \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix}, \tag{2.7}$$

where C_1 and C_2 are $n \times m$ matrices of rank m. Such a problem arises after the discretization of the Navier–Stokes equations by a spectral method [6]. Provided C_1 and C_2 satisfy the nondegeneracy condition,

$$C_2^T C_1 \text{ is invertible}$$

(which is a natural condition in the context of discretizations of the Navier–Stokes equations); then results analogous to Theorem 2.1 hold. To see this we introduce the projection operator

$$\pi_{12} := I - C_1(C_2^T C_1)^{-1} C_2^T \tag{2.8}$$

(cf. (2.2)). Clearly, if $u \in \text{range}(C_1)$, then $\pi_{12}u = 0$, and if $u \in C_2^{\perp}$, then $\pi_{12}u = u$.

Now follow the approach of Remark 1 to obtain

$$\pi_{12} K \pi_{12} u = \mu \pi_{12} M \pi_{12} u \tag{2.9}$$

(cf. (2.4)), which has $n - m$ eigenvalues for $u \in C_2^{\perp}$. A corresponding reduced eigenvalue problem can be derived (cf. Theorem 2.1(a)).

Remark 2. It is important to note that if K and M are large and sparse (as is the case in the applications we have in mind) then typically one would not explicitly form the matrices in (REVP1). This is because Q_2 is full and hence $Q_2^T K Q_2$ and $Q_2^T M Q_2$ are full. Rather one would employ iterative techniques to find selected eigenvalues as discussed in §5.

3. Some shifted eigenvalue problems. It is a common technique to shift the eigenvalues of an eigenvalue problem, so that if $Aw = \mu Bw$ then $(A - \gamma B)w = (\mu - \gamma)Bw$ and all the eigenvalues μ are shifted by γ, with the corresponding eigenvector remaining unchanged. In this section we look at some generalised eigenvalue problems that are closely related to EVP1 in that the new eigenvalue problems allow us to shift both the finite and infinite eigenvalues. First consider the eigenvalue problem

$$\begin{bmatrix} K - \gamma M & \delta_1 C \\ \delta_1 C^T & 0 \end{bmatrix} v = \sigma \begin{bmatrix} M & \delta_2 C \\ \delta_2 C^T & 0 \end{bmatrix} v \tag{EVP2}$$

for some $\delta_1, \delta_2, \gamma \in \boldsymbol{R}$. Note that $\gamma = \delta_2 = 0, \delta_1 = 1$ recovers EVP1. We have the following theorem about the eigenvalues of EVP2.

THEOREM 3.1. *Denote the finite eigenvalues of* REVP1 *by* $\mu_i, i = 1, \ldots, n - m$. *Assume* (2.5) *and*

$$\text{(i) } \delta_2 \neq 0, \qquad \text{(ii) } \delta_1 \delta_2^{-1} \neq \mu_i - \gamma. \tag{3.1}$$

Then EVP2 *has eigenvalues* σ_i, $i = 1, \ldots, n + m$ *with*

(a) $\sigma_i = \mu_i - \gamma$, $i = 1, \ldots, n - m$

(b) $\sigma_i = \delta_1 \delta_2^{-1}$, $i = n - m + 1, \ldots, n + m$.

Proof. Since M is positive definite it is straightforward to show, under (3.1(i)), that the matrix on the right-hand side of EVP2 is nonsingular and hence EVP2 has $n+m$ eigenvalues. Under the transformation introduced in the proof of Theorem 2.1, EVP2 is equivalent to

$$\begin{bmatrix} K_{11}-\gamma M_{11} & K_{12}-\gamma M_{12} & \delta_1 R_1 \\ K_{21}-\gamma M_{21} & K_{22}-\gamma M_{22} & 0 \\ \delta_1 R_1^T & 0 & 0 \end{bmatrix} v = \sigma \begin{bmatrix} M_{11} & M_{12} & \delta_2 R_1 \\ M_{21} & M_{22} & 0 \\ \delta_2 R_1^T & 0 & 0 \end{bmatrix} v,$$

and premultiplication by the inverse of the matrix on the right-hand side produces the eigenvalue problem

$$\begin{bmatrix} \delta_1\delta_2^{-1} & 0 & 0 \\ * & M_{22}^{-1}(K_{22}-\gamma M_{22}) & 0 \\ * & * & \delta_1\delta_2^{-1} \end{bmatrix} v = \sigma v$$

(where again $*$ denotes unique submatrices). Clearly this matrix has an eigenvalue at $\delta_1\delta_2^{-1}$ of multiplicity $2m$ and $n-m$ eigenvalues satisfying $(K_{22}-\gamma M_{22})z = \sigma M_{22} z$, which is precisely a simple shift of REVP1. Statements (a) and (b) now follow. □

Thus we see that the finite eigenvalues μ_i of EVP1 are shifted by γ, but the infinite eigenvalues are transformed to $\delta_1\delta_2^{-1}$. Thus any eigenvalue of EVP1 may be shifted to any location. Two examples of EVP2 that are the most likely to be useful are now introduced. Consider

$$\text{(EVP3)} \qquad \begin{bmatrix} K-\gamma M & 0 \\ 0 & 0 \end{bmatrix} v = \sigma \begin{bmatrix} M & C \\ C^T & 0 \end{bmatrix} v.$$

Clearly this has $n-m$ eigenvalues $\sigma_i = \mu_i - \gamma, i = 1, \ldots, n-m$, and $2m$ eigenvalues at zero. Similarly the eigenvalue problem

$$\text{(EVP4)} \qquad \begin{bmatrix} K & \delta_1 C \\ \delta_1 C^T & 0 \end{bmatrix} v = \sigma \begin{bmatrix} M & C \\ C^T & 0 \end{bmatrix}$$

leaves the finite eigenvalues μ_i of EVP1 unchanged, but transforms the infinite eigenvalues to δ_1. Note however that not all eigenvectors are preserved by the shifts. Eigenvectors of EVP1 corresponding to the infinite eigenvalues have the form $(0, w_2)$ and these are unchanged. However the eigenvectors of the finite eigenvalues of EVP1 are changed, but only in the last m components. To be precise, we have the following result, which may be deduced from the block form of the equations for EVP1 and EVP2.

LEMMA 3.2. (a) *Let μ be a finite eigenvalue of* EVP1. *Assume* (2.5) *and* (3.1). *If $(w_1, w_2), w_1 \in \boldsymbol{R}^n, w_2 \in \boldsymbol{R}^m$ is an eigenvector of* EVP1 *associated with μ, then (v_1, v_2), where $v_1 = w_1, v_2 = (\delta_1 - (\mu-\gamma)\delta_2)^{-1} w_2$ is the corresponding eigenvector of* EVP2.

(b) *Let $(0, w_2), w_2 \in \boldsymbol{R}^m$ be an eigenvector corresponding to an infinite eigenvalue of* EVP1. *Then $(0, v_2)$ with $v_2 = w_2$ is an eigenvector of* EVP2 *corresponding to the eigenvalue $\delta_1\delta_2^{-1}$.*

Remark 3. We have not gone into the detailed structure of the infinite eigenvalue of EVP1 or the eigenvalue $\delta_1\delta_2^{-1}$ of EVP2 of algebraic multiplicity $2m$. However analysis using the Weierstrass–Kronecker canonical form reveals that $\delta_1\delta_2^{-1}$ has geometric multiplicity m and that the eigenvalues occur in 2×2 pairs corresponding to Jordan

blocks of the form

$$\begin{bmatrix} \delta_1\delta_2^{-1} & 1 \\ 0 & \delta_1\delta_2^{-1} \end{bmatrix}.$$

This is to be expected following the results in [14].

4. Computational considerations. For K, C, and M small and/or full it will probably be a reasonable strategy to form $K_{22} = Q_2^T K Q_2$ and $M_{22} = Q_2^T M Q_2$ and to solve REVP1 directly. However, if K, C, and M are large and sparse the "reduced" matrices K_{22} and M_{22} will be full, and transformation methods will probably not be feasible. Iterative methods like simultaneous iteration or Arnoldi's method (probably applied to the EVP3 or EVP4) become attractive. It is therefore necessary to implement matrix $\times$ vector operations, which for the generalised eigenvalue problem reduce to solving systems of $(n+m)$ dimensional linear equations. These could be carried out directly on the $(n+m)$ dimensional systems [5], [7], [12], but an approach that involves only solving n dimensional systems is possible.

This is illustrated with reference to EVP3 with $\gamma = 0$. An iterative method will require the solution of a linear system of the form

$$\begin{bmatrix} M & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ q \end{bmatrix}, \qquad v \text{ given}, \tag{4.1}$$

using an n dimensional system with coefficient matrix M.

It is readily shown (say by a block Gaussian elimination approach) that (4.1) can be solved by the following algorithm:

(i) solve $Mw = Kv$ for $w \in \boldsymbol{R}^n$,
(ii) solve $[C^T M^{-1} C]p = C^T w$ for $p \in \boldsymbol{R}^m$,
(iii) solve $Mx = Cp$ for $x \in \boldsymbol{R}^n$,
(iv) set $u = w - x$.

This is the Uzawa algorithm [1] and to be efficient, step (ii) would be carried out iteratively to avoid the direct computation of $C^T M^{-1} C$. For example, one could precondition $C^T M^{-1} C$ by $C^T M_l^{-1} C$, where M_l is the "lumped mass" matrix derived from M.

"Shift-invert" [16] or Cayley transform [7], [12] techniques are also possible with iterative methods like subspace iteration, requiring the solution of *nonsymmetric* systems of the form

$$\begin{bmatrix} K - \gamma M & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} \nu \\ 0 \end{bmatrix} \tag{4.2}$$

for appropriate γ, and for three-dimensional partial differential equations this system will invariably have to be solved iteratively. However, if good estimates of the wanted eigenvalues are available and if direct solution of (4.2) is possible, then these approaches are likely to prove very efficient.

A standard approach to estimate the accuracy of an approximate eigenpair $(\tilde{\mu}, \tilde{w})$ of $Aw = \mu w$ is to calculate the "residual" vector $r := A\tilde{w} - \tilde{\mu}\tilde{w}$. Standard backward error analysis [23, p. 171] shows that typically a "small" residual indicates that $\tilde{\mu}$ is a "good" approximation to μ. A similar analysis holds for problems considered here. Let $(\tilde{\mu}, [\tilde{u}, \tilde{p}])$ denote an approximate eigenpair with $C^T \tilde{u} = 0$. Then the corresponding residual vector $r \in \boldsymbol{R}^{n+m}$ has the form

$$r = (r_1, 0), \qquad r_1 := K\tilde{u} + C\tilde{p} - \tilde{\mu} M \tilde{u} \in \boldsymbol{R}^n. \tag{4.3}$$

With $\tilde{u}^T\tilde{u} = 1$ it is readily shown that

$$\begin{bmatrix} K - r_1\tilde{u}^T & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{p} \end{bmatrix} = \tilde{\mu} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{p} \end{bmatrix},$$

i.e., $(\tilde{\mu}, [\tilde{u}, \tilde{p}])$ is an exact eigenpair of a perturbed problem (with the same block structure as (1.1), (1.2)). In §5 we use $\|r_1\|_2 (= \|r\|_2)$ to test the accuracy of an approximate eigenpair (cf. p383 of [11]).

5. Applications. The eigenvalue problem arises in the determination of the stability of steady solutions to the Navier–Stokes and related equations, and linearised stability theory [8], [19] shows that stability is determined by the eigenvalues with smallest real part of a linearised problem. The eigenvalue problem EVP1 arises if a mixed finite element method is used to discretize the linearised problem [5], [9], [15]. Of special interest is the case when the eigenvalues of smallest real part are complex, since algorithms for the detection of Hopf bifurcations in parameter dependent systems can be developed from knowledge of these eigenvalues. The matrices K, C, and M are sparse and very large. In [12], a problem with over 2×10^5 degrees of freedom is studied.

Since EVP1 involves large sparse nonsymmetric matrices one must fall back on iterative methods like Arnoldi's method or simultaneous iteration [2] to compute wanted eigenvalues. As mentioned in §4, shift-invert strategies are possible if good estimates of wanted eigenvalues are known. However, one can also apply iterative methods directly to EVP3 or EVP4 with appropriate choices for γ or δ_1, δ_2, respectively. To be precise two approaches are given.

(a) Choose $\gamma \approx \{\mathrm{Re}(\mu_1) + \mathrm{Re}(\mu_{n-m})\}/2$ in EVP3. Thus with $\rho_i = \mu_i - \gamma$, we have $\mathrm{Re}(\rho_1) + \mathrm{Re}(\rho_{n-m}) \approx 0$. The zero eigenvalue of multiplicity $2m$ of EVP3 is "in the middle" of the spectrum.

(b) Choose $\delta_1 \approx \{\mathrm{Re}(\mu_1) + \mathrm{Re}(\mu_{n-m})\}/2$ in EVP4. Thus the eigenvalue of multiplicity $2m$ of EVP2 is "in the middle" of the spectrum of EVP4, with $\mu_1 \ldots \mu_{n-m}$ being unchanged.

In both EVP3 and EVP4 the troublesome $2m$ multiple eigenvalue, which corresponds to the infinite eigenvalue of EVP1 (Lemma 2.2), will not be computed by the iterative algorithm. We note that the recent implicit polynomial filters algorithm of Sorensen [20] would appear to be an appropriate method to apply to reformulations like EVP2.

If the eigenvalues of EVP1 are known to be real, then simple shift strategies based on EVP3 and EVP4 allied to iterative methods provide the largest and smallest eigenvalues. However, when the eigenvalues may be complex then one may have to further transform the matrix eigenvalue problem. One approach is to utilize the Chebyshev transformation ideas of Saad [17], [18]. For a standard eigenvalue problem $Aw = \mu w$, the idea is to carry out a shifted Chebyshev polynomial transformation of A, say to $p_s(A)$, where the eigenvalues μ_i of A lying inside a certain ellipse in the complex plane are mapped to eigenvalues $p_s(\mu_i)$ of $p_s(A)$ satisfying $|p_s(\mu_i)| < 1$. The aim is to choose the polynomial p_s (and hence the ellipse) so that only the desired eigenvalues of A lie outside the ellipse. These become dominant, well-separated eigenvalues of $p_s(A)$ and hence are computed by an iterative solver applied to $p_s(A)$. These techniques were applied successfully to EVP3 to find eigenvalues of smallest real part of two problems from fluid dynamics and the results were reported in [5] and [7]. Transformation EVP4 was used to provide the numerical results in the following example.

5.1. Example. We consider a matrix arising from a mixed finite element discretization of the equations modeling two-dimensional double-diffusive convection in a box heated on the bottom boundary (see [22, Chap. 8]). The governing equations are solved in the Boussinesq approximation and are given in [4]. We do not reproduce these equations here but note that the model has nondimensional parameters: Prandtl number Pr, Rayleigh number Ra, salinity Rayleigh number Rs and τ (see [4, p. 254]), and interest centres on the loss of stability as Ra increases (which corresponds to increasing the temperature difference between the top and bottom boundary). Our calculations were performed with $\mathrm{Pr} = 10$ and $\tau = 10^{-2}$, which corresponds roughly to a salt solution and water, $\mathrm{Rs} = 2000$ and $\mathrm{Ra} = 2480$. The exact eigenvalues of the continuous problem, μ_i^e say, are known [22, (8.18)]) and the three leftmost eigenvalues are $\mu_1^e, \mu_2^e = 0.047486 \pm 24.502i$, $\mu_3^e = 0.098696$. A mixed finite element approximation was obtained in the usual way [13] using nine-node quadrilateral elements with biquadratic interpolation for velocities, temperatures and salinities, and discontinuous piecewise-linear interpolation for pressures [4]. The matrix was set up using ENTWIFE [24] with a 4×4 grid that leads to a matrix with the block structure of (1.1) with $n = 324$, $m = 48$, and hence $N(= n + m) = 372$. The three leftmost eigenvalues of the matrix problem to seven significant figures are $\mu_1, \mu_2 = 0.04932671 \pm 24.51725i$, $\mu_3 = 0.09874659$. (In fact numerical values are known with residuals (see (4.2)) less than 0.25×10^{-13}.)

In the following, "Arnoldi (k, l)" means Arnoldi's method restarted l times with subspace of dimension k [18]. First Arnoldi (20,1) with a random starting vector was applied to

$$A_1 := \begin{bmatrix} M & C \\ C^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix}$$

to obtain a very rough idea of the convex hull of the eigenvalues of EVP2 with $\gamma = 0$, $\delta_1 = 0$, and $\delta_2 = 1$, and hence a rough estimate of μ_{n-m} is obtained. (Of course, the matrix $\times$ vector operations with A_1 were performed by solving linear systems.) Next form EVP4 with $\delta_2 = 1$, $\delta_1 = \mathrm{Re}(\mu_{n-m})/2$, in line with strategy (b) above, and try to find the leftmost eigenvalues of

$$A_2 := \begin{bmatrix} M & C \\ C^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} K & \delta_1 C \\ \delta_1 C^T & 0 \end{bmatrix}.$$

Arnoldi (20, 50) failed to find any of the three leftmost eigenvalues, perhaps because of severe clustering of the eigenvalues [18]. The hybrid algorithm of Saad [18] (see above) utilizing the Chebyshev transformation to find the two leftmost eigenvalues (so that the ellipse passes through μ_3) is however successful. A two-step procedure was used:

(i) Arnoldi (20, 1) with a random starting vector was applied to $p_5(A_2)$ to obtain a "purified" starting vector.

(ii) Arnoldi (20, 1) was applied to $p_{42}(A_2)$. This computed μ_1, μ_2 with residuals less than 5×10^{-12} (see (4.3)).

Numerical experiments using EVP3 with γ chosen in (a) above produce similar results. This is not surprising since the distribution of the extremal eigenvalues of EVP3 and EVP4 is the same.

Acknowledgments. Y. Saad kindly sent us a copy of his hybrid algorithm [18] that was used to carry out the computations in §5.

REFERENCES

[1] K. ARROW, L. HURWICZ AND H. UZAWA, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[2] F. CHATELIN, *Spectral Approximation of Linear Operators*, Academic Press, New York, 1983.

[3] K. A. CLIFFE, *Numerical calculations of two-cell and single-cell Taylor flows*, J. Fluid Mech., 135 (1983), pp. 219–233.

[4] K. A. CLIFFE AND K. H. WINTERS, *Convergence properties of the finite-element method for Bénard convection in an infinite layer*, J. Comput. Physics, 60 (1985), pp. 346–351.

[5] K. A. CLIFFE, T. J. GARRATT, AND A. SPENCE, *Calculation of eigenvalues of the discretised Navier–Stokes and related equations*, in The Mathematics of Finite Elements and Applications, VII MAFELAP, J.R. Whiteman, ed., Academic Press, New York, 1990, pp. 470–486.

[6] A. J. CONLEY, *Spectral Methods for* 3D *Navier–Stokes Equations*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1992.

[7] T. J. GARRATT, *The Numerical Detection of Hopf Bifurcations in Large Systems Arising in Fluid Mechanics*, Ph.D. Thesis, School of Mathematical Sciences, University of Bath, UK, 1991.

[8] A. GEORGESCU, *Hydrodynamic Stability Theory*, Martinus Nijhoff, Dordrecht, the Netherlands, 1985.

[9] V. GIRAULT AND P. A. RAVIART, *Finite Element Approximation of the Navier–Stokes Equations*, Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1979.

[10] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.

[11] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1983.

[12] P. M. GRESHO, D. K. GARTLING, J. R. TORCZYNSKI, T. J. GARRATT, K. A. CLIFFE, A. SPENCE, K. H. WINTERS, AND J. W. GOODRICH, *Is the steady viscous incompressible* 2D *flow over a backward-facing step at* Re = 800 *stable?*, Internat. J. Numer. Meth. Fluids, 17 (1993), pp. 501–541.

[13] C. JOHNSON, *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, 1987.

[14] D.S. MALKUS, *Eigenproblems associated with the discrete* LBB *condition for incompressible finite elements*, Internat. J. for Engrg. Sci., 19 (1981), pp. 1299–1310.

[15] B. MERCIER, J. OSBORN, J. RAPPAZ, AND P. A. RAVIART, *Eigenvalue approximation by mixed and hybrid methods*, Math. Comp., 36 (1981), pp. 427–453.

[16] B. N. PARLETT AND Y. SAAD, *Complex shift and invert strategies for real matrices*, Linear Algebra Appl., 88/89 (1987), pp. 575–595.

[17] Y. SAAD, *Practical use of some Krylov subspace methods for solving indefinite and unsymmetric linear equations*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 203–228.

[18] ———, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.

[19] D. H. SATTINGER, *Transformation groups and bifurcation at multiple eigenvalues*, Bull. Amer. Math. Soc., 79 (1973), pp. 709–711.

[20] D. G. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[21] G. W. STEWART, *Perturbation theory for the generalised eigenvalue problem*, in Advances in Numerical Analysis, G.H. Golub and C. de Boor, eds., Academic Press, New York, 1978.

[22] J. S. TURNER, *Buoyancy effects in fluids*, CUP, Cambridge, 1973.

[23] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.

[24] K. H. WINTERS, *ENTWIFE User Manual (Release* 1), Harwell report AERE-R 11577, Harwell Laboratory, Didcot, UK, 1985.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1319–1332, October 1994

018

ON THE PERTURBATION OF THE CHOLESKY FACTORIZATION*

ZLATKO DRMAČ†, MATJAŽ OMLADIČ‡, AND KREŠIMIR VESELIĆ§

Abstract. The perturbation of the Cholesky factor of a perturbed positive definite matrix is considered. Estimates are included for small perturbations in the spectral norm as well as for large perturbations in the Euclidean norm. The results can be applied to floating point perturbations as well.

Key words. Cholesky factor, relative perturbation, floating point perturbation

AMS subject classifications. 15A23, 15A45

1. Introduction.

Let H be a Hermitian positive definite matrix of order n. Then, as it is well known, there exists a unique upper triangular matrix R with real and positive diagonal elements such that

$$H = R^*R. \tag{1}$$

The matrix R is called the *Cholesky factor* of H. It is given by the recursive formulae (*Cholesky algorithm*)

$$|R_{1i}|^2 + \cdots + |R_{ii}|^2 = H_{ii}, \qquad 1 \le i \le n, \tag{2}$$

$$\bar{R}_{1i}R_{1j} + \cdots + \bar{R}_{ii}R_{ij} = H_{ij}, \quad i = 1, \ldots, n-1, \quad j = i+1, \ldots, n. \tag{3}$$

From this it follows that R depends continuously on H.

In this note we derive some explicit perturbation estimates for the Cholesky factorization. Let $H + \delta H$ be the perturbed positive definite matrix and $R + \delta R$ be its perturbed Cholesky factor. Then

$$H + \delta H = (R + \delta R)^*(R + \delta R) = R^*(I + A)R, \tag{4}$$

where[1] $A = R^{-*}\delta H R^{-1}$, so we may go over to the "normalized" problem

$$I + A = (I + \Gamma)^*(I + \Gamma). \tag{5}$$

Here, of course,

$$\Gamma_{ii} > -1, \qquad 1 \le i \le n. \tag{6}$$

Once a norm estimate for Γ is obtained, we can estimate $\delta R = \Gamma R$ as

$$\|\delta R\| \le \|\Gamma\|\|R\|.$$

* Received by the editors February 22, 1993; accepted for publication (in revised form) August 4, 1993.

† Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Postfach 940, 58084 Hagen, Germany (zlatko.drmac@fernuni-hagen.de).

‡ Fakulteta za naravoslovje in tehnologijo, Jadranska 19, 61000 Ljubljana, Slovenia.

§ Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Postfach 940, 58084 Hagen, Germany (ma704@dhafeu11.bitnet).

[1] Here $R^{-*} = (R^{-1})^*$.

Sun [12] obtained an estimate for the Frobenius norm $\|\Gamma\|_F$ and then

$$\|\delta R\|_F \leq \|\Gamma\|_F \|R\|_2. \tag{7}$$

We treat more general norms including an estimate for the spectral norm $\|\Gamma\|_2$. Then

$$\|\delta R x\|_2 \leq \|\Gamma\|_2 \|Rx\|_2, \qquad x \in \mathbf{C}^n. \tag{8}$$

From this we obtain

$$\|\delta R\|_2 \leq \|\Gamma\|_2 \|R\|_2. \tag{9}$$

From (8) we obtain a columnwise estimate

$$\|\delta R_{\cdot,i}\|_2 \leq \|\Gamma\|_2 \|R_{\cdot,i}\|_2, \tag{10}$$

which may be more appropriate for differently scaled columns. Section 2 gives estimates for $\|\Gamma\|$ in terms of the same norm $\|A\|$, if the latter is small enough. For the spectral norm we obtain

$$\|\Gamma\|_2 \leq \frac{2c_n\|A\|_2}{1+\sqrt{1-4c_n^2\|A\|_2}}, \qquad c_n = \frac{1}{2} + [\log_2 n]. \tag{11}$$

The estimate $\|A\|_2 \leq \|H^{-1}\|_2 \|\delta H\|_2$ may be pessimistic, if we deal with the floating point perturbation of the type $|\delta H_{ij}| \leq \varepsilon |H_{ij}|$. Section 3 gives an estimate for A in terms of ε.

In Section 4 we prove some large perturbation estimates for the Frobenius norm. For example, whenever $I + A$ is positive definite, we have a simple global estimate

$$\|\Gamma\|_F \leq \sqrt{8n + 2\sqrt{n}}\|A\|_F. \tag{12}$$

After a first version of this paper was written, the authors became aware of a related paper [4] where results slightly sharper than (11), (12) were obtained by other methods. We thank the referee for pointing this out.

2. Small perturbations. We first consider the Cholesky factorization of $I + A$, $\|A\|_2 < 1$. For a Hermitian A set

$$\mathcal{P}(A) = \begin{bmatrix} \frac{A_{11}}{2} & A_{12} & \cdots & A_{1n} \\ 0 & \frac{A_{22}}{2} & \cdots & A_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{A_{nn}}{2} \end{bmatrix}. \tag{13}$$

Obviously $\mathcal{P}$ is an invertible linear operator mapping the real space $\mathcal{H}$ of Hermitian matrices onto the real algebra $\mathcal{T}$ of upper triangular matrices with real diagonal. (cf., also, Stewart and Sun [10]–[12].) For any $S \in \mathcal{T}$ we have

$$\mathcal{P}(S + S^*) = S. \tag{14}$$

Let $\|\cdot\|$ be any norm on $\mathcal{T}$. By the same symbol we denote the corresponding operator norm

$$\|\mathcal{P}\| = \max_{\substack{A\in\mathcal{H}\\ \nu(A)=1}} \|\mathcal{P}(A)\|, \tag{15}$$

where ν is a norm on $\mathcal{H}$. By $\mathcal{P}(I) = I/2$ we have

$$\|\mathcal{P}\| \geq \frac{1}{2}\frac{\|I\|}{\nu(I)}. \tag{16}$$

The following theorem is a strengthening of the corresponding result from [12]. (The latter deals with the Frobenius norm. An asymptotic estimate of this form was derived by Barlow and Demmel [1].) Our proof is simpler and is based on a homotopy argument.

THEOREM 2.1. *Let $A \in \mathcal{H}$, $S = \mathcal{P}(A)$, and $I + A$ be positive definite. Let*

$$p = \max \frac{\|\mathcal{P}(\Gamma^*\Gamma)\|}{\|\Gamma\|^2}, \tag{17}$$

where the maximum is taken over all $\Gamma \in \mathcal{T}$ with $\Gamma_{ii} > -1$ for all i. If

$$4p\|S\| \leq 1, \tag{18}$$

then the Cholesky factor of $I + A$ equals $I + \Gamma$ with

$$\|\Gamma\| \leq \frac{2\|S\|}{1 + \sqrt{1 - 4p\|S\|}}. \tag{19}$$

Proof. For $0 \leq \eta \leq 1$, the matrix $I + \eta A$ is positive definite and its Cholesky factorization is

$$(I + \Gamma_\eta)^*(I + \Gamma_\eta) = I + \eta A, \tag{20}$$

where, as mentioned above, Γ_η depends continuously on η and $\Gamma_0 = 0$, $\Gamma_1 = \Gamma$. Now (20) is equivalent to

$$\Gamma_\eta = \eta S - \mathcal{P}(\Gamma_\eta^*\Gamma_\eta). \tag{21}$$

Hence

$$\|\Gamma_\eta\| \leq \eta\|S\| + p\|\Gamma_\eta\|^2,$$

which implies that $\|\Gamma_\eta\|$ lies outside of the nonvoid open interval

$$\left(\frac{1 - \sqrt{1 - 4p\eta\|S\|}}{2p}\ ,\quad \frac{1 + \sqrt{1 - 4p\eta\|S\|}}{2p}\right).$$

Now by continuity, $\|\Gamma_\eta\|$ must lie left from that interval. Thus, (19) follows. □

From (19) it follows that

$$\|\Gamma\| \leq \|S\|(1 + 4p\|S\|) \leq 2\|S\|.$$

Using this and (21) we obtain

$$\|S\|(1-4p\|S\|) \le \|\Gamma\| \le \|S\|(1+4p\|S\|),$$

which shows that the estimate (19) cannot be essentially sharpened for small A.

By (17), (19), the practical use of Theorem 2.1 depends on p and, in the case of a self-adjoint matrix norm $\nu = \|\cdot\|$, its upper bound $\|\mathcal{P}\|$. We shall consider the Frobenius norm

$$\|A\|_F = \sqrt{\mathrm{Tr} A^* A},$$

and the spectral norm

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2, \qquad \|x\|_2 = (x^*x)^{1/2}.$$

The corresponding p's from Theorem 2.1 are denoted by p_F and p_2, respectively. We have

$$\|\mathcal{P}(A)\|_F^2 = \sum_{\substack{i,j \\ i<j}} |A_{ij}|^2 + \frac{1}{4}\sum_i |A_{ii}|^2 \le \sum_{\substack{i,j \\ i<j}} |A_{ij}|^2 + \frac{1}{2}\sum_i |A_{ii}|^2 = \frac{1}{2}\|A\|_F^2.$$

Here the equality sign is attained at any A with zero diagonals. Thus,

$$p_F \le \|\mathcal{P}\|_F = \frac{1}{\sqrt{2}}. \tag{22}$$

Now by Theorem 2.1, $\|S\|_F < 1/(2\sqrt{2})$ implies

$$\|\Gamma\|_F \le \frac{2\|S\|_F}{1+\sqrt{1-2\sqrt{2}\|S\|_F}} \le \frac{\sqrt{2}\|A\|_F}{1+\sqrt{1-2\|A\|_F}}. \tag{23}$$

The spectral norm is more complicated. A simple estimate

$$p_2 \le \|\mathcal{P}\|_2 \le \sqrt{\frac{n}{2}} \tag{24}$$

follows immediately from (22). For a better estimate we use a binary splitting to decompose the space of Hermitian matrices into a direct sum of subspaces that we illustrate for $n = 7$.

$$A = \begin{bmatrix} \cdot & \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * & * \\ * & * & * & \cdot & \cdot & \cdot & \cdot \\ * & * & * & \cdot & \cdot & \cdot & \cdot \\ * & * & * & \cdot & \cdot & \cdot & \cdot \\ * & * & * & \cdot & \cdot & \cdot & \cdot \end{bmatrix} + \begin{bmatrix} \cdot & * & * & \cdot & \cdot & \cdot & \cdot \\ * & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ * & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & * & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & * & * & \cdot & \cdot \end{bmatrix}$$

$$+ \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & * & \cdot & \cdot & \cdot & \cdot \\ \cdot & * & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * & \cdot \end{bmatrix} + \begin{bmatrix} * & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & * & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & * & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & * & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}.$$

In general, it holds that

$$A = A_q + \cdots + A_1 + A_0, \qquad q = [\log_2 n],$$

where the matrix A_0 is the diagonal part of A. This direct decomposition has the properties

$$\|\mathcal{P}(A_0)\|_2 = \frac{1}{2}\|A_0\|_2, \quad \|\mathcal{P}(A_i)\|_2 = \|A_i\|_2, \quad i = 1, \ldots, q,$$

$$\|A_i\|_2 \le \|A\|_2, \qquad i = 0, \ldots, q.$$

This immediately gives

$$1 \le \|\mathcal{P}\|_2 \le c_n = \frac{1}{2} + [\log_2 n]. \tag{25}$$

Next, we show that the operator $\mathcal{P}$ has spectral norm unbounded in n. In fact, $\|\mathcal{P}\|_2 \sim \log n$ as $n \to \infty$, so the $\sim \log_2 n$ estimate (25) is about the best possible. (Compare also related results of Mathias [8] which are slightly sharper. We bring our proof because of its simplicity.) Kahan [6], [7] has given an example of upper triangular matrix $Z \in \mathbf{C}^{n\times n}$, with zero diagonal, for which[2]

$$\frac{\|Z - Z^*\|_2}{\|Z + Z^*\|_2} > \frac{2}{\pi}\left(\log n + \frac{1}{4} - \log 2 + \frac{1}{2n}\right). \tag{26}$$

Since $\mathcal{P}(Z + Z^*) = Z$, we obtain

$$\frac{\|\mathcal{P}(Z + Z^*)\|_2}{\|Z + Z^*\|_2} > \frac{1}{\pi}(\log n + o(1)), \tag{27}$$

which shows that the $\log n$-like bound for $\|\mathcal{P}\|_2$ is almost attainable. Kahan's matrix Z reads

$$Z = \mathbf{i}\begin{bmatrix} 0 & 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n-1} \\ & 0 & 1 & \frac{1}{2} & \cdots & \frac{1}{n-2} \\ & & 0 & \ddots & \ddots & \vdots \\ & & & \ddots & \ddots & \frac{1}{2} \\ & & & & \ddots & 1 \\ & & & & & 0 \end{bmatrix} \in \mathbf{C}^{n\times n}, \qquad \mathbf{i}^2 = -1.$$

Even if we restrict the domain of $\mathcal{P}$ to the space of real symmetric matrices, $\|\mathcal{P}\|_2$ remains $\log n$-like. A real example $\mathcal{N}$ is constructed as follows. (A similar example appears independently in [4].)

Replace each Z_{ij} by the two by two block $|Z_{ij}|\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$. For example, in the case $n = 8$, we get

$$\mathcal{N} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & 0 & -1 & 0 & -\frac{1}{2} & 0 & -\frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & -1 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

[2] We thank the referee for pointing out an error in [6].

In fact, there is a permutation matrix P for which

$$\tilde{\mathcal{N}} = P^T \mathcal{N} P = \begin{bmatrix} \boldsymbol{\Theta} & C \\ -C & \boldsymbol{\Theta} \end{bmatrix}, \qquad C = -\mathbf{i}Z.$$

(Here $\boldsymbol{\Theta}$ denotes the null–matrix.)
Since $C^T = -\mathbf{i}Z^T$ and $Z^T = -Z^*$, we have

$$C - C^T = -\mathbf{i}(Z + Z^*),$$
$$C + C^T = -\mathbf{i}(Z - Z^*),$$

and then

$$\tilde{\mathcal{N}} + \tilde{\mathcal{N}}^T = \begin{bmatrix} \boldsymbol{\Theta} & -\mathbf{i}(Z + Z^*) \\ \mathbf{i}(Z + Z^*) & \boldsymbol{\Theta} \end{bmatrix},$$
$$\tilde{\mathcal{N}} - \tilde{\mathcal{N}}^T = \begin{bmatrix} \boldsymbol{\Theta} & -\mathbf{i}(Z - Z^*) \\ \mathbf{i}(Z - Z^*) & \boldsymbol{\Theta} \end{bmatrix}.$$

Finally,

$$\|\mathcal{N} + \mathcal{N}^T\|_2 = \|Z + Z^*\|_2,$$
$$\|\mathcal{N} - \mathcal{N}^T\|_2 = \|Z - Z^*\|_2,$$

and as in (27) we again obtain the lower $\log n$-like bound for $\|\mathcal{P}\|_2$.

According to (25), Theorem 2.1 applies if

$$\|S\|_2 < \frac{1}{4c_n} \tag{28}$$

and yields

$$\|\Gamma\|_2 \leq \frac{2\|S\|_2}{1 + \sqrt{1 - 4c_n\|S\|_2}}. \tag{29}$$

Remark 2.2. The quotient $\|\mathcal{P}(A)\|_2/\|A\|_2$ can sometimes be essentially less than $\|\mathcal{P}\|_2$. For instance, if $U = \operatorname{diag}(e^{\mathbf{i}\varphi_k})$ is unitary such that $U^*AU = |A|$, then

$$\|\mathcal{P}(A)\|_2 = \|\mathcal{P}(U|A|U^*)\|_2 = \|U\mathcal{P}(|A|)U^*\|_2$$
$$= \|\mathcal{P}(|A|)\|_2 \leq \||A|\|_2 = \|A\|_2,$$

because of the monotonicity of the spectral norm. (Here $|A|$ denotes the pointwise absolute value, $|A|_{ij} = |A_{ij}|$ for all i, j.)

The case of the perturbation of a general positive definite matrix H is easily reduced to that of Theorem 2.1.

COROLLARY 2.3. *Let $H \in \mathcal{H}$ be positive definite and R its Cholesky factor. Let δH be Hermitian perturbation such that $A = R^{-*}\delta H R^{-1}$ satisfies the conditions of Theorem* 2.1. *Then $H + \delta H$ is positive definite and its Cholesky factor reads*

$$R + \delta R = (I + \Gamma)R, \tag{30}$$

where $\delta R = \Gamma R$ and Γ is defined by (5), (6).

Note that by

$$\|A\|_2 \leq \|H^{-1}\|_2 \|\delta H\|_2, \tag{31}$$

the value $\|H^{-1}\|_2$ influences the estimate for $\|S\|_2$ and thus for $\|\Gamma\|_2$. This can lead to pesimistic estimates for relative or *floating point* errors. This is considered in the next section.

3. Floating point perturbations. If in Corollary 2.3 a relative perturbation bound is given by $\|\delta H\|_2 \le \varepsilon\|H\|_2$, then by (31) the key quantity $\|A\|_2$ is bounded by $\varepsilon\|H\|_2\|H^{-1}\|_2$. This condition may be pessimistic if we consider "floating point perturbations" of H, where $|\delta H_{ij}|$ is small with respect to $|H_{ij}|$ for every i, j, or, more generally, with respect to $\sqrt{H_{ii}H_{jj}}$.

THEOREM 3.1. *Let $H \in \mathcal{H}$ be positive definite, R its Cholesky factor, and δH Hermitian perturbation such that for all i, j*

$$|\delta H_{ij}| \le \varepsilon\sqrt{H_{ii}H_{jj}}, \tag{32}$$

where

$$\varepsilon < \frac{1}{4c_n^2 n\|B^{-1}\|_2} \tag{33}$$

and

$$B = \left[\frac{H_{ij}}{\sqrt{H_{ii}H_{jj}}}\right]. \tag{34}$$

Then $H + \delta H$ is positive definite and its Cholesky factor is given by

$$R + \delta R = (I + \Gamma)R,$$

with

$$\|\Gamma\|_2 \le \frac{2nc_n\varepsilon\|B^{-1}\|_2}{1 + \sqrt{1 - 4nc_n^2\varepsilon\|B^{-1}\|_2}}. \tag{35}$$

Proof. By (32) for any vector x, we have

$$\begin{aligned}|x^*\delta H x| &\le \varepsilon\sum_{i,j}\sqrt{H_{ii}H_{jj}}|x_i||x_j| = \varepsilon\left(\sum_i |\sqrt{H_{ii}}x_i|\right)^2 \\ &\le n\varepsilon\sum_i |\sqrt{H_{ii}}x_i|^2 \le n\varepsilon\|B^{-1}\|_2 x^*Hx,\end{aligned}$$

where we have used $\|x\|_2^2 \le \|B^{-1}\|_2 x^*Bx$. Setting here $x = R^{-1}y$, we obtain $|y^*Ay| \le n\varepsilon\|B^{-1}\|_2\|y\|_2^2$ and hence

$$\|A\|_2 \le n\varepsilon\|B^{-1}\|_2. \tag{36}$$

Recall,

$$A = R^{-*}\delta H R^{-1}.$$

Now by (33), we have

$$\|A\|_2 < \frac{1}{4c_n^2},$$

which implies (28). Then by (29) we obtain (35). □

It is instructive to compare the norm estimate (31) with the estimates (10), (8), (35). Corollary 2.3 and (28), (31) yield

$$\|\Gamma\|_2 \leq 2c_n\kappa(H)\frac{\|\delta H\|_2}{\|H\|_2}, \qquad \kappa(H) = \|H\|_2\|H^{-1}\|_2,$$

while (35) yields

$$\|\Gamma\|_2 \leq 2nc_n\|B^{-1}\|_2 \max_{i,j}\frac{|\delta H_{ij}|}{\sqrt{H_{ii}H_{jj}}}.$$

Now, $\|B^{-1}\|_2$ is newer much larger and can be much smaller than $\kappa(H)$ [9]. Note also that (8) and (10) give *local* estimates for the factor $R + \delta R$. This is important if R has very small and very large columns. The matrix B, obtained from H by diagonal scaling, was shown to be of crucial importance in floating point perturbation theory of the eigenvalue problem [2]. Our result shows that B controls the Cholesky decomposition in a similar sense.

Remark 3.2. If δH has at most p nonzero elements in any row (any column), then the factor n in the estimates of Theorem 3.1 can be replaced by p. This can be easily shown by using the estimates from [9] or by the Hadamard product technique.

Example 3.3. As an example, consider the matrix

$$H = \begin{bmatrix} \eta^2 & \eta \\ \eta & 2 \end{bmatrix}, \qquad \eta \ll 1.$$

Here

$$R = \begin{bmatrix} \eta & 1 \\ 0 & 1 \end{bmatrix}, \quad \|H\|_2\|H^{-1}\|_2 > \frac{2}{\eta^2}, \quad \|B^{-1}\|_2 = 2 + \sqrt{2}.$$

Any perturbation δH of the type (32) with $\varepsilon < 0.01$ causes perturbation $\delta R = \Gamma R$ with

$$\|\Gamma\|_2 < 20.5\varepsilon,$$

i.e., for all x,

$$\|\delta Rx\|_2 < 20.5\varepsilon\|Rx\|_2$$

holds. Note that this implies

$$\frac{\|\delta R\|_2}{\|R\|_2} < 20.5\varepsilon.$$

Furthermore, since $\lambda_{\min}(H) < \eta^2 \ll 1$, the application of Corollary 2.3 gives pessimistic estimate, because H is almost singular for small η.

Example 3.4. Let $H = R^TR$ be the Cholesky factorization of real positive definite $H \in \mathbf{R}^{n\times n}$ and $\tilde{R} = R + \delta R$ the computed Cholesky factor of H using floating point arithmetic with precision ε. Then (see [2, Lemma 4.5])

$$\tilde{R}^T\tilde{R} = H + \delta H,$$

where

$$|\delta H_{ij}| \leq (n+5)\varepsilon\sqrt{H_{ii}H_{jj}}, \qquad 1 \leq i,j \leq n.$$

Using Theorem 3.1, we obtain

$$\|\delta Rx\|_2 \leq \frac{2n(n+5)c_n\varepsilon\|B^{-1}\|_2}{1+\sqrt{1-4n(n+5)c_n^2\varepsilon\|B^{-1}\|_2}}\|Rx\|_2, \qquad x \in \mathbf{R}^n,$$

provided that

$$\|B^{-1}\|_2 \leq \frac{1}{4n(n+5)c_n^2\varepsilon}.$$

If, e.g., $\|B^{-1}\|_2 \approx 100$, $n = 1000$, $\varepsilon \approx 10^{-16}$ we have

$$\|\delta Rx\|_2 \leq \text{const} \cdot 10^{-7}\|Rx\|_2, \qquad x \in \mathbf{R}^n.$$

Thus, the singular values of $\tilde{R}$ and R, for example, coincide in seven digits at least, see [2, Thm. 2.14].

We now present another approach that permits a better treatment of perturbations preserving certain sparse structures. We also obtain some multiplicative estimates similar to those in [13].

Let

$$\delta H = \dot{H} \circ E \tag{37}$$

with the *relative error mask* matrix $\dot{H}$ and the *relative error* matrix E; see [3]. Here $\dot{H}_{ij}$ can be H_{ij} or $\sqrt{H_{ii}H_{jj}}$ or some other measure for the relative size of perturbation δH_{ij}, E_{ij} is the relative perturbation of H_{ij} (as measured relative to given $\dot{H}_{ij}$) and $\circ$ denotes the Hadamard (pointwise) matrix multiplication. Thus, $\delta H_{ij} = \dot{H}_{ij}E_{ij}$ for all i, j. It is reasonable to assume both $\dot{H}$ and E Hermitian. Let $\tilde{H} = H + \delta H$ be positive definite and

$$\tilde{H} = R^*R + \delta H = R^*(I + R^{-*}\delta H R^{-1})R = R^*K^2R,$$

with

$$K = \sqrt{I + R^{-*}\delta H R^{-1}}.$$

($\sqrt{\cdot}$ is the square root of a positive definite matrix.) If

$$K = QT (= T^*Q^*)$$

is the (unique) QR factorization of K with upper triangular T and $T_{ii} > 0$, $1 \leq i \leq n$, then

$$\tilde{H} = R^*T^*Q^*QTR = (TR)^*(TR) = \tilde{R}^*\tilde{R}$$

is the unique Cholesky factorization of $\tilde{H}$ with

$$\tilde{R} = TR = Q^*\sqrt{I + R^{-*}\delta H R^{-1}}R$$

and

$$\|T\|_2 = \|\sqrt{I + R^{-*}\delta H R^{-1}}\|_2 \leq \sqrt{1 + \|R^{-*}\delta H R^{-1}\|_2}.$$

If we define

$$\dot{H}_S = D^{-1}\dot{H}D^{-1}, \quad R_S = RD^{-1}, \quad H_S = R_S^* R_S,$$

with an arbitrary diagonal positive definite D, then

$$R^{-*}\delta H R^{-1} = R^{-*}(\dot{H} \circ E)R^{-1} = R_S^{-*}(\dot{H}_S \circ E)R_S^{-1}$$

holds. Hence

$$\begin{aligned}\|T\|_2^2 &\leq 1 + \|R_S^{-1}\|_2^2 \|\dot{H}_S \circ E\|_2 \\ &\leq 1 + \frac{\sqrt{\|\dot{H}_S \circ \overline{\dot{H}_S}\|_1}}{\lambda_{\min}(H_S)} \sqrt{\|E \circ \overline{E}\|_\infty},\end{aligned}$$

with

$$\overline{E} = (E^*)^\tau, \quad \overline{\dot{H}_S} = (\dot{H}_S^*)^\tau.$$

Here we have used the Cauchy–Schwarz inequality for the Hadamard product and the spectral norm. (See [5, p. 212]) If we define the condition number

$$\zeta(H, \dot{H}) = \min_D \frac{\sqrt{\|\dot{H}_S \circ \overline{\dot{H}_S}\|_1}}{\lambda_{\min}(H_S)},$$

then

$$\|T\|_2 \leq \sqrt{1 + \zeta(H, \dot{H})\sqrt{\|E \circ \overline{E}\|_\infty}}.$$

On the other side,

$$\begin{aligned}\sigma_{\min}(T) &= \sqrt{\lambda_{\min}(I + R^{-*}\delta H R^{-1})} \\ &\geq \sqrt{1 - \|R^{-*}\delta H R^{-1}\|_2} \geq \sqrt{1 - \zeta(H, \dot{H})\sqrt{\|E \circ \overline{E}\|_\infty}},\end{aligned}$$

provided that

$$\sqrt{\|E \circ \overline{E}\|_\infty} < \zeta(H, \dot{H})^{-1}. \tag{38}$$

Thus, we have proved the following multiplicative perturbation bound for the Cholesky factorization.

THEOREM 3.5. *Let $H \in \mathcal{H}$ be positive definite, R its Cholesky factor, and $\delta H = \dot{H} \circ E$ Hermitian perturbation satisfying* (38). *Then the Cholesky factor of $H + \delta H$ reads*

$$\tilde{R} = TR,$$

where

$$\sqrt{1 - \zeta(H, \dot{H})\sqrt{\|E \circ \overline{E}\|_\infty}} \leq \sigma_{\min}(T) \leq \sigma_{\max}(T) \leq \sqrt{1 + \zeta(H, \dot{H})\sqrt{\|E \circ \overline{E}\|_\infty}}.$$

Remark 3.6. We can bound $\zeta(H, H)$ by

$$\zeta(H, H) \leq \frac{1}{\lambda_{\min}(H_B)},$$

where H_B is balanced H (i.e., $H_B = D_B H D_B$, has unit rows and columns and D_B is the diagonal scaling). Thus, the error term is then approximatively given by

$$0.5\ \|H_B^{-1}\|_2 \sqrt{\|E \circ \overline{E}\|_\infty} + o(\|E \circ \overline{E}\|_\infty).$$

Next, we want to estimate relative errors in the elements of the Cholesky factor as functions of the relative errors in the elements of H. Since $\delta R = (T - I)R$, we immediately obtain

$$|\delta R| \leq |T - I||R|, \tag{39}$$

an estimate comparable with the one given by Sun [13]. Note, T is the Cholesky factor of $I + R^{-*}\delta H R^{-1}$ and the estimate

$$\|T - I\|_2 \leq \frac{2c_n \zeta(H, \dot{H}) \sqrt{\|E \circ \overline{E}\|_\infty}}{1 + \sqrt{1 - 4c_n^2 \zeta(H, \dot{H}) \sqrt{\|E \circ \overline{E}\|_\infty}}} \tag{40}$$

holds, provided that $\sqrt{\|E \circ \overline{E}\|_\infty} < 0.25 c_n^{-2} \zeta(H, \dot{H})^{-1}$. Since

$$\delta R_{ij} = \sum_{k=i}^{j} (T - I)_{ik} R_{kj}$$

and therefore for $1 \leq i \leq j \leq n$

$$|\delta R_{ij}| \leq \sqrt{\sum_{k=i}^{j} (T - I)_{ik}^2} \sqrt{\sum_{k=i}^{j} R_{kj}^2} \leq \|T - I\|_2 \sqrt{\sum_{k=i}^{j} R_{kj}^2},$$

we see that the magnitude of $|\delta R_{ij}|$ relative to

$$\sqrt{\sum_{k=i}^{j} R_{kj}^2}$$

is given by (40). Thus, the relative errors in $\tilde{R}_{ii}$, $1 \leq i \leq n$ are determined by the condition of H and bounded by $\|T - I\|_2$. The relative error in an element with magnitude of at least about the average of the corresponding column has nearly the same bound. If all elements in a column have about the same modulus, they all have relative errors about $o(\|T - I\|_2)$. Furthermore, if R is the optimally pivoted Cholesky factor then

$$|\delta R_{ij}| \leq \|T - I\|_2 |R_{ii}|, \qquad 1 \leq i \leq j \leq n.$$

4. Large perturbations. The estimates for Γ in §2 are asymptotically sharp, but their validity is restricted by a too strong condition (18). It is desirable to obtain estimates that are valid whenever the factorization (5) exists. The relation (5) can be written as

$$A = \Gamma + \Gamma^* + \Gamma^*\Gamma. \tag{41}$$

Taking the trace in (41), we obtain

$$\mathrm{Tr}A = 2\mathrm{Tr}\Gamma + \|\Gamma\|_F^2, \tag{42}$$

a central identity in our analysis. One can easily show that, with the proper notion of the angle, (42) is in fact the cosine theorem for a certain matrix triangle. By (6) and (42), we obtain

$$\|\Gamma\|_F \leq \sqrt{2n + \sqrt{n}\|A\|_F}.$$

Combining this with the local estimate

$$\|\Gamma\|_F \leq \sqrt{2}\|A\|_F$$

for $\|A\|_F \leq \frac{1}{2}$ (see (23)), we obtain a simple global estimate

$$\|\Gamma\|_F \leq \sqrt{8n + 2\sqrt{n}}\|A\|_F.$$

If A is positive or negative definite better estimates are available. We first prove the monotonicity property of the Cholesky factorization.

LEMMA 4.1. *Let K, H be positive definite with the Cholesky factorizations*

$$K = R_K^* R_K, \qquad H = R_H^* R_H, \tag{43}$$

and $H - K$ positive semidefinite ($K \preceq H$). Then

$$\|R_K R_H^{-1}\|_2 \leq 1. \tag{44}$$

Proof. From $R_K^* R_K \preceq R_H^* R_H$ we obtain $(R_K R_H^{-1})^*(R_K R_H^{-1}) \preceq I$ and (44) follows.[3] □

We now treat the case $I + A$ with A positive semidefinite.

LEMMA 4.2. *Let $\Gamma \in \mathcal{T}$ be a matrix satisfying* (5), *where $A \in \mathcal{H}$ is positive semidefinite. Then*

$$\frac{\|A\|_F}{1 + \sqrt{1 + \|A\|_2}} \leq \|\Gamma\|_F \leq \|\mathcal{P}(A)\|_F \leq \frac{1}{\sqrt{2}}\|A\|_F. \tag{45}$$

Proof. From (41), using the monotonicity of the diagonal, we obtain

$$A_{ii} \geq 2\Gamma_{ii} \geq 0, \qquad 1 \leq i \leq n, \tag{46}$$

[3] We thank the referee for improving the statement and the proof of this lemma.

and then

$$\sum_{i=1}^{n} \Gamma_{ii}^2 \leq \frac{1}{4} \sum_{i=1}^{n} A_{ii}^2. \tag{47}$$

On the other side, if we set $(I+\Gamma)^{(i)}$ for the $i \times i$ leading submatrix of $I+\Gamma$, then

$$\begin{bmatrix} \Gamma_{1,i+1} \\ \vdots \\ \Gamma_{i,i+1} \end{bmatrix} = \left((I+\Gamma)^{(i)}\right)^{-*} \begin{bmatrix} A_{1,i+1} \\ \vdots \\ A_{i,i+1} \end{bmatrix}. \tag{48}$$

Hence

$$\left\| \begin{bmatrix} \Gamma_{1,i+1} \\ \vdots \\ \Gamma_{i,i+1} \end{bmatrix} \right\| \leq \frac{1}{\sigma_{\min}((I+\Gamma)^{(i)})} \left\| \begin{bmatrix} A_{1,i+1} \\ \vdots \\ A_{i,i+1} \end{bmatrix} \right\|, \qquad 1 \leq i \leq n-1, \tag{49}$$

and

$$\|\Gamma - \operatorname{diag} \Gamma\|_F \leq \frac{1}{\sqrt{\lambda_{\min}(I+A)}} \frac{\|A - \operatorname{diag} A\|_F}{\sqrt{2}}, \tag{50}$$

where $\sigma_{\min}(\cdot)$ denotes the minimal singular value and $\lambda_{\min}(\cdot)$ the minimal eigenvalue of a matrix. This inequality is related to the simple fact that a diagonal perturbation of a diagonal matrix causes only a diagonal perturbation of its Cholesky factor. Finally, from (47) and (50), we obtain

$$\begin{aligned} \|\Gamma\|_F^2 &= \sum_{i=1}^{n} \Gamma_{ii}^2 + \|\Gamma - \operatorname{diag} \Gamma\|_F^2 \\ &\leq \|\mathcal{P}(\operatorname{diag} A)\|_F^2 + \|\mathcal{P}(A - \operatorname{diag} A)\|_F^2 = \|\mathcal{P}(A)\|_F^2 \end{aligned}$$

and the upper bound in (45) follows. To prove the lower bound, we use the majorization inequality between eigenvalues $\lambda_i(\cdot)$ and singular values $\sigma_i(\cdot)$ of a matrix (see, e.g., [11]) to obtain

$$\operatorname{Tr}(I+\Gamma) \leq \sum_{i=1}^{n} \sigma_i(I+\Gamma) = \sum_{i=1}^{n} \sqrt{1+\lambda_i(A)}$$

and then

$$\operatorname{Tr}\Gamma \leq \sum_{i=1}^{n} \frac{\lambda_i(A)}{1+\sqrt{1+\lambda_i(A)}}.$$

Now, (42) implies

$$\begin{aligned} \|\Gamma\|_F^2 &= \operatorname{Tr}A - 2\operatorname{Tr}\Gamma \geq \sum_{i=1}^{n} \left[\lambda_i(A) - \frac{2\lambda_i(A)}{1+\sqrt{1+\lambda_i(A)}} \right] \\ &= \sum_{i=1}^{n} \left[\frac{\lambda_i(A)}{1+\sqrt{1+\lambda_i(A)}} \right]^2 \geq \frac{\|A\|_F^2}{(1+\sqrt{1+\|A\|_2})^2} \end{aligned}$$

and the lower bound for $\|\Gamma\|_F$ follows. □

Remark 4.3. The estimates (48), (49) and (50) do not depend on the semidefiniteness of the perturbation. Thus, for example, if $\lambda_{\min}(A)$ is not too close to -1, $\lambda_{\min}(A) > -0.5$, say, then $\|\Gamma - \operatorname{diag} \Gamma\|_F \leq \|A - \operatorname{diag} A\|_F$.

Note added in proof. Ji Guang Sun [14] pointed out that the estimate (36) can be replaced by

$$\|A\|_F \leq \|\delta B\|_F \|B^{-1}\|_2, \qquad \delta B = \left[\frac{\delta H_{ij}}{\sqrt{H_{ii}H_{jj}}}\right].$$

Thus, using Theorem 2.1, we obtain

$$\|\Gamma\|_F \leq \frac{\sqrt{2}\|B^{-1}\|_2\|\delta B\|_F}{1+\sqrt{1-2\|B^{-1}\|_2\|\delta B\|_F}}$$

provided that

$$\|\delta B\|_F < \frac{1}{2\|B^{-1}\|_2}.$$

Acknowledgment. K. Veselić would like to thank Professor B. Parlett, at Berkeley, for his constructive criticism. The authors also thank Professors V. Hari, Zagreb, and Ji Guang Sun, Umeå, for their comments. Finally, we thank an anonymous referee for his very detailed and constructive report.

REFERENCES

[1] J. BARLOW AND J. DEMMEL, *Computing accurate eigensystems of scaled diagonally dominant matrices*, SIAM J. Numer. Anal., 27 (1990), pp. 762–791.
[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] Z. DRMAČ, *Computing the Singular and the Generalized Singular Values*, Ph.D. Thesis, Fernuniversität Hagen, Germany, 1994.
[4] A. EDELMAN AND W. F. MASCARENHAS, *On Parlett's matrix norm inequality for the Cholesky decomposition*, April 1993, Preprint.
[5] R. A. HORN AND CH. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, 1991, p. 212.
[6] W. KAHAN, *Every $n\times n$ Matrix Z with real spectrum satisfies $\|Z-Z^*\|_2 \leq \|Z+Z^*\|_2(\log_2 n+0.038)$*, Proc. Amer. Math. Soc., 19 (1973), pp. 235–241.
[7] ———, *Spectra of nearly Hermitian matrices*, Proc. Amer. Math. Soc., 48, 1 (1975), pp. 11–17.
[8] R. MATHIAS, *The Hadamard operator norm of a circulant and applications*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1152–1167.
[9] A. VAN DER SLUIS, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14-23.
[10] G. W. STEWART, *Perturbation bounds for the QR factorization of a matrix*, SIAM J. Numer. Anal., 14 (1977), pp. 509–518.
[11] G. W. STEWART AND J. G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1991.
[12] J. G. SUN, *Perturbation bounds for the Cholesky and QR factorizations*, BIT, 31 (1991), pp. 341–352.
[13] ———, *Componentwise perturbation bounds for some matrix decompositions*, BIT, 32 (1992), pp. 702–714.
[14] ———, private communication, 1993.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1333–1353, October 1994

019

TOWARDS A DIVIDE AND CONQUER ALGORITHM FOR THE REAL NONSYMMETRIC EIGENVALUE PROBLEM*

LOYCE ADAMS† AND PETER ARBENZ‡

Abstract. Theory is developed that could be used towards developing a divide and conquer algorithm for the nonsymmetric eigenvalue problem. The shortcomings of this theory and its application to the Hessenberg and nonsymmetric tridiagonal problems are discussed. The conclusion is made that the method may not be as promising as the divide and conquer methods for symmetric problems.

Key words. tridiagonal matrix, Hessenberg matrix, modified eigenvalue problem, divide and conquer algorithm

AMS subject classifications. 15A18, 65F15, 65W05

1. Introduction.

In 1981, Cuppen [7] introduced a divide and conquer algorithm for the computation of the spectral decomposition of real symmetric tridiagonal matrices. The algorithm, which was primarily designed for parallel computers, turned out to be faster than and comparably accurate to the well-known QR algorithm even on sequential computers [8]. We review this algorithm in §2. Similar divide and conquer algorithms have proven to be efficient for the bidiagonal singular value problem [18] and the unitary eigenvalue problem [14].

In this paper we investigate the advisability of using algorithms similar to those of Cuppen to solve the eigenvalue problem for real *nonsymmetric* matrices. In §3 we investigate how far the theory for symmetric rank-one modified eigenvalue problems carries over to the nonsymmetric case. This theory is then applied in §4 to two special cases of eminent interest: the eigenvalue problems for nonsymmetric tridiagonal matrices and for real (upper) Hessenberg matrices. In §5 we discuss stability issues that arise in the nonsymmetric case. Jessup [17] recently investigated the Hessenberg and nonsymmetric tridiagonal problems using a rank-two splitting assuming the matrices are diagonalizable.

The interest for the Hessenberg form stems from the fact that general matrices are transformed into this form before the QR algorithm is applied. We note here that the QR algorithm computes a Schur decomposition. In contrast, divide and conquer algorithms work with eigenvectors and possibly principal vectors. They therefore provide a spectral decomposition or, in the case of defective matrices, a partial spectral decomposition.

The nonsymmetric tridiagonal eigenvalue problem arises, for example, in connection with the nonsymmetric Lanczos algorithm [12, p. 502]. It is also possible to transform a general matrix into a similar tridiagonal matrix by means of (nonorthogonal) Householder elementary reflectors. This process however is not stable [25, p. 403]. Nevertheless, efforts have been made to stabilize it (see [10] and the references therein).

* Received by the editors September 30, 1991; accepted for publication (in revised form) August 6, 1993.

† Department of Applied Mathematics, University of Washington, Seattle, Washington 98195 (adams@amath.washington.edu). The work of this author was partially supported by National Science Foundation Presidential Young Investigator grant ASC-8858101 and Department of Energy grant DE-FG06-88ER25061 and was completed while the author was visiting the Institut für Wissenschaftliches Rechnen, Eidgenössische Technische Hochschule Zürich, Switzerland.

‡ Eidgenössische Technische Hochschule Zentrum, Institut für Wissenschaftliches Rechnen, 8092 Zürich, Switzerland (arbenz@inf.etthz.ch).

Divide and conquer algorithms have been successfully implemented for solving the symmetric tridiagonal eigenvalue problem or for the computation of the singular value decomposition (SVD) of bidiagonal matrices on shared memory multiprocessors [8], [18]. The implementation of these algorithms posed difficulties on distributed memory multiprocessors due to the excessive amount of data that must be transferred between processors [16]. For our investigation we assume therefore that our target machine is a shared memory multiprocessor computer.

2. Cuppen's divide and conquer algorithm. In this section we consider the problem of determining the spectral decomposition

$$\tilde{T} = \tilde{X}\tilde{\Lambda}\tilde{X}^T, \qquad \tilde{T}, \tilde{X}, \tilde{\Lambda} \in \mathbb{R}^{n\times n} \tag{2.1}$$

of the symmetric tridiagonal matrix

$$\tilde{T} = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Cuppen [7] introduced the decomposition

$$\begin{aligned} \tilde{T} = T + D &= \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + \beta_k \begin{pmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & \theta & 1 & & \\ & & & 1 & \theta & & \\ & & & & & 0 & \\ & & & & & & \ddots \\ & & & & & & & 0 \end{pmatrix} \\ &= T_1 \oplus T_2 + \theta\beta_k \mathbf{u}\mathbf{u}^T, \end{aligned} \tag{2.2}$$

where $\mathbf{u} = \mathbf{e}_k + \theta\mathbf{e}_{k+1}$ with $\theta = \pm 1$ and $\mathbf{e}_j$ denotes the jth unit vector. The introduction of the factor θ was indeed an idea of Dongarra and Sorensen [8] to avoid cancellation when forming the new diagonal elements of T_1 and T_2.

As the element of T at position $(k, k+1)$ vanishes, the computation of its spectral decomposition amounts to the solution of two *independent* symmetric tridiagonal eigenvalue problems for T_1 and T_2 of order k and $n-k$, respectively. With the spectral decompositions $T_i = X_i\Lambda_i X_i^T$, $i = 1, 2$, we get

$$\tilde{T} = X\left[\Lambda + \theta\beta_k \mathbf{v}\mathbf{v}^T\right] X^T, \tag{2.3}$$

where

$$X = X_1 \oplus X_2, \quad \Lambda = \Lambda_1 \oplus \Lambda_2, \quad \mathbf{v} = X^T\mathbf{u} = \begin{pmatrix} X_1^T\mathbf{e}_k \\ \theta X_2^T\mathbf{e}_1 \end{pmatrix}.$$

Thus, if we know Λ_1 and Λ_2, we obtain the eigenvalues and vectors of $\tilde{T}$ by computing the spectral decomposition of the matrix in square brackets, i.e., the spectral decomposition of a diagonal modified by a matrix of rank one.

As the matrices $(\tilde{T}-\lambda)\oplus 1$ and $(T-\lambda)\oplus(1+\theta\beta_k\mathbf{u}^T(T-\lambda)^{-1}\mathbf{u})$ for $\lambda\notin\sigma(T)$ are congruent [3]–[5], the eigenvalues and vectors of $\tilde{T}$ can be obtained from the ones of T by an investigation of the rational function [3], [6]–[8], [11], [13], [25]

$$\begin{aligned} f(\lambda) &= 1+\theta\beta_k\mathbf{u}^T(T-\lambda)^{-1}\mathbf{u} = 1+\theta\beta_k\mathbf{v}^T(\Lambda-\lambda)^{-1}\mathbf{v} \\ &= 1+\theta\beta_k\sum_{i=1}^{n}\frac{\nu_i^2}{\lambda_i-\lambda} = \frac{\det(\tilde{T}-\lambda)}{\det(T-\lambda)}, \qquad \mathbf{v}=\{\nu_i\}_{i=1}^n. \end{aligned} \tag{2.4}$$

f is called the (modified) Weinstein determinant [13], [24]. Alternatively, $f(\lambda)=0$ is called the secular equation [11]. By (2.4) one easily obtains the interlacing properties

$$\lambda_j\le\tilde{\lambda}_j\le\lambda_{j+1}, \quad 1\le j<n, \quad \lambda_n\le\tilde{\lambda}_n, \tag{2.5}$$

for positive $\theta\beta_k$. (Similar inequalities hold if $\theta\beta_k<0$.) Note that the λ_j do not appear ordered in the diagonal of Λ!

From (2.3) it is seen that the eigenvalue λ_j persists (with unchanged eigenvectors) if $\nu_j=0$. In this case we can *deflate*, i.e., remove, the corresponding jth row and column from $\Lambda+\theta\beta_k\mathbf{v}\mathbf{v}^T$. Moreover, if λ is an eigenvalue of T of multiplicity $m>1$, by an orthogonal similarity transformation of (2.3) the corresponding eigenvectors can be rotated such that at least $m-1$ of them are orthogonal to $\mathbf{u}$ [6]. This choice of eigenvectors introduces (at least) $m-1$ zero components in the vector $\mathbf{v}=X^T\mathbf{u}$ thus permitting further deflation. Of course, in a numerical context one must deal with the problem of "almost vanishing" ν_k's and "almost equal" eigenvalues. These issues are discussed in [8] and [18].

The result of the deflation process is a diagonal matrix $\Lambda'\in\mathbb{R}^{n'\times n'}$, $n'\le n$, which has only simple eigenvalues $\lambda'_j\in\sigma(T)$ and a vector $\mathbf{v}'$ whose elements are all nonzero. The eigenvalues of $\Lambda'+\theta\beta_k\mathbf{v}'\mathbf{v}'^T$ are the eigenvalues of $\tilde{T}$ that, at the same time, are *not* eigenvalues of T. The eigenvalues of Λ' and $\Lambda'+\theta\beta_k\mathbf{v}'\mathbf{v}'^T$ *strictly* interlace, i.e., satisfy formulae corresponding to (2.5) but with strict inequality signs.

The deflation process is of great importance for the success of the divide and conquer algorithm. First, the investigation of the Weinstein determinant is simplified since its poles coincide with the eigenvalues of Λ' and, second, n' is often considerably smaller than n [7], [8].

For the computation of the zeros of f, a quadratically and monotonically convergent root finder has been proposed by Bunch, Nielson, and Sorensen [6]. As soon as a zero $\tilde{\lambda}$ of f is found, a corresponding eigenvector $\tilde{\mathbf{x}}$ of $\tilde{T}$ can be computed by

$$\tilde{\mathbf{x}}=(\tilde{\lambda}-T)^{-1}\mathbf{u}=X(\tilde{\lambda}-\Lambda_1\oplus\tilde{\lambda}-\Lambda_2)^{-1}\mathbf{v}. \tag{2.6}$$

Calculation of eigenvectors by (2.6) can lead to a loss of orthogonality for close eigenvalues. Recently, work was done by Sorensen and Tang [20] and Gu and Eisenstat [15] to make this eigenvector calculation more stable.

For the complexity analysis for this algorithm, we assume that we have no deflation and that in the average s iteration steps are required to obtain one eigenvalue with the mentioned zerofinder. $f(\lambda)$ in (2.4) and its derivative are computed by

$$\begin{aligned} f(\lambda) &= 1+\theta\beta_k\mathbf{v}^T\mathbf{h}, \qquad \mathbf{h}=\mathbf{h}(\lambda)=(\Lambda-\lambda)^{-1}\mathbf{v}, \\ f'(\lambda) &= \theta\beta_k\mathbf{h}^T\mathbf{h}, \end{aligned} \tag{2.7}$$

in $6n+O(1)$ flops. Notice that after convergence, $h(\tilde{\lambda})$ is an (unnormalized) eigenvector of $\Lambda+\theta\beta_k\mathbf{v}\mathbf{v}^T$ corresponding to the just found eigenvalue $\tilde{\lambda}$. So, on the

average, the computation of one eigenvalue together with its normalized eigenvector costs $(6s+1)n+O(1)$ flops. Due to the special structure of X, the transformation of an eigenvector of $\Lambda+\theta\beta_k\mathbf{v}\mathbf{v}^T$ to one of $\tilde{T}$ needs n^2 flops. Altogether it costs

$$Z(n) = 2Z\left(\frac{n}{2}\right) + n^3 + (6s+1)n^2 + O(n) \cong \frac{4}{3}n^3 + 2(6s+1)n^2 + O(n) \tag{2.8}$$

to compute the complete spectral decomposition of $\tilde{T}$, provided that the split eigenvalue problems are solved with the same divide and conquer algorithm.

It is also possible to compute the eigenvectors of $\tilde{T}$ directly by inverse iteration as soon as its eigenvalues are known [9]. In this way the expensive back transformation could be saved and one would get an overall $O(n^2)$ algorithm. Unfortunately, inverse iteration does not yield sets of orthogonal eigenvectors in the presence of close eigenvalues. If the close eigenvalues were not deflated, a further orthogonalization may be necessary, hence bringing back the potentially $O(n^3)$ complexity.

3. Rank-one modified eigenvalue problems. Let $A \in \mathbb{R}^{n\times n}$ and consider the *modified eigenvalue problem*

$$\tilde{A}\mathbf{x} := (A+\mathbf{u}\mathbf{v}^T)\mathbf{x} = \lambda\mathbf{x}, \qquad \mathbf{u},\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}. \tag{3.1}$$

$\tilde{A}$ is called a rank-one modification of A.

We consider the problem of determining the eigenvalues and eigenvectors of $\tilde{A}$ assuming that we know all the eigenvalues and an orthonormal basis of the corresponding eigenspaces of A. Interestingly, we do not need to know the principal vectors of A.

3.1. Nonpersistent eigenvalues. An eigenvalue of $\tilde{A}$ is called *nonpersistent* if it is *not* an eigenvalue of A. Otherwise it is called persistent [24, p. 39]. We first deal with the former simpler case. We start with a result that can be formulated in a similar way for higher rank modifications [1].

PROPOSITION 3.1. *For $\lambda \notin \sigma(A)$, the spectrum of A, the matrices*

$$B := \begin{pmatrix} \lambda-\tilde{A} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \quad \text{and} \quad C := \begin{pmatrix} \lambda-A & \mathbf{0} \\ \mathbf{0}^T & \zeta(\lambda) \end{pmatrix} \tag{3.2}$$

with

$$\zeta(\lambda) := 1 - \mathbf{v}^T(\lambda-A)^{-1}\mathbf{u} \tag{3.3}$$

satisfy the equation

$$C = MBN, \tag{3.4}$$

where

$$M := \begin{pmatrix} I_n & \mathbf{0} \\ \mathbf{v}^T(A-\lambda)^{-1} & 1 \end{pmatrix}\begin{pmatrix} I_n & \mathbf{u} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} I_n & \mathbf{u} \\ \mathbf{v}^T(A-\lambda)^{-1} & \zeta(\lambda) \end{pmatrix} \tag{3.5a}$$

and

$$N := \begin{pmatrix} I_n & \mathbf{0} \\ \mathbf{v}^T & 1 \end{pmatrix}\begin{pmatrix} I_n & (A-\lambda)^{-1}\mathbf{u} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} I_n & (A-\lambda)^{-1}\mathbf{u} \\ \mathbf{v}^T & \zeta(\lambda) \end{pmatrix}. \tag{3.5b}$$

Proof. The proposition is proved by verification. □

Because $\det M = \det N = 1$ for all λ, we have

$$\det(\lambda - \tilde{A}) = \det(\lambda - A)\zeta(\lambda), \qquad \lambda \notin \sigma(A). \tag{3.6}$$

Thus, $\lambda \notin \sigma(A)$ is an eigenvalue of $\tilde{A}$ if and only if $\zeta(\lambda) = 0$.

Furthermore, as N is invertible, it provides a bijective mapping from the nullspace of C onto the nullspace of B. Analogously, M^H maps the nullspace of C^H one-to-one onto the nullspace of B^H.

If λ is not an eigenvalue of A, a vector $\mathbf{x}$ in the nullspace of C must have the form

$$\mathbf{x} = \begin{pmatrix} \mathbf{0} \\ \xi \end{pmatrix}, \qquad \xi \in \mathbb{C} \setminus \{0\}.$$

Therefore,

$$N\mathbf{x} = \begin{pmatrix} (A - \lambda)^{-1}\mathbf{u}\xi \\ \zeta(\lambda)\xi \end{pmatrix} = \xi \begin{pmatrix} (A - \lambda)^{-1}\mathbf{u} \\ 0 \end{pmatrix}$$

is a vector spanning the nullspace of B. From the form of $N\mathbf{x}$ it is clear that

$$(A - \lambda)^{-1}\mathbf{u} \tag{3.7a}$$

is a right eigenvector of $\tilde{A}$ corresponding to λ. Similarly one shows that

$$(A^H - \bar{\lambda})^{-1}\mathbf{v} \tag{3.7b}$$

is a left eigenvector of $\tilde{A}$ corresponding to λ.

Note that the *geometric* multiplicity of a nonpersistent eigenvalue of $\tilde{A}$ cannot exceed one. The *algebraic* multiplicity, however, can exceed one, as is demonstrated by the following example.

Example 3.1. Choose

$$A = \begin{pmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix} \quad \text{and} \quad \tilde{A} = \begin{pmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}.$$

Then, $\sigma(A) = \{e^{2\pi ij/n}, j = 0, \ldots, n-1\}$ and $\sigma(\tilde{A}) = \{0\}$. Equation (3.1) holds with

$$\mathbf{u} = \mathbf{e}_1, \qquad \mathbf{v} = -\mathbf{e}_n.$$

Thus, $\zeta(\lambda)$ becomes

$$\zeta(\lambda) = \frac{\det(\lambda - \tilde{A})}{\det(\lambda - A)} = \frac{\lambda^n}{\lambda^n - 1}.$$

$\lambda = 0$ is an (algebraically) n-fold eigenvalue of $\tilde{A}$. Its one-dimensional right eigenspace is spanned by (cf. (3.7a))

$$(A - 0)^{-1}\mathbf{u} = A^T\mathbf{u} = \mathbf{e}_n$$

and its left eigenspace by (cf. (3.7b))

$$(A^H - 0)^{-1}\mathbf{v} = A\mathbf{v} = \mathbf{e}_1.$$

Also to get right principal vectors corresponding to λ we can, at least theoretically, proceed in the following bootstrapping manner [1]: Let us define $\mathcal{N}_j := \mathcal{N}((\tilde{A} - \lambda)^j)$. We recursively construct vectors $\mathbf{q}_j$ that span $\mathcal{N}_j \ominus \mathcal{N}_{j-1} = \mathcal{N}_j \cap \mathcal{N}_{j-1}^{\perp}$ such that $\mathcal{N}_k$ is spanned by $\mathbf{q}_1, \ldots, \mathbf{q}_k$. This construction breaks down as soon as $k = d$, where d is the smallest integer for which $\mathcal{N}_{d+1} = \mathcal{N}_d$. Notice that d is the algebraic multiplicity of λ.

Formula (3.7a) yields a basis vector of $\mathcal{N}_1$. If $\mathbf{x} \in \mathcal{N}_{j+1} \ominus \mathcal{N}_j$, $j \geq 1$, then clearly $\mathbf{y} := (\tilde{A} - \lambda)\mathbf{x} \in \mathcal{N}_j \ominus \mathcal{N}_{j-1}$ and $\mathbf{y} \in \mathcal{R}(\tilde{A} - \lambda)$. If, on the other hand, $\mathbf{y} \in (\mathcal{N}_j \ominus \mathcal{N}_{j-1}) \cap \mathcal{R}(\tilde{A} - \lambda)$, then $(\tilde{A} - \lambda)^+\mathbf{y} \in \mathcal{N}_{j+1} \ominus \mathcal{N}_j$. In fact, $(\tilde{A} - \lambda)(\tilde{A} - \lambda)^+$ is the orthogonal projector onto $\mathcal{R}(\tilde{A} - \lambda)$ [12, p. 423]. So, $(\tilde{A} - \lambda)(\tilde{A} - \lambda)^+\mathbf{y} = \mathbf{y}$. Therefore,

$$(\tilde{A} - \lambda)^+ : (\mathcal{N}_j \ominus \mathcal{N}_{j-1}) \cap \mathcal{R}(\tilde{A} - \lambda) \longrightarrow \mathcal{N}_{j+1} \ominus \mathcal{N}_j$$

is bijective.

So, provided that $\mathbf{q}_j \in (\mathcal{N}_j \ominus \mathcal{N}_{j-1}) \cap \mathcal{R}(\tilde{A} - \lambda)$, the nonzero vector

$$\mathbf{q}_{j+1} := (\tilde{A} - \lambda)^+\mathbf{q}_j \tag{3.8}$$

spans $\mathcal{N}_{j+1} \ominus \mathcal{N}_j$.

Because $\mathcal{R}(\tilde{A} - \lambda) = \mathcal{N}(\tilde{A}^H - \bar{\lambda})^{\perp}$, the vector $\mathbf{q}_j$ is in $\mathcal{R}(\tilde{A} - \lambda)$ if and only if

$$\mathbf{s}_1{}^H\mathbf{q}_j = 0, \tag{3.9}$$

where $\mathbf{s}_1$ is the left eigenvector of A corresponding to λ.

Using (3.4) in the form $B^+ = NC^+M$, (3.8) becomes

$$\mathbf{q}_{j+1} = (\lambda - A)^{-1}\mathbf{q}_j. \tag{3.10}$$

So, starting with $\mathbf{q}_1$, we construct the principal vectors $\mathbf{q}_j$ of grade $j > 1$ using (3.10) until (3.9) fails to be true.

Example 3.1 (continued). With the above we can set

$$\mathbf{q}_1 = \mathbf{e}_n\,, \qquad \mathbf{s}_1 = \mathbf{e}_1.$$

As $\mathbf{s}_1^H\mathbf{q}_1 = 0$, there is a right principal vector of grade two, obtainable by (3.10)

$$\mathbf{q}_2 = (A - \lambda)^{-1}\mathbf{e}_n = (A - 0)^{-1}\mathbf{e}_n = A^T\mathbf{e}_n = \mathbf{e}_{n-1}\,.$$

We continue as described until we get to $\mathbf{q}_n = \mathbf{e}_1$. At this point

$$\mathbf{s}_1{}^H\mathbf{q}_n = 1 \neq 0$$

and the process breaks down.

Left principal vectors can be obtained in a similar way, starting with $\mathbf{s}_1$.

3.2. Persistent eigenvalues. Proposition 3.1 can be generalized to the case where λ is an eigenvalue of the unmodified matrix A, i.e., a persistent eigenvalue.

PROPOSITION 3.2. *Let λ be an eigenvalue of A with geometric multiplicity m. Let $W_L, W_R \in \mathbb{C}^{n\times m}$ be matrices with orthonormal columns spanning $\mathcal{N}(A^H - \bar{\lambda})$ and $\mathcal{N}(A-\lambda)$, respectively. Then the matrices*

$$B = \begin{pmatrix} \lambda - A - \mathbf{u}\mathbf{v}^T & \mathbf{0} & O_{n\times m} \\ \mathbf{0}^T & 1 & \mathbf{0}^T \\ O_{m\times n} & \mathbf{0} & O_m \end{pmatrix} \tag{3.11a}$$

and

$$C = \begin{pmatrix} \lambda - A & \mathbf{0} & O_{n\times m} \\ \mathbf{0}^T & 1 - \mathbf{v}^T(\lambda - A)^+\mathbf{u} & \mathbf{v}^T W_R \\ O_{m\times n} & W_L^H\mathbf{u} & O_m \end{pmatrix} \tag{3.11b}$$

satisfy the equation

$$C = MBN, \tag{3.12}$$

where

$$M = \begin{pmatrix} I_n - W_L W_L^H & (I - W_L W_L^H)\mathbf{u} & W_L \\ -\mathbf{v}^T(\lambda - A)^+ & 1 - \mathbf{v}^T(\lambda - A)^+\mathbf{u} & O_{n\times m} \\ W_L^H & W_L^H\mathbf{u} & O_m \end{pmatrix} \tag{3.13a}$$

and, similarly,

$$N = \begin{pmatrix} I_n - W_R W_R^H & -(\lambda - A)^+\mathbf{u} & W_R \\ \mathbf{v}^T(I_n - W_R W_R^H) & 1 - \mathbf{v}^T(\lambda - A)^+\mathbf{u} & \mathbf{v}^T W_R \\ W_R^H & O_{m\times n} & O_m \end{pmatrix}. \tag{3.13b}$$

Proof. The proposition is proved by verification. □

The submatrix

$$Z_e(\lambda) = \begin{pmatrix} 1 - \mathbf{v}^T(\lambda - A)^+\mathbf{u} & \mathbf{v}^T W_R \\ W_L^H\mathbf{u} & 0 \end{pmatrix} \in \mathbb{C}^{(m+1)\times(m+1)} \tag{3.14}$$

of C is a generalization of $\zeta(\lambda)$ in (3.3). $(\lambda - A)^+$ is the Moore–Penrose generalized inverse of $\lambda - A$. In the real symmetric case, $\zeta(\lambda)$ and $Z_e(\lambda)$ are called Weinstein determinant and extended Weinstein matrix [24].

Proposition 3.2 gives a means for computing the eigenspaces of $\tilde{A}$ corresponding to a persistent eigenvalue λ. With (3.12) it is easily verified that

$$\mathcal{N}(\tilde{A} - \lambda) = \left\{ \omega(A-\lambda)^+\mathbf{u} + W_R\mathbf{w} \,\middle|\, \begin{pmatrix} \omega \\ \mathbf{w} \end{pmatrix} \in \mathcal{N}(Z_e(\lambda)) \right\} \tag{3.15a}$$

and

$$\mathcal{N}(\tilde{A}^H - \bar{\lambda}) = \left\{ \omega(A^H - \bar{\lambda})^+\mathbf{v} + W_L\mathbf{w} \,\middle|\, \begin{pmatrix} \omega \\ \mathbf{w} \end{pmatrix} \in \mathcal{N}(Z_e^H(\lambda)) \right\}. \tag{3.15b}$$

Principal vectors corresponding to persistent eigenvalues can be computed in a way similar to those corresponding to nonpersistent eigenvalues. The main difference is that the spaces $\mathcal{N}_j \ominus \mathcal{N}_{j-1}$ can have dimensions greater than one [1].

As in the symmetric case, it is natural to choose the matrices W_L and W_R such that $W_L^H \mathbf{u} = \pi \mathbf{e}_1$, $|\pi| = \|\mathbf{u}\|$, and $W_R^H \mathbf{v} = \phi \mathbf{e}_1$, $|\phi| = \|\mathbf{v}\|$, respectively. This means that columns 2 to m of W_L and W_R are orthogonal to $\mathbf{u}$ and $\mathbf{v}$, respectively, and thus are left and right eigenvectors of both A and $\tilde{A}$. The extended Weinstein matrix then has the form

$$Z_e(\lambda) = \hat{Z}_e(\lambda) \oplus O_{m-1}, \tag{3.16}$$

where

$$\hat{Z}_e(\lambda) = \begin{pmatrix} \sigma & \phi \\ \pi & 0 \end{pmatrix}, \qquad \sigma = 1 - \mathbf{v}^T (\lambda - A)^+ \mathbf{u}. \tag{3.17}$$

To determine whether there are further eigenvectors corresponding to λ, we must investigate $\hat{Z}_e$. We have five cases to consider.

1. If $\pi\phi \neq 0$ then rank $\hat{Z}_e(\lambda) = 2$ and we get no more right or left eigenvectors, and $\dim \mathcal{N}(\tilde{A} - \lambda) = m - 1$.

2. If $\phi = 0, \pi \neq 0$ then $\operatorname{rank} \hat{Z}_e(\lambda) = 1$. In this case, $(0\ 1)^T$ spans $\mathcal{N}(\hat{Z}_e(\lambda))$ and $\mathbf{x}_m = W_R \mathbf{e}_1$, the first column of W_R, is the mth right eigenvector and $\dim \mathcal{N}(\tilde{A} - \lambda) = m$. Likewise, up to normalization, $\mathbf{y}_m = (A^H - \bar{\lambda})^+ \mathbf{v} - (\bar{\sigma}/\bar{\pi}) W_L \mathbf{e}_1$ is the mth left eigenvector.

3. If $\phi \neq 0$ and $\pi = 0$, then the rank of $\hat{Z}_e(\lambda)$ is one. Up to normalization, $\mathbf{x}_m = (A - \lambda)^+ \mathbf{u} - (\sigma/\phi) W_R \mathbf{e}_1$ is the mth right eigenvector, and $\dim \mathcal{N}(\tilde{A} - \lambda) = m$. Likewise, $\mathbf{y}_m = W_L \mathbf{e}_1$ is the mth left eigenvector.

4. If $\phi = \pi = 0$, $\sigma \neq 0$, $\hat{Z}_e(\lambda)$ has rank one. We easily find that $\mathbf{x}_m = W_R \mathbf{e}_1$ and $\mathbf{y}_m = W_L \mathbf{e}_1$ are the mth right and left eigenvectors of $\tilde{A}$, respectively, and $\dim \mathcal{N}(\tilde{A} - \lambda) = m$.

5. If $\phi = \pi = \sigma = 0$, $\hat{Z}_e$ equals the 2×2 zero matrix and has a null space spanned by $(1,0)^T$ and $(0,1)^T$. Hence the mth eigenvectors are given as in case 4. The $m+1$st right and left eigenvectors are given by $\mathbf{x}_{m+1} = (A - \lambda)^+ \mathbf{u} / \|(A - \lambda)^+ \mathbf{u}\|$ and $\mathbf{y}_{m+1} = (A^H - \bar{\lambda})^+ \mathbf{v} / \|(A^H - \bar{\lambda})^+ \mathbf{v}\|$, respectively, and $\dim \mathcal{N}(\tilde{A} - \lambda) = m + 1$.

Clearly, for *any* λ the dimensions of $\mathcal{N}(\tilde{A} - \lambda)$ and $\mathcal{N}(A - \lambda)$ differ by at most one.

We also note that the left and right eigenspaces of $\tilde{A}$ corresponding to λ found by the above process produces vectors that are orthonormal, and hence further techniques do not have to be used to get an orthonormal $\tilde{W}_L$ and $\tilde{W}_R$.

3.3. Deflation. After having determined left and right eigenspaces and computed the extended Weinstein matrix for all $\lambda \in \sigma(A)$, one may wonder if this information can be used to reduce the cost of the search of the remaining eigenvalues $\lambda \in \sigma(\tilde{A}) \setminus \sigma(A)$. We know that left and right eigenvectors corresponding to different eigenvalues are orthogonal, but we see no way to reduce the order of the given matrix eigenvalue problem for the computation of the remaining eigenvalues as can be done in the symmetric eigenvalue problem in the deflation process. Notice that it is not even possible to determine the cardinality of $\sigma(\tilde{A}) \setminus \sigma(A)$. To be able to deflate, we must assume that A is diagonalizable. This assumption is of course true for symmetric matrices. If $AX = X\Lambda$, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, we can deflate in the proper meaning of the word, i.e., we can transform the original modified eigenvalue problem into a modified eigenvalue problem of lower order, containing all the information to compute the eigenvalues in $\sigma(\tilde{A}) \setminus \sigma(A)$ and corresponding eigenvectors. Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$

and $Y := (X^{-1})^H = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$. Then

$$A = X\Lambda Y^H = \sum_{i=1}^{n} \lambda_i \mathbf{x}_i \mathbf{y}_i^H. \tag{3.18}$$

Bases for left and right eigenspaces corresponding to an m-fold eigenvalue λ_k, say, are obtained by selecting those vectors $\mathbf{x}_{i_j}$ and $\mathbf{y}_{i_j}$ for which $\lambda_{i_j} = \lambda_k, j = 1, \ldots, m$. Notice that diagonalizable matrices have no degenerate eigenvalues.

Using the spectral decomposition (3.18), the modified matrix $\tilde{A}$ becomes

$$\tilde{A} = X(\Lambda + \hat{\mathbf{u}}\hat{\mathbf{v}}^H)Y^H, \qquad \hat{\mathbf{u}} = Y^H\mathbf{u}, \quad \hat{\mathbf{v}} = X^H\mathbf{v}. \tag{3.19}$$

If $\Lambda + \hat{\mathbf{u}}\hat{\mathbf{v}}^H$ has the spectral decomposition $\Lambda + \hat{\mathbf{u}}\hat{\mathbf{v}}^H = \hat{X}\tilde{\Lambda}\hat{Y}^H$, then

$$\tilde{A} = \tilde{X}\tilde{\Lambda}\tilde{Y}^H, \qquad \tilde{X} = X\hat{X}, \quad \tilde{Y} = Y\hat{Y}. \tag{3.20}$$

So we must investigate the eigenvalue problem

$$(\Lambda + \hat{\mathbf{u}}\hat{\mathbf{v}}^H)\mathbf{w} = \lambda\mathbf{w} \tag{3.21}$$

to get the spectral decomposition of $\tilde{A}$. As left and right eigenvectors of Λ corresponding to the above eigenvalue λ_k are given by the unit vectors $\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_m}$, the extended Weinstein matrix in λ_k becomes

$$Z_e(\lambda_k) = \begin{pmatrix} 1 - \sum_{\lambda_i \neq \lambda_k} \frac{\bar{\hat{v}}_i \hat{u}_i}{\lambda_k - \lambda_i} & \bar{\hat{v}}_{i_1} \cdots \bar{\hat{v}}_{i_m} \\ \hat{u}_{i_1} & \\ \vdots & \mathbf{O} \\ \hat{u}_{i_m} & \end{pmatrix}.$$

We now chose a unitary $Q = Q(\lambda_k) \in \mathbb{C}^{m\times m}$ such that

$$Q^H \begin{pmatrix} \hat{v}_{i_1} \\ \vdots \\ \hat{v}_{i_m} \end{pmatrix} = \overline{\phi(\lambda_k)}\mathbf{e}_1, \qquad |\phi(\lambda_k)|^2 = \sum_{j=1}^{m} |\hat{v}_{i_j}|^2,$$

and replace columns $i_1, \ldots, i_m$ in X and Y by

$$[\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_m}]Q, \qquad [\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_m}]Q.$$

As we transformed left and right eigenvectors by the same Q, the relation $Y = (X^{-1})^H$ remains valid. Therefore, (3.18) and (3.19) still hold with the modified X and Y, but now $\hat{v}_{i_j} = 0$, $j = 2, \ldots, m$. So, $\mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_m}$ are right eigenvectors of $\tilde{A}$ corresponding to the eigenvalue λ_k. If $\phi(\lambda_k) = 0$, $\mathbf{x}_{i_1}$ is a further right eigenvector corresponding to this same eigenvalue λ_k.

We can proceed in this way for every eigenvalue of A and finally permute the columns of X and Y such that $\hat{\mathbf{v}} = X^H\mathbf{v}$ has n' leading nonzero elements followed by $n - n'$ zeros, $\hat{\mathbf{v}}^H = (\hat{\mathbf{v}}_1^H, \mathbf{0}^T)$. Splitting $\hat{\mathbf{u}} = Y^H\mathbf{u}$ in the same way, $\hat{\mathbf{u}}^H = (\hat{\mathbf{u}}_1^H, \hat{\mathbf{u}}_2^H)$, we obtain

$$\tilde{A} = [X_1, X_2] \begin{pmatrix} \Lambda_1 + \hat{\mathbf{u}}_1\hat{\mathbf{v}}_1^H & 0 \\ \hat{\mathbf{u}}_2\hat{\mathbf{v}}_1^H & \Lambda_2 \end{pmatrix} [Y_1, Y_2]^H, \tag{3.22}$$

where $X_1, Y_1 \in \mathbb{C}^{n\times n'}$ and $X_2, Y_2 \in \mathbb{C}^{n\times(n-n')}$. Note, that Λ_1 has only simple eigenvalues. The matrix in parenthesis in (3.22) is a rank-one modified diagonal matrix, the secular equation of which is

$$\zeta(\lambda) = 1 - \hat{\mathbf{v}}^H(\lambda - \Lambda)^{-1}\hat{\mathbf{u}} = 1 - \sum_{i=1}^{n'} \frac{\bar{\hat{v}}_i \hat{u}_i}{\lambda - \lambda_i} = 0. \tag{3.23}$$

The zeros of this function yield the nonpersistent eigenvalues. If $\zeta(\lambda) = 0$, $\lambda \notin \sigma(\Lambda)$, by (3.7a) and (3.22)

$$X(\Lambda - \lambda)^{-1}\hat{\mathbf{u}}$$

is a corresponding eigenvector of $\tilde{A}$.

It is however possible that $\zeta(\lambda) = 0$ for a $\lambda \in \sigma(A)$. Considering again λ_k from above, this can be the case if $|\hat{v}_{i_1}| = |\phi(\lambda_k)| > 0$ but $\hat{u}_{i_j} = 0$, $j = 1, \ldots, m$. This means that we are in case 3 of the analysis of $\hat{Z}_e$. A second situation occurs when $\zeta(\lambda_k)$ and all the $\hat{v}_{i_j}$ vanish, but one of the $\hat{u}_{i_j}$ is nonzero. In this case λ_k is degenerate!

Example 3.2. Let

$$\tilde{A} = \Lambda + \mathbf{u}\mathbf{v} = \begin{pmatrix} 1 & & \\ & 3 & \\ & & 2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (2,1,0) = \begin{pmatrix} 3 & 1 & 0 \\ 2 & 4 & 0 \\ 2 & 1 & 2 \end{pmatrix}.$$

Then, $W_L(2) = W_R(2) = \operatorname{span}(\mathbf{e}_3)$. After deflation of the eigenvalue two we get

$$\zeta(\lambda) = 1 - \frac{2}{\lambda - 1} - \frac{1}{\lambda - 3},$$

which vanishes when $\lambda = 2$. Notice that

$$Z_e(2) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

$\lambda = 2$ is a defective eigenvalue of $\tilde{A}$ of algebraic multiplicity 2.

Given $\tilde{A} = A + \mathbf{u}\mathbf{v}^T$, we have presented the theory for finding each eigenvalue of $\tilde{A}$ with its associated geometric multiplicity and orthonormal bases for its left and right eigenspaces. This was done by knowing the exact same information about matrix A. Hence, in theory, a recursive algorithm for the nonsymmetric eigenvalue problem can be specified.

In addition, we gave the theory for finding the left and right principal vectors and algebraic multiplicity of each eigenvalue of $\tilde{A}$. This was done by knowing the left and right eigenvectors of A and $\tilde{A}$ corresponding to the eigenvalue. We also showed how to deflate in the case where λ is both an eigenvalue of A and $\tilde{A}$ and A is diagonalizable.

4. Application of the theory. We now apply the theory of the last section to the nonsymmetric tridiagonal eigenvalue problem. At the end of this section we will make a few remarks on the Hessenberg eigenvalue problem. We discuss the details as well as the hurdles involved in realizing a recursive algorithm for these problems. We do not further analyze the issues of deflation or the calculation of the principal vectors.

4.1. The tridiagonal eigenvalue problem. We consider the eigenvalue problem

$$\tilde{T}\mathbf{x} = \lambda\mathbf{x}, \tag{4.1}$$

where $\tilde{T}$ is the nonsymmetric irreducible tridiagonal matrix

$$\tilde{T} = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_2 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \gamma_n & \alpha_n \end{pmatrix}, \quad \beta_i \neq 0, \quad \gamma_i \neq 0. \tag{4.2}$$

We do not assume that $\tilde{T}$ is diagonalizable. Notice, that tridiagonal matrices with $\beta_i\gamma_{i+1} > 0$, for all i, are symmetrizable [25, p. 335]. We split $\tilde{T}$ into T and a rank-one modification. To zero out the elements of T at position β_k and γ_{k+1} simultaneously, we must choose the splitting vectors $\mathbf{u}$ and $\mathbf{v}$ appropriately:

$$\begin{aligned} \tilde{T} &= T + \begin{pmatrix} O_{k-1} & & & \\ & \frac{1}{\omega}\beta_k & \beta_k & \\ & \gamma_{k+1} & \omega\gamma_{k+1} & \\ & & & O_{n-k-1} \end{pmatrix}, \qquad \omega \neq 0 \\ &= \begin{pmatrix} T_1 & O \\ O & T_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{\omega}\beta_k\mathbf{e}_k \\ \gamma_{k+1}\mathbf{e}_1 \end{pmatrix} (\mathbf{e}_k^T, \omega\mathbf{e}_1^T) = T + \mathbf{u}\mathbf{v}^T, \end{aligned} \tag{4.3}$$

where

$$\mathbf{u} = \begin{pmatrix} \frac{1}{\omega}\beta_k\mathbf{e}_k \\ \gamma_{k+1}\mathbf{e}_1 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} \mathbf{e}_k \\ \omega\mathbf{e}_1 \end{pmatrix}.$$

To compute the nonpersistent eigenvalues of $\tilde{T}$ we must form the rational function $\zeta(\lambda)$ that appears in the secular equation (3.3). This function is given by

$$\begin{aligned} \zeta(\lambda) &= 1 + \mathbf{v}^T(T-\lambda)^{-1}\mathbf{u} \\ &= 1 + \tfrac{\beta_k}{\omega}\mathbf{e}_k^T(T_1-\lambda)^{-1}\mathbf{e}_k + \omega\gamma_{k+1}\mathbf{e}_1^T(T_2-\lambda)^{-1}\mathbf{e}_1 = 0. \end{aligned} \tag{4.4}$$

In case we use a Newton-type zerofinder for this purpose, the derivative of ζ is given by

$$\zeta'(\lambda) = -\mathbf{v}^T(T-\lambda)^{-2}\mathbf{u} = -\frac{\beta_k}{\omega}\mathbf{e}_k^T(T_1-\lambda)^{-2}\mathbf{e}_k - \omega\gamma_{k+1}\mathbf{e}_1^T(T_2-\lambda)^{-2}\mathbf{e}_1.$$

A divide and conquer algorithm for the tridiagonal eigenvalue problem based on the above theory has the following structure.

ALGORITHM TDC $(\tilde{T}, (\tilde{\lambda}_j, \tilde{m}_j, \tilde{W}_L^j, \tilde{W}_R^j, j = 1, 2, \ldots, \tilde{d}))$.
{This algorithm computes $\tilde{\lambda}_j$, $\tilde{m}_j$, $\tilde{W}_L^j \in \mathbb{C}^{n\times\tilde{m}_j}$, $\tilde{W}_R^j \in \mathbb{C}^{n\times\tilde{m}_j}$, $j = 1, \ldots, \tilde{d}$, the $\tilde{d}$ distinct eigenvalues $\tilde{\lambda}_j$ of $\tilde{T}$, their geometric multiplicity and bases for their associated left and right eigenspaces.}

1(a). Split $\tilde{T}$ according to (4.3).

1(b). Compute the eigenvalues and corresponding left and right eigenvectors of T_1 and T_2 by calls to TDC or another appropriate subroutine.

1(c). Merge the eigenvalues and eigenvectors of T_1 and T_2 to get the d distinct eigenvalues λ_j of T, their geometric multiplicities m_j, and bases $W_L^j \in \mathbb{C}^{n\times m_j}$ and $W_R^j \in \mathbb{C}^{n\times m_j}$ for the corresponding left and right eigenspaces.

2. Find the persistent eigenvalues $\lambda \in \sigma(\tilde{T}) \cap \sigma(T)$ together with the corresponding eigenvectors. This is an investigation of the extended Weinstein matrices $Z_e(\lambda_j)$, $j = 1, \ldots, d$, evaluated at the eigenvalues of T.

3. Find the nonpersistent eigenvalues $\lambda \in \sigma(\tilde{T}) \setminus \sigma(T)$ together with the corresponding eigenvectors. Here, a zero-finding procedure is needed to locate the zeros of the function ζ in (4.4).

The structure of the complete divide and conquer algorithm can be represented by a balanced binary tree with the task of solving the original problem being the root of the tree. The edges coming from the root represent the splitting of this problem into the two subproblems depicted as the descendent nodes of the root. Edges from these nodes (and intermediate nodes) toward the leaves represent recursive calls to TDC in step 1(b). The leaves of the tree represent the smallest subproblems that we are willing to solve by some black box eigensolver.

Once the eigenproblems at the leaves are solved, the "glue-back" problems at the next level up in the tree are solved by the merge operation in step 1(c) followed by steps 2 and 3. This "glue-back" operation is then repeated at each successive level in the tree until we finally solve the original problem at the root of the tree.

A shared memory parallel implementation that is identical in structure to that described in Dongarra and Sorensen [8] would apply to the nonsymmetric case as well. Briefly, p processors can simultaneously solve the eigenproblems at the leaf nodes. As we go up the tree, all "glue-back" problems at a given level can be solved simultaneously. Moreover, a given "glue-back" involves a rootfinding problem that can utilize multiple processors, so that at any given tree level, all $p \leq n$ processors can be kept busy. Hence we get more fine-grain parallelism and less coarse-grain parallelism as we go up the tree.

The memory needed to store the left and right eigenvectors of all these problems is four $n \times n$ arrays. Only vectors for two consecutive levels need to be stored at a given time, with two arrays allocated for each level. This is twice the storage the QR algorithm needs. The QR algorithm however computes Schur vectors. Notice that the four arrays occupy the shared memory in a parallel implementation.

We want to elaborate on the three steps of the algorithm.

1. Different strategies can be chosen to decide when to switch from TDC to another eigenvalue solver. In their divide and conquer algorithm for the symmetric tridiagonal eigenvalue problem, Dongarra and Sorensen [8] switch to the QR algorithm as soon as there are as many subproblems as these are available processors, i.e., after $\log_2(p)$ splits where p is the number of processors. For the unitary eigenvalue problem, Gragg and Reichel [14] (see also [2]) divide until they end up with 1×1 or 2×2 problems.

We can go either way. Switching presents the problem of how to solve the nonsymmetric tridiagonal eigenvalue problem at the leaves of the binary tree. A possibility is the LR algorithm [23, p. 319]. Because of the pivoting that must be performed to retain stability, this algorithm destroys the tridiagonal form. Therefore, we may just as well apply the nonsymmetric Hessenberg-QR algorithm here. This loss of the

tridiagonal form causes the need for much *private* memory! In the worst case, we must provide each processor with $O((n/p)^2)$ words of memory just to perform the LR or QR algorithm at the leaf level.

Eigenvectors for the leaf problems are best computed by inverse iteration with $T - \lambda_j$ to get the right eigenvectors and with $T^T - \bar{\lambda}_j$ for the left eigenvectors. They could be computed all together if inverse vector iteration is applied to the seven-diagonal Hermitian matrix

$$P^T \begin{pmatrix} O_n & T - \lambda_j \\ T^T - \bar{\lambda}_j & O_n \end{pmatrix} P.$$

Here, P denotes the odd-even permutation matrix.

The overall complexity of this step is $Cn^3 + O(n^2)$ unless we divide until we end up with 1×1 or 2×2 problems. We can however force the constant C to be as small as we like if we only divide sufficiently often, ending up with $O(n^2)$ complexity if we divide to 2×2 problems.

2. In step 2 we must compute the extended Weinstein matrices $Z_e(\lambda_j)$. W_L^j and W_R^j are known from step 1. The transformation of Z_e into the form (3.16) is easy as $m_j \leq 2$ because we are dealing with irreducible tridiagonal matrices.

If $\pi\phi \neq 0$ we must also compute

$$\sigma = 1 + \frac{\beta_k}{\omega} \mathbf{e}_k^T (T_1 - \lambda_j)^+ \mathbf{e}_k + \omega\gamma_{k+1} \mathbf{e}_1^T (T_2 - \lambda_j)^+ \mathbf{e}_1 = 0.$$

To circumvent computing the SVD we can proceed as follows. Let us consider the computation of $\mathbf{e}_k^T (T_1 - \lambda_j)^+ \mathbf{e}_k$ and set $\mathbf{w} = (\lambda_j - T_1)^+ \mathbf{e}_k$. Then $(\lambda_j - T_1)\mathbf{w} = P_{\mathcal{R}(\lambda_j - T_1)} \mathbf{e}_k$ with $\mathbf{w} \in \mathcal{N}(\lambda_j - T_1)^\perp$. Here, $P_{\mathcal{R}(\lambda_j - T_1)}$ is the orthogonal projector onto the range of $\lambda_j - T_1$. Let $\lambda_j - T_1 = QR$ be the QR factorization of $\lambda_j - T_1$. Then, the upper triangular matrix has the form [12, p. 374]

$$R = \begin{pmatrix} R_1 & \mathbf{r} \\ \mathbf{0}^T & 0 \end{pmatrix}.$$

Therefore,

$$\mathcal{R}(\lambda_j - T_1) = \mathcal{R}(QR) = \operatorname{span}(Q\mathbf{e}_k)^\perp, \qquad \mathcal{N}(\lambda_j - T_1) = \operatorname{span}\left(\begin{pmatrix} R_1^{-1}\mathbf{r} \\ -1 \end{pmatrix} \right).$$

Let

$$\mathbf{z} := \begin{pmatrix} R_1^{-1}\mathbf{r} \\ -1 \end{pmatrix} \Big/ \sqrt{1 + \|R_1^{-1}\mathbf{r}\|^2}$$

and $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$. Then $P_{\mathcal{R}(\lambda_j - T_1)} = [\mathbf{q}_1, \ldots, \mathbf{q}_{k-1}][\mathbf{q}_1, \ldots, \mathbf{q}_{k-1}]^*$ and

$$\mathbf{w} = (I_k - \mathbf{z}\mathbf{z}^H) \begin{pmatrix} R_1^{-1}[\mathbf{q}_1, \ldots, \mathbf{q}_{k-1}]^* \mathbf{e}_k \\ 0 \end{pmatrix}.$$

Each eigenvalue causes $O(n)$ operations as we are dealing with a tridiagonal T. As there are at most n matrices at a level in the problem tree, the overall complexity of this step is $O(n^2)$. Again, as T is tridiagonal, the auxiliary memory needed for the QR decomposition is only $O(n)$ words.

3. Find $\sigma(\tilde{T})\backslash\sigma(T)$. We must find the roots of the secular equation (4.4). Then the associated left and right eigenvectors associated with these roots are calculated by (3.7a) and (3.7b).

The main problem here is the location of the nonpersistent eigenvalues. They can be almost anywhere within the union of Gerschgorin disks. As we realistically cannot assume that the matrices are diagonalizable, we do not even know the number of eigenvalues we must look for unless we find all principal vectors corresponding to the persistent eigenvalues.

Jessup [17] discusses zerofinders that are based on the principle of the argument (Cauchy's integral formula). We doubt, as does Jessup, their practicability. The calculation of the integer integrals involved is too expensive although the integrals need not be computed very accurately.

A reasonable choice for a zero finder is Newton's iteration. To get good starting guesses for the iteration, eigenvalue bracketing procedures based on the matrix sign function could be applied [21]. The application of such procedures on the root level would, however, be so involved with respect to memory and complexity that it is certainly better to use such a method right from the start. Such methods compute Schur normal forms via unitary similarity transformations and are therefore intrinsically stable.

Forming the function ζ and its derivative costs $O(n)$ flops. If we assume that we need s iterations on the average to find a zero of ζ, where s is bounded with increasing n, then the overall complexity to find all nonpersistent eigenvalues would be $O(n^2)$. The computation of the corresponding eigenvectors with formula (3.7), with inverse iteration or with the procedure of step 2, would cost another $O(n^2)$ flops.

The described procedure gives us an overall $O(n^2)$ algorithm. As remarked in §2, a similar algorithm has been implemented by Gates [9] for the symmetric divide and conquer algorithm. The problem with the Gates algorithm was the possible loss of orthogonality among the eigenvectors that made reorthogonalization necessary and brought back the $O(n^3)$ complexity through the backdoor. Our algorithm has of course no better behaviour. So, the computed right eigenvectors corresponding to an eigenvalue λ must be orthogonalized against the left eigenvectors corresponding to nearby eigenvalues and vice versa.

We do not see an inexpensive way to prevent a parallel zero-finding procedure from finding the same zeros without communication among the processes. Recall that this was possible for the symmetric problem since we knew a priori distinct intervals that each contained one nonpersistent eigenvalue. The processors working on the same problem can of course communicate their information on already computed zeros (via shared memory). Convergence to equal eigenvalues cannot be excluded this way, however computation of all zeros is guaranteed. Hence, it is highly questionable whether a zerofinder can be found for the nonsymmetric tridiagonal problem that will parallelize as efficiently as its symmetric counterpart. If this difficulty is overcome, as well as the stability issues we discuss in §5, a shared memory parallel algorithm with the same structure as that given in Dongarra and Sorensen [8] would cost $O(n)$ with n processors.

4.2. The Hessenberg eigenvalue problem. A divide and conquer algorithm for the eigenvalue problem

$$\tilde{H}\mathbf{x} = \lambda\mathbf{x}, \tag{4.5}$$

where $\tilde{H}$ is an irreducible upper Hessenberg matrix proceeds formally much like the divide and conquer algorithm for the tridiagonal eigenvalue problem (4.1).

We define a matrix H to be

$$H := \tilde{H} - h_{k+1,k} \begin{pmatrix} \mathbf{p} \\ \mathbf{e}_1 \end{pmatrix} \begin{pmatrix} \mathbf{e}_k \\ \mathbf{q} \end{pmatrix}^T = \tilde{H} + \mathbf{u}\mathbf{v}^T, \tag{4.6}$$

where $\mathbf{p} \in \mathbb{R}^k$ and $\mathbf{q} \in \mathbb{R}^{n-k}$ are *arbitrary* vectors. Thus $\tilde{H}$ differs from H by a rank-one modification. H has a zero entry at position $(k+1, k)$. We write

$$H = \begin{pmatrix} H_1 & H_{12} \\ 0 & H_2 \end{pmatrix}, \tag{4.7}$$

where $H_1 \in \mathbb{R}^{k \times k}$ and $H_2 \in \mathbb{R}^{(n-k)\times(n-k)}$ are irreducible Hessenberg matrices and $H_{12} \in \mathbb{R}^{k\times(n-k)}$.

There are two essential differences that distinguish a Hessenberg divide and conquer from our previous algorithm TDC for the tridiagonal eigenvalue problem.

(i) As H is *not* the direct sum of H_1 and H_2, merging the eigenvalues and eigenvectors of H_1 and H_2 is not trivial anymore as descibed below. This concerns not only complexity. Let $(\lambda, \mathbf{x})$ be an eigenpair of H_1. Then, clearly, $(\lambda, \binom{\mathbf{x}}{0})$ is an eigenpair of H. On the other hand, if $(\lambda, \mathbf{y})$ is an eigenpair of H_2, then $(\lambda, \binom{\mathbf{w}}{\mathbf{y}})$ is an eigenpair of H if and only if

$$(\lambda - H_1)\mathbf{w} = H_{12}\mathbf{y}. \tag{4.8}$$

Equation (4.8) can be satisfied precisely if $H_{12}\mathbf{y} \in \mathcal{R}(\lambda - H_1)$. If $\lambda \notin \sigma(H_1)$ this is always the case. If $\lambda \in \sigma(H_1) \cap \sigma(H_2)$ a solution may or may not exist. In the latter case, λ is a degenerate eigenvalue.

(ii) The Weinstein determinant

$$\begin{aligned} \zeta(\lambda) &= 1 + \alpha \begin{pmatrix} \mathbf{e}_k \\ \mathbf{q} \end{pmatrix}^T (H - \lambda)^{-1} \begin{pmatrix} \mathbf{p} \\ \mathbf{e}_1 \end{pmatrix}, \qquad \alpha = h_{k+1,k}, \\ &= 1 + \alpha \left[\mathbf{e}_k^T (H_1 - \lambda)^{-1} \mathbf{p} - \mathbf{e}_k^T (H_1 - \lambda)^{-1} H_{12} (H_2 - \lambda)^{-1} \mathbf{e}_1 \right. \\ &\qquad \left. + \mathbf{q}^T (H_2 - \lambda)^{-1} \mathbf{e}_1 \right]. \end{aligned} \tag{4.9}$$

Note that the complexity to form $Z(\lambda)$ does not increase essentially if $\mathbf{p}$ and $\mathbf{q}$ are nonzero, because the vectors $(H_1^T - \lambda)^{-1}\mathbf{e}_k$ and $(H_2 - \lambda)^{-1}\mathbf{e}_1$ must be computed anyway!

Due to the Hessenberg structure of H, the evaluation of $\zeta(\lambda)$ costs $O(n^2)$ flops unless H is diagonalizable and we use formula (3.23). This is an order of magnitude more than in the tridiagonal case.

The main problem of the tridiagonal divide and conquer algorithm, that of finding the zeros in step 3 of the algorithm, remains unchanged.

The Hessenberg divide and conquer algorithm has of course higher complexity than the tridiagonal divide and conquer algorithm. The transformation into Hessenberg form is a natural intermediate step for the stable computation of the Schur normal form of a full matrix. A divide and conquer algorithm however cannot take advantage of this matrix structure. The stability of the algorithm cannot be increased by working with this form. We therefore see no reason for working with Hessenberg matrices in the context of a divide and conquer algorithm.

5. Stability. A principal difficulty in nonsymmetric eigenvalue problems is stability. Eigenvalues and eigenvectors may be very sensitive to small changes in the matrix elements. All eigensolvers, divide and conquer algorithms included, are experiencing this problem.

Let us consider the following example.

Example 5.1. Let

(5.1)
$$\tilde{T} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & \eta & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = T + \mathbf{u}\mathbf{v}^T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & \eta & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -\eta \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}^T.$$

This matrix has the spectrum $\sigma(\tilde{T}) = \{\pm\frac{1}{2}\sqrt{4+\eta} \pm \frac{1}{2}\sqrt{\eta}\}$.

If η is perturbed by a small value ϵ, the eigenvalues change by

$$\tilde{\lambda}_i(\eta+\epsilon) - \tilde{\lambda}_i(\eta) = \left(\pm\frac{1}{4}\frac{1}{\sqrt{4+\eta}} \pm \frac{1}{4}\frac{1}{\sqrt{\eta}}\right)\epsilon + O(\epsilon^2), \qquad i = 1, \ldots, 4.$$

Thus, the eigenvalues are very sensitive to changes if $\eta \approx 0$ or $\eta \approx -4$. If $\eta = 0$ or $\eta = -4$, then $\tilde{T}$ is a matrix with two degenerate eigenvalues.

If $\eta \approx \sqrt{\epsilon}$, then, for all i,

$$\tilde{\lambda}_i(\sqrt{\epsilon}+\epsilon) - \tilde{\lambda}_i(\sqrt{\epsilon}) = \frac{1}{4}\epsilon^{3/4} + O(\epsilon). \tag{5.2}$$

The eigenvectors are as sensitive as the eigenvalues,

$$\|\tilde{\mathbf{q}}_i(\eta+\epsilon) - \tilde{\mathbf{q}}_i(\eta)\| = \frac{3\sqrt{2}}{4}\epsilon^{3/4} + O(\epsilon), \qquad i = 1, \ldots, 4.$$

This is the typical behavior of eigenvalues and eigenvectors of matrices that are close to defective matrices.

If we assume that $\epsilon \approx$ *macheps* $\approx 10^{-16}$, then $\sqrt{\epsilon}$ is a small value but still not small enough to allow deflation. With this assumption we lose about four significant digits in computing the eigenvalues and eigenvectors of $\tilde{T}$ if $\eta \approx 10^{-8}$ using any eigensolver. This is what is actually observed in a numerical calculation with standard routines such as those in MATLAB.

If the zeros of the spectral function

$$\zeta(\lambda) = 1 + \eta\mathbf{e}_2^T \begin{pmatrix} \lambda & -1 \\ -1 & \lambda-\eta \end{pmatrix}^{-1} \mathbf{e}_2 + \mathbf{e}_1^T \begin{pmatrix} \lambda-1 & -1 \\ -1 & \lambda \end{pmatrix}^{-1} \mathbf{e}_1 \tag{5.3}$$

corresponding to (5.1) are computed, the behavior (5.2) is observed as roundoff is introduced when solving the two 2×2 systems of linear equations in (5.3). This is in accordance with backward error analysis that considers these zeros as exact zeros of a function with nearby coefficients.

The eigenvalues of $\tilde{T}$ in Example 5.1 are quite sensitive to perturbations of the matrix elements. One may suggest that the large condition number $4/\sqrt{\eta} + O(1)$ of the left matrix in (5.3) is a consequence of this sensitivity. This is however not the case. Sensitive eigenvalues $\tilde{\lambda}$ of $\tilde{T}$ do not necessarily imply large condition numbers

of $\tilde{\lambda} - T_1$ or $\tilde{\lambda} - T_2$. This is especially true when the eigenvalues of T and $\tilde{T}$ are well separated as the following example shows.

Example 5.2. The matrix

$$\tilde{T} = \begin{pmatrix} 3 & -6 & 0 & 0 \\ 1 & -\frac{5}{3} & -\frac{5}{18} & 0 \\ 0 & 1 & -\frac{11}{15} & -\frac{3}{50} \\ 0 & 0 & 1 & -\frac{3}{5} \end{pmatrix}$$

$$= T + \mathbf{u}\mathbf{v}^T = \begin{pmatrix} 3 & -6 & 0 & 0 \\ 1 & -\frac{35}{18} & 0 & 0 \\ 0 & 0 & \frac{4}{15} & -\frac{3}{50} \\ 0 & 0 & 1 & -\frac{3}{5} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{5}{18} \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}^T$$

has a single four-fold eigenvalue 0. Numerical approximations of the eigenvalues and vectors of $\tilde{T}$, computed with MATLAB, have absolute errors of $O(\epsilon^{1/4})$. This is due to the high sensitivity of the eigenvalue of $\tilde{T}$ with respect to perturbations of the matrix elements.

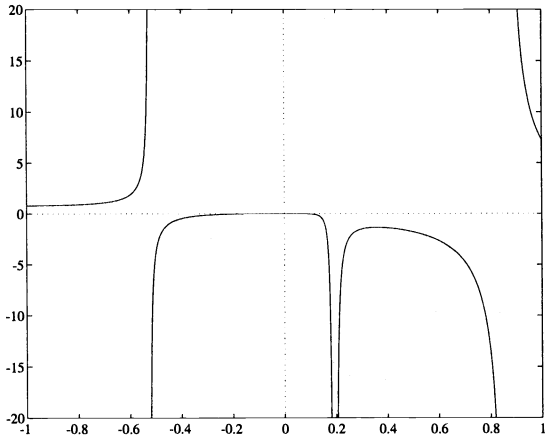


FIG. 5.1. *Plot of* $\zeta(\lambda) = 1 + \mathbf{v}^T(T - \lambda)^{-1}\mathbf{u}$, $-1 < \lambda < 1$, *of Example* 5.2.

T has no eigenvalue close to zero. So, the eigenvector of $\tilde{T}$ corresponding to zero could be computed with high accuracy without difficulty using (3.7) if a good enough approximation of the eigenvalue were available. This has some resemblance to the situation in the symmetric tridiagonal eigenvalue problem where eigenvectors may be inaccurately computed due to eigenvalues that are not known to have sufficient precision [15], [20]. If we use $\lambda = 0$ in (3.7), the calculated eigenvector using double precision is $(10, 4, 5, 6)^T$, which is exact to within *macheps*. However, for this problem, it is difficult for a rootfinder to accurately locate the root of $\zeta(\lambda)$ in (4.4) corresponding to $\tilde{\lambda} = 0$. This is because ζ is very flat near zero (cf. Fig. 5.1). In particular, in the intervals $[-.0007, .0]$ and $[0, .0007]$ the values of $\zeta(\lambda)$ range from -10^{-11} to 0 and 0 to -10^{-11}, respectively. If we use the value $-.00002$ as the computed approximation of 0 (the associated value of ζ is $8.88 \cdot 10^{-16}$), then the corresponding eigenvector

computed by (3.7) is $(9.9975, 4.9988, 5.9988, 10.9987)^T$. So here, an absolute error of 10^{-5} in the eigenvalue caused an error in the third decimal place of each component in the eigenvector.

The two examples above have shown matrices with ill-conditioned eigenvalues whose spectral decomposition is hard to compute with any algorithm. The problems were more or less ill posed and their solutions intrinsically hard. In the context of divide and conquer algorithms, the question arises: What happens when the given matrix $\tilde{T}$ has a well-conditioned spectral decomposition but T has a block with sensitive eigenvalues and vectors?

As a consequence of the assumption that our matrices need not be diagonalizable, we do not use the spectral decomposition to compute either the zeros $\tilde{\lambda}$ of ζ or the eigenvectors. Rather we use formula (4.4) to get the $\tilde{\lambda}$ and formula (3.7) to compute the corresponding eigenvectors. Therefore, the sensitivity of the eigenvalues and vectors is determined by the condition number of $T - \tilde{\lambda}$, $\tilde{\lambda} \in \sigma(\tilde{T})$. This number is worse the closer the eigenvalues of T and $\tilde{T}$ are. If $\tilde{\lambda}$ is even a persistent eigenvalue, the corresponding eigenvectors of $\tilde{T}$ are determined by (3.15), which involve the eigenvectors of T corresponding to $\tilde{\lambda}$. In this way we inherit the sensitivity of the eigenvalues of the submatrices of T.

Notice that on the other hand it may be advantageous to have the eigenvalues of T and $\tilde{T}$ close to each other. The eigenvalues of T would then be a good starting guess for an iterative procedure to find the eigenvalues of $\tilde{T}$.

An illustrative example for close eigenvalues of T and $\tilde{T}$ is the following example.

Example 5.3. Let

$$\tilde{T} = \begin{pmatrix} 1+\eta & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = T + \mathbf{u}\mathbf{v}^T = \begin{pmatrix} 1+\eta & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}^T.$$

T has the spectrum $\sigma(T) = \{\frac{1}{2}[\eta \pm \sqrt{\eta^2 + 4\eta}], 0, 2\}$. For $\eta = 0$, the triple eigenvalue 0 has geometric multiplicity 2. $\tilde{T}$ has four well-separated eigenvalues all of which are well conditioned. The eigenvalue of $\tilde{T}$ closest to zero is

$$\tilde{\lambda} = -\frac{1}{2}\eta + \frac{1}{2}\eta^2 + O(\eta^3).$$

The corresponding eigenvector is given by

$$\tilde{\mathbf{x}} = \begin{pmatrix} 1 + \frac{3}{4}\eta + O(\eta^2) \\ 1 + \frac{9}{4}\eta + O(\eta^2) \\ -1 - \frac{5}{4}\eta + O(\eta^2) \\ 1 + \frac{3}{4}\eta + O(\eta^2) \end{pmatrix}.$$

Calculating this eigenpair with MATLAB for a small value η reflects these expansions. The errors in the computed numbers are of the order $O(macheps)$.

The condition number of $T - \tilde{\lambda}$ is $4/\eta + O(1)$. Again this can be observed in a MATLAB computation. For $\eta = \sqrt{macheps}$, approximately eight significant digits are lost if $\tilde{\mathbf{x}}$ is computed by formula (3.7a), assuming $\tilde{\lambda}$ is given exactly.

So, the divide and conquer algorithm indeed can suffer severely from the ill conditioning of split matrices. It should however be noted that the decomposition of $\tilde{T}$

given in Example 5.3 is not the only possible one. For instance, if we write

$$\tilde{T} = \begin{pmatrix} 1+\eta & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1+\eta & -1 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{2} \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2 \\ 0 \end{pmatrix}^T,$$

the matrix T has well-separated eigenvalues and $T - \tilde{\lambda}$ is well conditioned in a neighborhood of 0. Here, we have used the freedom of how to choose the parameter ω in (4.3). In Example 5.3 we set $\omega = 1$, here we changed it to $\omega = 2$.

It would be interesting to find out to what extent a good selection of ω can improve stability in general. To our knowledge, this question has only been investigated for divide and conquer algorithms for systems of linear equations [19]. Of course, the selection of the split-index k also has an influence on the stability of the eigenvalues of T.

6. Conclusions. In this paper we have derived the theory needed to devise divide and conquer algorithms for nonsymmetric eigenvalue problems. We have formulated an algorithm for solving eigenvalue problems involving tridiagonal matrices and have commented on an analogous algorithm for Hessenberg matrices. The tridiagonal divide and conquer algorithm has a very favorable complexity: $O(n^2)$ and $O(n)$ for the computation of all eigenvalues and eigenvectors on sequential and parallel computers with $O(n)$ processors, respectively.

As expected, the cost for the Hessenberg divide and conquer algorithm is one power of n higher. We see no reason to compute with Hessenberg matrices in a divide and conquer algorithm. The higher order complexity is not rewarded by a gain in stability! The Hessenberg form is well suited for the computation of the Schur normal form but inappropriate if eigenvectors are desired. The computation of the Schur normal form on the other hand should not be approached by a divide and conquer algorithm.

The difficulty of stably tridiagonalizing arbitrary matrices prevents a widespread use of nonsymmetric tridiagonal eigenproblem solvers. It may be possible to transform arbitrary matrices stably into a very sparse but no longer tridiagonal form [22]. The divide and conquer algorithm could eventually take advantage of this form as systems of linear equations with such matrices must be formed in the course of the algorithm.

There are however two important issues that must be solved successfully before a divide and conquer algorithm will be feasible.

Zero finding. There are two difficult problems connected with finding the zeros of the secular equation $\zeta(\lambda) = 0$. The first is determining the number of zeros of ζ and the second is the actual search procedure.

The first problem is difficult as geometric and algebraic multiplicities of the eigenvalues of nonsymmetric matrices can differ and, as Example 3.2 illustrates, counting only geometric multiplicities is not sufficient. In theory, one can determine the algebraic multiplicities by finding the principal vectors by the procedure described in §3. However, in practice, this procedure will most likely suffer from stability problems.

It is possible to determine the number of the zeros of a function in specific domains of the complex plane. Jessup [17] discusses methods based on the principle of the argument (Cauchy's integral formula). Stickel [21] introduced a method to determine the number of eigenvalues in a particular parallelogram. Both methods are too expensive.

One must probably resort to Newton's iteration. As with any zerofinder, in order not to find the same zero several times, it may be necessary to successively deflate zeros, which introduces communication and potentially ruins parallelism. The amount of work to find the zeros of ζ depends to a great extent on the nature of the zero. So, load balancing is also difficult to achieve.

Stability. It is principally arguable whether it is reasonable to try to compute eigenvectors of nonsymmetric matrices instead of Schur vectors as is done stably in the QR algorithm.

The former can be arbitrarily ill conditioned. Assuming the matrices to be diagonalizable does not help as the diagonalizable matrices are a dense subset of the algebra of matrices.

Even if the original matrix has well-conditioned eigenvalues and eigenvectors, it is not clear whether the split subproblems have this property (cf., Example 5.3). There may be ways to minimize the condition of the eigenvalues of the split submatrices by choosing the split-index k or the parameter ω in (4.3) properly.

In summary, it appears to us that zerofinding and stability make it difficult to find a successful and fast implementation of a divide and conquer algorithm for the large nonsymmetric eigenvalue problem. There may be subclasses of this eigenvalue problem that can be computed safely. In any case, an implementation of a divide and conquer algorithm must be able to detect a loss of accuracy.

REFERENCES

[1] L. Adams and P. Arbenz, *Towards a divide and conquer algorithm for the real nonsymmetric eigenvalue problem*, Tech. Report 91-8, Department of Applied Mathematics, University of Washington, Seattle, WA, August 1991.

[2] G. Ammar, L. Reichel, and D. Sorensen, *An implementation of a divide and conquer algorithm for the unitary eigenproblem*, ACM Trans. Math. Software, 18 (1992), pp. 292–307.

[3] P. Arbenz, W. Gander, and G. H. Golub, *Restricted rank modification of the symmetric eigenvalue problem: Theoretical considerations*, Linear Algebra Appl., 104 (1988), pp. 75–95.

[4] P. Arbenz and G. H. Golub, *On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[5] C. Beattie and D. Fox, *Schur complements and the Weinstein–Aronszajn theory for modified matrix eigenvalue problems*, Linear Algebra Appl., 108 (1988), pp. 37–61.

[6] J. R. Bunch, C. P. Nielson, and D. C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[7] J. J. M. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

[8] J. J. Dongarra and D. C. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. s139–s154.

[9] K. Gates, *Using inverse iteration to improve the divide and conquer algorithm*, Tech. Report 159, Departement Informatik, ETH Zürich, May 1991.

[10] G. A. Geist, *Reduction of a general matrix to tridiagonal form*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 362–373.

[11] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[13] S. H. Gould, *Variational Methods for Eigenvalue Problems*, 2nd ed., University of Toronto Press, Toronto, 1966.

[14] W. B. Gragg and L. Reichel, *A divide and conquer method for unitary and orthogonal eigenproblems*, Numer. Math., 57 (1990), pp. 695–718.

[15] M. Gu and S. C. Eisenstat, *A stable and efficient algorithm for the rank-one moodification of the symmetric eigenproblem*, Research Report YALEU/DCS/RR-916, Department of Computer Science, Yale University, New Haven, CT, September 1992.

[16] I. C. F. IPSEN AND E. R. JESSUP, *Solving the tridiagonal eigenvalue problem on the hypercube*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 203–229.

[17] E. R. JESSUP, *A case against a divide and conquer approach to the nonsymmetric eigenvalue problem*, Appl. Numer. Math., 12 (1993), pp. 403–420.

[18] E. R. JESSUP AND D. C. SORENSEN, *A parallel algorithm for computing the singular value decomposition of a matrix*, Tech. Memo. 102, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, December 1987.

[19] V. MEHRMANN, *Divide and conquer methods for block tridiagonal linear systems*, Parallel Comput., 19 (1993), pp. 257–279.

[20] D. C. SORENSEN AND P. T. P. TANG, *On the orthogonality of eigenvectors computed by divide-and-conquer techniques*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.

[21] E. STICKEL, *Separating eigenvalues using the matrix sign function*, Linear Algebra Appl., 148 (1991), pp. 75–88.

[22] W.-P. TANG, *A stabilized algorithm for tridiagonalization of an unsymmetric matrix*, Tech. Report CS-88-14, Department of Computer Science, University of Waterloo, Ontario, April 1988.

[23] D. S. WATKINS, *Fundamentals of Matrix Computations*, Wiley, New York, 1991.

[24] A. WEINSTEIN AND W. STENGER, *Methods of Intermediate Problems for Eigenvalues*, Academic Press, New York, 1972.

[25] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1354–1362, October 1994

020

GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING CAN FAIL IN PRACTICE*

LESLIE V. FOSTER†

Abstract. Even though Gaussian elimination with partial pivoting is very widely used, $n \times n$ matrices can be constructed where the error growth in the algorithm is proportional to 2^{n-1}. Thus for moderate or large n, in theory, there is a potential for disastrous error growth. However, prior to 1993 no reports of such an example in a practical application had appeared in the literature. Examples are presented that arise naturally from integral and differential equations and that lead to disastrous error growth in Gaussian elimination with partial pivoting.

Key words. Gaussian elimination, numerical stability, integral equations

AMS subject classifications. 65F05, 65R20, 65G05

1. Introduction.

Gaussian elimination with partial pivoting (GEPP) is one of the most widely used algorithms in scientific computing. When applied to an $n \times n$ matrix A it results in a factorization $PA = LU$, where P is a permutation matrix, L is lower triangular, and U is upper triangular. Let $\hat{\mathbf{x}}$ represent the solution to $A\mathbf{x} = \mathbf{b}$ computed in floating point arithmetic on a computer with relative machine precision ϵ. Then it is known [Wil] that

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} \leq 4n^2 \mathrm{cond}_\infty(A)\rho\,\epsilon,$$

where $\mathbf{x}$ is the exact solution, $\mathrm{cond}_\infty(A)$ is the condition number of A in the supremum norm and ρ is the growth factor,

$$\rho = \frac{\max_{i,j,k} |a_{i,j}^{(k)}|}{\max_{i,j} |a_{i,j}|} \tag{1.1}$$

with $a_{i,j}^{(k)}$ denoting the i,j element after the kth step of elimination. Thus GEPP is considered numerically stable unless ρ is large.

The theory for GEPP suggests that ρ can be very large. The sharpest bound is $\rho \leq 2^{n-1}$ and this is attained, for example, for matrices A_n of the form [HH], [Wil], [GVL]

$$A_5 = \mathrm{diag}(\pm 1) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} & & & 0 \\ & T & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & \theta \end{pmatrix},$$

where T is an $n-1 \times n-1$ nonsingular upper triangular matrix and $\theta = \max |a_{ij}|$. Thus for moderate or large n the growth factor can be large. However, more than 25 years ago Wilkinson reported:

* Received by the editors November 4, 1992; accepted for publication (in revised form) August 13, 1993.

† Department of Mathematics and Computer Science, San Jose State University, San Jose, California, 95192 (foster@sjsumcs.sjsu.edu).

> "It is our experience that any substantial increase in size of elements of successive A_n is extremely uncommon even with partial pivoting. ... No example which has arisen naturally has in my experience given an increase by a factor as large as 16."

Since Wilkinson made his remarks, Dongarra et al. [DBMS] report an example where ρ is 23 and Higham and Higham [HH] report several natural, noncontrived examples where the growth factor is between $n/2$ and n. Although the growth factors reported in these papers are larger than those mentioned by Wilkinson, they are much less than the theoretical limit of 2^{n-1}. For random matrices Trefethen and Schreiber [TS] show that the average growth factor does not grow exponentially. In this paper we present a class of practical examples where the growth factors do grow exponentially. Recently Wright [Wri] also presented such a class. Wright's paper and ours are different in that we consider Volterra integral equations, which are not discussed by Wright, and the growth factors for our matrices can be closer to the theoretical limit than the growth factors for Wright's matrices. Also the matrices in our examples are dense whereas Wright's are sparse. The papers are related in that results in both papers apply to boundary value problems.

In the next section we show that when the quadrature method [Bak], [DM], [Linz] is used to numerically solve certain Volterra integral equations, large growth factors can result. In §3 we illustrate the theory of §2 with a population growth model and with a two-point boundary value problem. In the last section we present a brief discussion of the implications of such examples.

Software, in the form of MATLAB m files for constructing the above examples is available via the gopher system. Type "gopher sundance.sjsu.edu" on a computer with a gopher client and follow the menus.

2. A class of Volterra integral equations that lead to large growth factors. A linear Volterra integral equation of the second kind is of the form:

$$x(s) - \int_0^s k(s,t)x(t)\,dt = G(s). \tag{2.1}$$

Such equations show up in a wide variety of applications [Bur], [Jer], [Linz]. We consider for known $k(s,t)$, $\beta(s)$, and $G(s)$ the following modification of (2.1):

$$x(s) - \int_0^s k(s,t)x(t)\,dt + \beta(s)x(L) = G(s). \tag{2.2}$$

In general it is not possible to find an exact solution to (2.1) or (2.2). However, a variety of approximation techniques [Bak], [DM], [Linz] can be used, including the commonly used quadrature method. This is the method that we use, "starting" our procedure with the block quadrature method [DM]. We use Newton–Cotes quadrature formulas of order $p \geq 1$. To be specific for any n we divide $0 \leq s \leq L$ into $n-1$ equal subintervals of length $h = L/(n-1)$. For $i = 1,\ldots,n$, $j = 1,\ldots,n$, let $s_i = (i-1)h$, $t_j = (j-1)h$, $\beta_i = \beta(s_i)$, $b_i = G(s_i)$, $k_{ij} = k(s_i,t_j)$, and let x_i be the numerical approximation to $x(s_i)$. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)^T$, where the superscript T indicates transpose. We approximate $\int_0^{s_i} k(s,t)x(t)\,dt$. Our approximation depends on i. For $1 \leq i \leq p+1$ we integrate an interpolating polynomial of degree p through $(t_j, k_{ij}\,x_j), j = 1,\ldots,p+1$. For $i \geq p+1$ we use composite integration. For example if $i = kp + l$ we use standard pth order closed Newton–Cotes composite integration for the integral from 0 to s_{kp+1}. For the integral from

s_{kp+1} to s_{kp+l}, we integrate an interpolating polynomial of degree $p+1$ through $(t_j, k_{ij}\, x_j), j = kp+l-p-1, \ldots, kp+l$.

With these approximations the vector $\mathbf{x}$ satisfies

$$A\mathbf{x} = \mathbf{b}. \tag{2.3}$$

For example, for $p = 2$ and $n = 7$ the matrix A is

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & \beta_1 \\
-\frac{5hk_{21}}{12} & 1-\frac{2hk_{22}}{3} & \frac{hk_{23}}{12} & 0 & 0 & 0 & \beta_2 \\
-\frac{hk_{31}}{3} & -\frac{4hk_{32}}{3} & 1-\frac{hk_{33}}{3} & 0 & 0 & 0 & \beta_3 \\
-\frac{3hk_{41}}{8} & -\frac{9hk_{42}}{8} & -\frac{9hk_{43}}{8} & 1-\frac{3hk_{44}}{8} & 0 & 0 & \beta_4 \\
-\frac{hk_{51}}{3} & -\frac{4hk_{52}}{3} & -\frac{2hk_{53}}{3} & -\frac{4hk_{54}}{3} & 1-\frac{hk_{55}}{3} & 0 & \beta_5 \\
-\frac{hk_{61}}{3} & -\frac{4hk_{62}}{3} & -\frac{17hk_{63}}{24} & -\frac{9hk_{64}}{8} & -\frac{9hk_{65}}{8} & 1-\frac{3hk_{66}}{8} & \beta_6 \\
-\frac{hk_{71}}{3} & -\frac{4hk_{72}}{3} & -\frac{2hk_{73}}{3} & -\frac{4hk_{74}}{3} & -\frac{2hk_{75}}{3} & -\frac{4hk_{76}}{3} & 1-\frac{hk_{77}}{3}+\beta_7
\end{pmatrix}$$

In this case rows three, five, and seven are produced by using Simpson's rule for integration. Rows four and six have an odd number of intervals of integration and involve the usual Simpson rule and Simpson 3/8 rule [Linz, p. 99]. Row two arises from the block quadrature method [DM].

This approach has the attractive feature that involves higher order approximations for $p > 1$ and that, except for the last column, the matrix A is block lower triangular and so, in principle, (2.3) can be solved in $O(n^2)$ not $O(n^3)$ operations.

For large n we have the following results.

THEOREM 2.4. *Assume that $k(s,t)$ is bounded for $0 \le s \le L$, $0 \le t \le L$. For any fixed order of integration $p \ge 1$ and for sufficiently large n no row interchanges are required when* GEPP *is applied to the matrix A in* (2.3).

Proof. Let w_{ij} be the weights in the numerical integration formula corresponding to the i,j element of A and choose n such that $\delta \equiv \max_{1\le i,\, j\le n} |w_{ij}k_{ij}h| \le 1/(p+1)$. The first $n-1$ columns of A are lower triangular except for a $p \times p$ diagonal block in columns 2 through $p+1$. Outside these columns no row interchanges are required in GEPP since $\delta \le 1/(p+1) \le \frac{1}{2}$. Let $\hat{A}$ consist of columns 2 to $p+1$ of A and $\hat{I}$ be an $n \times p$ matrix with ones on the diagonal and zeros elsewhere. Then $\hat{A} = \hat{I} - B$, where $\max_{1\le i\le n,\, 1\le j\le p} |b_{ij}| \le \delta$. Let $\hat{A}^k$ be $\hat{A}$ after k steps of GEPP, let $B^k = \hat{I} - \hat{A}^k$, and let $x_k = \max_{1\le i\le n,\, 1\le j\le p} |b^k_{ij}|$. Clearly, no row interchanges are required at the first step of GEPP applied to $\hat{A}$.

We now use induction. For some k, $1 \le k \le p-1$ assume no row interchanges are required through step k of GEPP and that $x_k \le \delta/(1-k\delta)$. Then for $i = k+1, \ldots, n$, $|\hat{a}^k_{ik}| \le x_k \le [1/(p+1)]/[(1-k/(p+1)] \le \frac{1}{2}$ and $|\hat{a}^k_{kk}| \ge 1 - x_k \ge \frac{1}{2}$. Therefore no pivoting will be required at step $k+1$. Also since $\hat{a}^{k+1}_{ij} = \hat{a}^k_{ij} - \hat{a}^k_{kj}\hat{a}^k_{ik}/\hat{a}^k_{kk}$ then for $i \ne j$, $|\hat{a}^{k+1}_{ij}| \le x_k + x_k x_k/(1-x_k) = x_k/(1-x_k)$. For $i = k+1, \ldots, p$ we then have $|\hat{a}^{k+1}_{ii} - 1| \le x_k + x_k x_k/(1-x_k) = x_k/(1-x_k)$. Consequently $x_{k+1} \le x_k/(1-x_k)$. This and $x_k \le \delta/(1-\delta k)$ imply that $x_{k+1} \le \delta/[1-(k+1)\delta]$. This completes the proof and also shows that $\max_{1\le i\le n,\, 1\le j\le p} |b^p_{ij}| \le \delta/(1-p\delta)$. □

THEOREM 2.5. *Assume that $k(s,t)$ is continuous over $0 \le s \le L$, $0 \le t \le L$, and $\beta(s)$ is continuous over $0 \le s \le L$. Let $\gamma(s)$ be the solution to the integral equation*

$$\gamma(s) - \int_0^s k(s,t)\gamma(t)\,dt = \beta(s). \tag{2.6}$$

For any fixed p the growth factor ρ for GEPP *applied to* (2.3) *satisfies*

$$\lim_{n\to\infty} \rho = \frac{\max_{0\le\tau\le s\le L}\{1,\ |\beta(s) + \int_0^\tau k(s,t)\gamma(t)\,dt|,\ |1+\beta(L) + \int_0^\tau k(s,t)\gamma(t)\,dt|\}}{\max_{0\le s\le L}\{1,\ |\beta(s)|,\ |1+\beta(L)|\}}. \tag{2.7}$$

Proof. Select a fixed τ, $0 \le \tau \le L$ and suppose that k is an integer, $p+1 \le k \le n-1$ such that $hk = \tau$. Assume that h is sufficiently small so that no pivoting is required in GEPP applied to A in (2.3). Then after step k of GEPP, we have

$$\begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & I & 0 \\ L_{31} & 0 & 1 \end{pmatrix} \begin{pmatrix} U_{11} & 0 & \mathbf{u}_1 \\ 0 & A_{22} & \mathbf{u}_2 \\ 0 & A_{32} & u_n \end{pmatrix} = A = \begin{pmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & \tilde{a}_{nn} \end{pmatrix} + \mathbf{e}_n^T \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \beta_n \end{pmatrix}. \tag{2.8}$$

In (2.8) there are k, $n-k-1$, and 1 elements, respectively, in the first, second, and third block rows and columns. Here $\mathbf{e}_n = (0,0,\ldots,0,1)^T$.

Now let $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \beta_n)^T$, $\mathbf{u} = (\mathbf{u}_1^T, \mathbf{u}_2^T, u_n)^T$ and $\mathbf{v}_1 \in R^k$ satisfy $U_{11}\mathbf{v}_1 = \mathbf{u}_1$. Then from (2.8) $L_{11}\mathbf{u}_1 = \boldsymbol{\beta}_1$ and $L_{11}U_{11} = A_{11}$ so that $A_{11}\mathbf{v}_1 = \boldsymbol{\beta}_1$. This last equation is the equation produced when the quadrature method is applied to (2.6) over the interval $0 \le s \le \tau = kh$. By standard results [DM, pp. 126–127], for $s_i = ih$ fixed and $i \le k$

$$v_i - \gamma(s_i) \to 0 \quad \text{as } h \to 0. \tag{2.9}$$

In the proof of Theorem 2.4 it was shown that $U_{11} = I + \tilde{B}$, where $\tilde{b}_{ij} = 0$, for $j > i > p+1$, and where $\max_{1\le i,j\le n} |\tilde{b}_{ij}| \le \delta/(1-p\delta)$, with $\delta \to 0$ as $h \to 0$. It then follows from (2.9) that for ih fixed and $i \le k$

$$u_i - \gamma(s_i) \to 0 \quad \text{as } h \to 0. \tag{2.10}$$

From the second block row in (2.8), we have $L_{21}\mathbf{u}_1 + \mathbf{u}_2 = \boldsymbol{\beta}_2$ and $L_{21}U_{11} = A_{21}$. Therefore $\mathbf{u}_2 = -A_{21}\mathbf{v}_1 + \boldsymbol{\beta}_2$. Since elements in $-A_{21}$ come from discrete approximations to integrals and from (2.9) it follows that for i such that $k+1 \le i \le n-1$ and for s_i fixed then

$$u_i \to \int_0^\tau k(s,t)\gamma(t)\,dt + \beta(s_i) \text{ as } h \to 0. \tag{2.11}$$

From the third block row in (2.8), we also have $L_{31}\mathbf{u}_1 + u_n = \tilde{a}_{nn} + \beta_n$ and $L_{31}U_{11} = A_{31}$ so that $u_n = \tilde{a}_{nn} - A_{31}\mathbf{v}_1 + \beta_n$. Since $\tilde{a}_{nn} \to 1$ as $h \to 0$, we see that

$$u_n \to 1 + \beta(L) + \int_0^\tau k(s,t)\gamma(t)\,dt. \tag{2.12}$$

In proving (2.10), (2.11), and (2.12) we have assumed that $k \ge p+1$. These equations are also true for $k \le p$. For example, for $i \le k \le p$ as $h \to 0$ it follows easily that $u_i \to \beta(0)$ and $\gamma(s_i) \to \beta(0)$ so that (2.10) is true. The theorem follows from (2.10)–(2.12) and the definition of growth factor. □

COROLLARY 2.13. *With the assumptions of Theorem* 2.5

$$\lim_{n\to\infty} \rho \geq \frac{\max_{0\leq s\leq L}\{1,\ |\gamma(s)|,\ |1+\gamma(L)|\}}{\max_{0\leq s\leq L}\{1,\ |\beta(s)|,\ |1+\beta(L)|\}}.$$

Proof. The result follows by letting $\tau = s$ in (2.7) and using (2.6). □

Corollary 2.13 shows that for any of the numerical integration schemes that we have outlined, large growth occurs for sufficiently large n if the solution $\gamma(s)$ to (2.6) is large relative to the coefficient $\beta(s)$ in (2.2). The next section shows that large growth can happen for practical problems where A is well conditioned.

3. Examples. Our first example comes from a simple model for population dynamics. Let $x(s)$ represent the population of a species at time s and let x_0 be the initial population. For some fixed time L assume for $0 \leq s \leq L$ that births occur at a rate $r(s)$ and deaths are governed by a survival function $f(s)$ where a fraction $f(s-t)$ of the organisms born at time t are alive at time s, $t \leq s \leq L$. It follows [Jer] that

$$x(s) = x_0 f(s) + \int_0^s f(s-t)r(t)\,dt, \qquad 0 \leq s \leq L. \tag{3.1}$$

If we assume that the birth rate is proportional to the population, $r(t) = \kappa x(t)$ for some constant κ, then (3.1) becomes a Volterra integral equation of the second kind (2.1) with $k(s,t) = \kappa f(s-t)$ and $G(s) = x_0 f(s)$. On the other hand if we introduce a birth control policy where the birth rate is reduced by an amount proportional to the final population so that $r(t) = \kappa x(t) - \alpha x(L)$, for a constant α, then (3.1) reduces to the form (2.2) with $k(s,t) = \kappa f(s-t)$, $G(s) = x_0 f(s)$, and $\beta(s) = \int_0^s \alpha f(s-t)\,dt$.

We can now illustrate the results of §2 by assuming, say, that $x_0 = 1$, $\kappa = 1$, $L = 50$, $\alpha = .5$, and $f(s) = e^{-cs}$ with $c = .25$ so that $\beta(s) = \alpha(1 - e^{-cs})/c$. For most functions $f(s)$, (2.1) or (2.2) with $k(s,t) = \kappa f(s-t)$ do not have exact solutions in terms of the usual transcendental functions and are not equivalent to ordinary differential equations. However, to calculate errors we choose a simple $f(s)$ so that (2.2) has an exact solution $x(s) = x_0[\alpha+(\kappa-c-\alpha)e^{(\kappa-c)(s-L)}]/[\alpha+(\kappa-c-\alpha)e^{-(\kappa-c)L}]$. In this case the solution to (2.6) is $\gamma(s) = \alpha[1 - e^{(\kappa-c)s}]/(\kappa - c)$. Thus by Corollary 2.13 for large n the growth factor should be approximately

$$\frac{1+\alpha[1-e^{(\kappa-c)L}]/(\kappa-c)}{1+\alpha(1-e^{-cL})/c} = 4.3 \times 10^{15},$$

or larger, and partial pivoting should have large error growth.

In Fig. 3.2 we have plotted, versus n, the growth factors as computed by (1.1) when solving (2.3) for $p = 2$ by GEPP and complete pivoting. For partial pivoting row interchanges are required for $n \leq 92$ and the growth factor remains moderate. However for $n \geq 93$ no row interchanges are required for partial pivoting and the growth factor becomes large. For $n = 200$ the computed growth factor is 4.02×10^{15}, close to the above theoretical estimate. Also in Fig. 3.2 we have plotted, versus n, the relative error in the approximate solution to (2.1) obtained by solving (2.3) by partial pivoting and complete pivoting. Here by relative error we mean $\max_{i=1,\ldots,n} |x(s_i) - x_i| / \max_{i=1,\ldots,n} |x(s_i)|$, where $x(s)$ is the true solution. Gaussian elimination with complete pivoting is numerically stable and for this example the relative error decreases proportional to h^4 (since our numerical integration is based on the Simpson rule). On the other hand, for partial pivoting, the large growth factor leads a large relative error in the calculated answer. GEPP *is unstable and inaccurate for* $n \geq 93$.

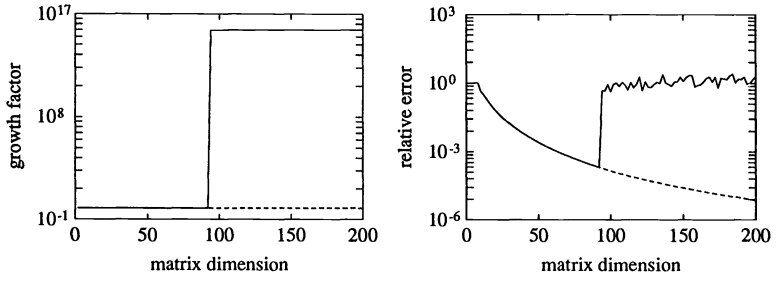


FIG. 3.2. *The growth factor (left), and the relative error (right), when solving* (2.3) *with partial pivoting (solid lines) and complete pivoting (dashed lines).*

To further illustrate the difficulty with partial pivoting in Fig. 3.3 we have plotted the approximate solution to (2.2) obtained by solving (2.3) using partial pivoting with $n = 100$ and the true solution. In this case, to solve (2.3) we used the MATLAB "\" operator, which is based on Linpack's implementation of GEPP, on a SUN SPARCstation. The large discrepancy for $s \geq 42$ is due to the instability of partial pivoting.

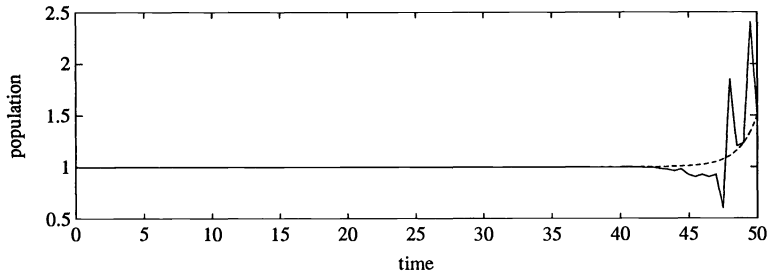


FIG. 3.3. *The true solution (dashed) to* (2.3) *and approximate solution (solid) calculated when using partial pivoting for* $n = 100$.

Finally we note that on a modern workstation it requires less than a second to set up and solve (2.3) for $n = 100$, say, and that the linear systems solved to produce the graphs in Figs. 3.2 and 3.3 are well conditioned. For example, the matrices used to produce Fig. 3.2 all have condition numbers less than 162.

For our second example we consider a boundary value problem. Suppose constants L, k, and C and a function $g(t)$, $0 \leq t \leq L$, are known and that an unknown function $x(t)$, $0 \leq t \leq L$, satisfies

$$x'(t) = kx(t) + g(t), \quad 0 < t < L \quad \text{with } x(L) = Cx(0). \tag{3.4}$$

This example is simple enough so that we can find the solution exactly, but suppose that we wished to solve it numerically. We choose to first convert the differential equation (3.4) into an integral equation by integrating from zero to s and substitute

in $x(0) = x(L)/C$ to get

$$x(s) - \int_0^s kx(t)\,dt - x(L)/C = \int_0^s g(t)\,dt \equiv G(s). \tag{3.5}$$

This is of the form (2.2) with $k(s,t) = k$ and $\beta(s) = -1/C$. If we apply the quadrature method to (3.4) using the trapezoid rule to approximate the integrals, the resulting linear system $Ax = b$ is simple enough in this case so that we can exactly describe L and U in an LU factorization of A. It is easy to check for $b = 1 - kh/2$ and $\omega = (1 + kh/2)/(1 - kh/2)$ that

$$A = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & -1/C \\ -\frac{kh}{2} & 1-\frac{kh}{2} & 0 & \cdots & 0 & -1/C \\ -\frac{kh}{2} & -kh & 1-\frac{kh}{2} & \ddots & \vdots & \vdots \\ \vdots & \vdots & & \ddots & 0 & -1/C \\ -\frac{kh}{2} & -kh & \cdots & -kh & 1-\frac{kh}{2} & -1/C \\ -\frac{kh}{2} & -kh & \cdots & -kh & -kh & 1-1/C-\frac{kh}{2} \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & \cdots & & & 0 \\ -\frac{kh}{2} & 1 & \ddots & & & \vdots \\ -\frac{kh}{2} & -\frac{kh}{b} & 1 & & & \\ -\frac{kh}{2} & -\frac{kh}{b} & -\frac{kh}{b} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ -\frac{kh}{2} & -\frac{kh}{b} & -\frac{kh}{b} & \cdots & -\frac{kh}{b} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & & 0 & -\frac{1}{C} \\ 0 & b & \ddots & & \vdots & -\frac{b\omega}{C} \\ \vdots & \ddots & b & & & -\frac{b\omega^2}{C} \\ & & & \ddots & 0 & \vdots \\ & & & & b & -\frac{b\omega^{n-2}}{C} \\ 0 & \cdots & & & 0 & -\frac{b\omega^{n-1}}{C} + b \end{pmatrix}$$

From this factorization it follows easily that if $|kh| \le 2/3$, no row interchanges are required in GEPP and that there will be large elements in U if $b\omega^{n-1}/C$ is large.

As a specific example let $k = 1$, $L = 40$, and $C = 6$. An application producing such numbers would be a solution mixture problem over the time period $-40 \le \hat{t} \le 0$ where fluid with a solute concentration 1 enters a tank of volume 1 at a rate 1, where mixed fluid leaves at a rate 1, where the ratio of the final to the initial amount of solute in the tank is 6 and where we let $t = -\hat{t}$ to transform the domain to $0 \le t \le 40$. In this example, if $n \ge 61$ we have $kh \le \frac{2}{3}$ and the growth factor is large. For $n = 61$, say, the condition number of A is 88, the relative error in the calculated solution when solving (2.3) by a QR factorization is 1.1%. Yet if partial pivoting is used to solve (2.3) then due to a growth factor of 1.28×10^{17} the relative error is 860%. GEPP *fails again for this concrete physical example.*

From the above decomposition it follows that when $C = 1$, say, and $kh = 2/3$ that the growth factor is $(2/3)(2^{n-1} - 1)$. This is quite close to the maximum theoretical growth factor of 2^{n-1}. In comparison, the maximum growth factor reported in [Wri] is proportional to $(\sqrt{2})^n$. We might also note that we can generalize the results for our second example to boundary value problems for systems of m differential equations. For such systems we can get growth factors, approximately, as big as $[2(1.5)^m - 1]^{(n/m)-1}/[3(1.5)^m - 3]$ where large growth occurs in the last m columns of U. For example if $m = 5$ and $n = 90$, growth of 2×10^{18} can occur in the last five columns of U.

4. Conclusions. The existence of practical examples where partial pivoting fails leads to a number of questions for the scientific computing community. To initiate a debate we propose the following answers.

1. *Why haven't practical examples where partial pivoting fails been reported earlier?* We expect that the primary reason is that such examples are indeed rare. Our problems and our approach to solving these problems were carefully selected. However, we should note that most software packages in numerical linear algebra do not report information about the growth factor and so it is possible that large growth factors do occur from time to time in practice, but have gone unreported.

2. *Should widely used packages in numerical linear algebra provide information about growth factors?* In view of our examples we believe that such information should be reported if, for example, a user requests an "expert" solution. The authors of Lapack are planning to incorporate this in future releases of Lapack [Dem]. For a dense matrix a bound on the effect of the growth factor on the error in the calculated solution can be determined in $O(n^2)$ operations [CG], [ER], [GVL] which is small compared to the work in factoring the matrix.

Both Linpack [DBMS] and Lapack [ABB] had difficulty with the matrices in our examples. For the second example in §3, if $L = 60$, $n = 100$, $k = 1$, and $C = 6$ Linpack DGECO reports an estimated condition number of 1.5×10^{12} when the actual condition of A is 132. On the other hand, if $L = 40$ and $n = 61$ Linpack DGECO reports an estimated condition number of 534, which is much closer to 88, the true condition number. However, this might lead the user into thinking that the calculated solution is correct while, as we have seen in §3, it is not. With Lapack for our examples, if the growth factors were not too large, then the "expert" routine DGESVX successfully uses iterative refinement to overcome the inaccuracy due the large growth factor. However, if the growth factor is sufficiently large, iterative refinement does not converge. Also the Lapack condition estimator fails in some cases. Neither package warns the user that GEPP is unstable due to a large growth factor.

For our examples, the reason that the Linpack and Lapack condition estimators fail is that they rely on the ability to solve $Ax = y$. However due to large growth factors, the solutions to $Ax = y$ are not calculated correctly. The underlying condition estimators are not failing themselves, rather they are working with incorrect solutions to $Ax = y$.

3. *Are tests on random matrices useful for comparing and analyzing algorithms in numerical linear algebra?* Such tests are valuable in that they allow the quick generation of many examples. Also random matrices can be amenable to theoretical analysis. However tests with random matrices are not sufficient. For example, the matrices in our illustrations contained only negative numbers below the diagonal. If random matrices were generated so that signs of elements were random, then for $n = 50$, say, the probability of this sign pattern is negligible (10^{-737}). It would never show up in random sampling. We feel that tests with random matrices are often overused.

4. *Is there a need for a collection of test matrices arising from practical problems?* Since there are phenomena that occur in practice that do not show up in tests with random matrices, there is such a need. The collection could complement existing collections such as the one in [DGL]. Ideally the new collection would be in an easy-to-use format such as MATLAB m files so that the collection is compact and flexible code could be included that would generate the matrices for different parameter choices, similar to the style used by Higham [Hig]. However, the focus would be on interesting

matrices that can arise in practice. The collection of examples in Hansen's Regularization Tools [Han] would be a good beginning. Indeed the genesis of this paper came from observing some of the sign patterns for the matrices in Hansen's collection.

REFERENCES

[ABB] E. ANDERSON, Z. BAI, C. H. BISCHOF, J. W. DEMMEL, J. J. DONGARRA, J. J. DU CROZ, A. GREENBAUM, S. J. HAMMARLING, A. M. MCKENNEY, S. OSTROUCHOV, AND D. SORENSON, *Lapack User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[Bak] C. T. H. BAKER, *The Numerical Treatment of Integral Equations*, Oxford University Press, Oxford, 1977.

[Bur] T. A. BURTON, *Volterra Integral and Differential Equations*, Academic Press, New York, 1983.

[CG] E. CHU AND J. A. GEORGE, *An Algorithm to Estimate the Error in Gaussian Elimination Without Pivoting*, Tech. report CS-84-21, University of Waterloo, Ontario, 1984.

[Dem] J. W. DEMMEL, private communication, June, 1993.

[DBMS] J. J. DONGARRA, J. R. BUNCH, C. B. MOLER, AND G. W. STEWART, *Linpack User's Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1979.

[DGL] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.

[DM] L. M. DELVES AND J. I. MOHAMED, *Computational Methods for Integral Equations*, Cambridge University Press, Cambridge, 1985.

[ER] A. M. ERISMAN AND J. K. REID, *Monitoring the stability of the triangular factorization of a sparse matrix*, Numer. Math., 22 (1974), pp. 183–186.

[GVL] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations,* 2nd ed., Johns Hopkins University Press, Baltimore, 1989.

[Han] P. C. HANSEN, *Regularization Tools*, Report UNIC-92-03, Danish Computing Center for Research and Education, Technical University of Denmark, 1992.

[Hig] N. J. HIGHAM, *Algorithm* 694: *A collection of test matrices in MATLAB*, ACM Trans. Math. Software, 17 (1991), pp. 289–305.

[HH] N. J. HIGHAM AND D. J. HIGHAM, *Large growth factors in Gaussian elimination with pivoting*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 155–164.

[Jer] A. J. JERRI, *Introduction to Integral Equations with Applications*, Marcel Dekker, New York, 1985.

[Linz] P. LINZ, *Analytical and Numerical Methods for Volterra Equations*, Society for Industrial and Applied Mathematics, Philadelphia, 1985.

[TS] L. N. TREFETHEN AND R. S. SCHREIBER, *Average-case stability of Gaussian elimination*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 335–360.

[Wil] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

[Wri] S. J. WRIGHT, *A collection of problems for which Gaussian elimination with partial pivoting is unstable*, SIAM J. Sci. Statist. Comput., 14 (1993), pp. 231–238.

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 4, pp. 1363–1380, October 1994

021

A HYBRID TRIDIAGONALIZATION ALGORITHM FOR SYMMETRIC SPARSE MATRICES*

IAN A. CAVERS†

Abstract. This paper considers sequential direct methods for the reduction of a sparse symmetric matrix to tridiagonal form using sequences of orthogonal similarity transformations. Currently, the best published approach combines a bandwidth-reducing preordering algorithm, perhaps Gibbs–Poole–Stockmeyer (GPS), with a band-preserving tridiagonalization such as the EISPACK BANDR routine. This paper introduces a new hybrid tridiagonalization algorithm, BANDHYB, based upon band-preserving reduction techniques that rearrange the elimination sequence of nonzero entries to take better advantage of the sparsity within the band of the permuted matrix. For many practical sparse problems a GPS-BANDHYB approach substantially reduces the CPU requirements of tridiagonalization, compared to a GPS-BANDR scheme, without increasing storage requirements. Over a wide range of sparse problems the new algorithm reduced CPU time by an average of 31%, with reductions of more than 60% observed for some problems.

Key words. sparse matrices, tridiagonalization

AMS subject classifications. 65F50, 65F30

1. Introduction.

This paper considers methods for the similarity reduction of a large $n \times n$ sparse symmetric matrix A to a symmetric tridiagonal matrix T. The class of algorithms studied are sequential direct methods of the generic form

$A_0 := A$
FOR $i := 1, 2, \ldots, k$
$\quad A_i := Q_i^T A_{i-1} Q_i$

in which A is systematically reduced to tridiagonal form ($A_k = T$) using a sequence of k carefully selected orthogonal similarity transformations $Q_i^T A_{i-1} Q_i$.

For symmetric dense eigenvalue problems, tridiagonalization is an important intermediate step used in many solution methods [15]. The tridiagonalization of such problems is typically conducted using either Householder or Givens reduction [21] requiring $O(n^3)$ flops.[1] When a matrix is sparse, however, both standard tridiagonalization algorithms quickly destroy sparsity. In fact, for most sparse problems of interest, little advantage can be made of sparsity with such algorithms. The reduction requires $O(n^3)$ flops and $O(n^2)$ storage, and the original matrix might as well have been treated as dense [7]. This paper reexamines the sparse tridiagonalization process, developing a new hybrid algorithm based on Givens transformations, that reorders the elimination sequence of nonzero entries to improve sparsity exploitation. This hybrid algorithm is better suited to the reduction of general sparse symmetric matrices than previous alternatives to the standard Householder or Givens reductions.

In the past 20–25 years, sparse matrix research has paid particular attention to the direct solution of sparse linear systems, producing many clever methods that successfully exploit matrix sparsity. (See [5] for example.) Attempts to extend this success to sparse tridiagonalization have been hindered by the higher levels of fill associated with orthogonal similarity transformations. Section 2 of this paper presents two previous attempts to overcome the difficulties associated with the tridiagonalization of

* Received by the editors April 6, 1992; accepted for publication (in revised form) August 18, 1993.

† Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z4 (cavers@cs.ubc.ca).

[1] Following [15] a flop is defined to be any floating point arithmetic operation.

general sparse symmetric matrices. Section 3 of the paper introduces our bandwidth contraction algorithm that reorders the elimination of nonzero entries to control the introduction of fill and take better advantage of sparsity. In §4, bandwidth contraction is combined with a column-oriented, band-preserving reduction to produce an effective hybrid tridiagonalization algorithm. Section 5 then describes implementations of the bandwidth contraction and hybrid tridiagonalization algorithms. This section includes an extensive comparison between both sparse algorithms and the EISPACK BANDR [8] routine on a large number of sparse problems from the Harwell–Boeing collection [6]. The hybrid tridiagonalization algorithm is shown to substantially reduce the computational requirements for most sparse problems. Finally, §6 provides concluding remarks and directions for future research.

2. Existing sparse tridiagonalization methods. This section explores existing direct tridiagonalization approaches, analyzing their limitations and their potential extension to improve sparsity exploitation. After a brief discussion of the role of Givens rotations in sparse tridiagonalization, we review sparse tridiagonalization schemes based on customized Givens reductions. We then explore an alternative band-preserving tridiagonalization approach for banded matrices, including a discussion of its extension for the tridiagonalization of general sparse symmetric matrices.

2.1. Givens rotations and sparse tridiagonalization. Givens rotations, $G(i,j,\theta)$, are often chosen for sparse matrix application because they permit fine-grained control over the elimination of nonzeros and the introduction of fill entries. In fact, rotations can be constructed to zero any entry of a matrix using *any* other entry in its row or column as the *zeroing entry.* (In Fig. 1, x_1 is the zeroing entry.) The

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 & x & & x & x & \\ x_2 & & x & x & & x \end{bmatrix} \longrightarrow \begin{bmatrix} \hat{x}_1 & x & x & x & x & x \\ 0 & x & x & x & x & x \end{bmatrix}$$

FIG. 1. *Sparsity structure unioning by a Givens rotation.*

trivial example in Fig. 1 illustrates an important property of rotations that impacts matrix sparsity. If cancellation is ignored and x_1 is nonzero, the sparsity structure of both modified rows is the union of the structure of the two rows prior to the rotation's application. (In the special case of x_2 or x_1 zero, the rotation is either the identity or it swaps the sparsity structure of the two rows, avoiding the introduction of fill entries. It is advantageous to have a zero *zeroing entry.* In addition to fill avoidance, swapping matrix rows to eliminate nonzero x_2 requires no floating point computations.)

The algorithms in this paper use Givens rotations to construct orthogonal similarity transformations of the form $G(i,j,\theta)^T A\, G(i,j,\theta)$, which modify both rows and columns i and j of A. By symmetry of the sparse problems and similarity transformations, all algorithms need only to consider transformation modifications to either the upper or lower triangular portion of each matrix. Without loss of generality we work with a matrix's lower triangular portion.

From the wide selection of rotations that can be constructed to eliminate a particular nonzero entry, this paper concentrates on algorithms using *adjacent rotations.* A rotation $G(i,j,\theta)^T$ is considered adjacent if $|i-j| = 1$. As discussed in §6, future study will consider the use of nonadjacent rotations.

2.2. Customized sparse Givens reduction. Using $O(n^3)$ flops and $O(n^2)$ storage, the standard Givens reduction tridiagonalizes a symmetric matrix column by column as shown in Fig. 2. Within each column nonzero entries are eliminated

```
FOR col:= 1 TO n-2 DO
    FOR row:= n DOWNTO col+2 DO
        A := G(col + 1, row, θ)^T A G(col + 1, row, θ)
```

FIG. 2. *Givens reduction.*

from the bottom up using the column's subdiagonal entry as the zeroing entry. Dense Givens rotations, which assume that the unreduced portion of the matrix is dense, are used throughout.

The flexibility of Givens rotations permits both the order in which a column is zeroed and the plane of the zeroing rotation to be modified. These variable elements of the basic algorithm can be used to construct customized sparse Givens reduction algorithms that attempt to take better advantage of sparsity. Unfortunately, the experimentation of Duff and Reid [7] shows that, independent of rotation plane selection and the elimination order of each column's nonzeros, adaptations of Givens column-by-column reduction for large sparse matrices generally experience overwhelming levels of fill and matrix sparsity is quickly destroyed. In fact, Duff and Reid conclude that if a Givens reduction approach is taken, typically little advantage is made of sparsity and it is preferable to treat the matrix as dense.

2.3. The tridiagonalization of banded matrices. If tridiagonalization algorithms are to effectively utilize matrix sparsity, it appears essential to restrict the accumulation of fill entries to some maintainable substructure of the matrix. Suppose that a symmetric matrix of bandwidth[2] b is to be reduced to tridiagonal form. In addition, for the remainder of this subsection assume that the band is dense. Applying a column-by-column Givens reduction to the matrix leads to overwhelming levels of fill outside the band, quickly destroying matrix sparsity. Alternatively, the algorithm of Rutishauser [18] and Schwarz [20] (subsequently referred to as the Rutishauser–Schwarz or, simply, R-S algorithm) controls the encumbering effects of fill by actively preserving a matrix's banded structure throughout tridiagonalization.

The pseudocode in Fig. 3 outlines the Rutishauser–Schwarz algorithm. Once again, Givens transformations are used by this band-preserving tridiagonalization algorithm to provide fine-grained control over the elimination process. Globally, the

```
FOR col:= 1 TO n-2 DO
    FOR diag:= min(b,n-col) DOWNTO 2 DO
        /*Zero A_{col+diag,col}.*/
        A := G(col + diag, col + diag − 1, θ)^T A G(col + diag, col + diag − 1, θ)
        IF bandwidth(A) > b THEN
            Annihilate bulges with additional adjacent Givens transformations.
```

FIG. 3. *The Rutishauser–Schwarz algorithm.*

effect is that the banded matrix is reduced column by column. Indeed, within each column, adjacent transformations eliminate the nonzeros from the outside in. The key difference is that R-S immediately removes fill created outside the band of the

[2] Bandwidth (or semibandwidth) is defined as $b = \max_{i,j\in\{1\ldots n\},i\neq j} |i-j|$ such that $A_{ij} \neq 0$.

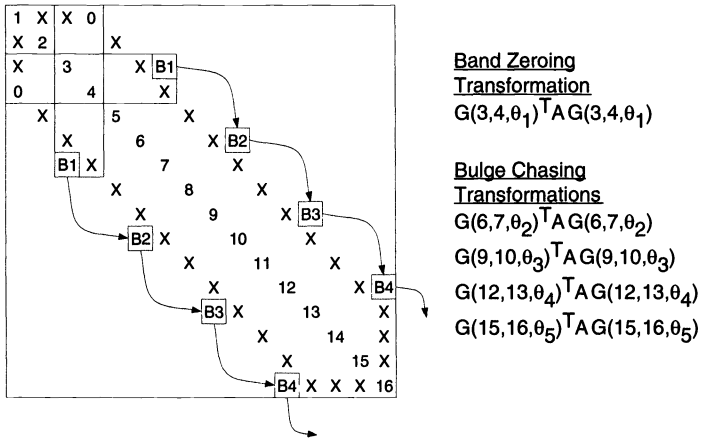


FIG. 4. *Bulge chasing to preserve bandwidth.*

original matrix. The symmetric elimination of a band nonzero produces a pair of fill entries, or *bulges*, outside the band as illustrated by the B1 entries in Fig. 4. Before eliminating the next band nonzero R-S chases the bulges off the end of the matrix with an additional sequence of adjacent transformations. (See Fig. 4.) In this fashion the algorithm maintains the banded structure of the unreduced portion. During the reduction of a typical column k, the elimination of band entry A_{ik} requires $\lceil \frac{n-b-i+1}{b} \rceil$ bulge-chasing transformations. Adjacent transformations are used exclusively by the algorithm because nonadjacent rotations create triangular bulges consisting of several nonzero entries that require a larger number of bulge-chasing transformations.

Table 1 provides formula for the tridiagonalization costs of the Rutishauser–Schwarz algorithm. The analysis assumes $2 < b \leq (n/2 - 1)$ and $C_{\text{R-S}}$ is the non-analytic term $\text{Mod}(n-1,b)(\text{Mod}(n-1,b)-b)$, which typically can be safely ignored without incurring large errors.[3] $F^{\text{G}}_{\text{R-S}}$ and $T_{\text{R-S}}$ refer to flop and transformation counts, respectively. In addition to $F^{\text{G}}_{\text{R-S}}$, the construction of each transformation

TABLE 1
Tridiagonalization costs of the Rutishauser–Schwarz algorithm for a densely banded matrix.

$F^{\text{G}}_{\text{R-S}}$	$T_{\text{R-S}}$
$(6b - \frac{8}{b} + 2)n^2 - (6b^2 + 5b - \frac{16}{b} + 5)n$ $+2b^3 + 4b^2 - b - \frac{8}{b} + 3 + (\frac{8}{b} + 3)C_{\text{R-S}}$	$\frac{(b-1)(n-1)^2 + C_{\text{R-S}}}{2b}$

requires one square root. Although $T_{\text{R-S}}$ is comparable to the number of transformations used by Givens reduction on the same densely banded problem, in general the band-preserving approach modifies fewer nonzero entries with each transformation. As a result, for matrices of moderate bandwidth, $F^{\text{G}}_{\text{R-S}}$ is smaller than the floating-point requirements of Givens reduction.

The EISPACK [8] implementation of the R-S algorithm BANDR does not use standard Givens transformations. Instead it employs a "root free" variant of the transformation, with identical fill properties, often referred to as a *fast Givens trans-*

[3] $Mod(x, y)$ is the remainder from the division of integer x by integer y.

formation [9]. In this case the computational requirements of the band-preserving tridiagonalization are reduced to

$$F_{\text{R-S}}^{\text{FG}} = \left(4b - \frac{10}{b} + 6\right) n^2 - \left(4b^2 + 3b - \frac{20}{b} + 10\right) n + \frac{4b^3}{3} + \frac{5b^2}{2} - \frac{5b}{6} - \frac{10}{b} + 7 + \left(\frac{10}{b} + 2\right) C_{\text{R-S}} \tag{1}$$

and n square roots. The analysis leading to $F_{\text{R-S}}^{\text{FG}}$ ignores the cost of periodic rescaling and assumes that a reduction uses the two types of fast Givens transformations in equal proportion. For either variant of the band-preserving R-S algorithm $O(bn)$ storage is required.

We note that BANDR will not be faster for all bandwidths than the best dense matrix algorithm, Householder's reduction. Assuming that the Householder reduction requires $\frac{4}{3}n^3$ flops [15], while BANDR requires $4bn^2$, the dense matrix algorithm requires fewer flops for bandwidths larger than $\frac{n}{3}$. However, the selection of a tridiagonalization algorithm is often strongly influenced by the lower storage requirements of the Rutishauser–Schwarz algorithm.

2.4. Generalization of band-preserving tridiagonalization techniques. For general sparse symmetric matrices we can extend the band-preserving techniques of Rutishauser and Schwarz to form the following two-stage tridiagonalization algorithm.

1. $A := P^T AP$, where P is a bandwidth reducing permutation matrix.
2. Tridiagonalize A using the R-S algorithm.

Many heuristic algorithms have been suggested for the identification of bandwidth reducing preorderings [3], [14], but the most widely accepted algorithms are RCM [11] and GPS [13], [17]. The experience of Gibbs, Poole, and Stockmeyer [14] and our experimentation with the Harwell–Boeing test matrix collection suggest that GPS most frequently provides the smallest bandwidth preorderings over a wide range of problems. Consequently, GPS was selected for the experimentation in §5 and will be referenced as the *bandwidth-reducing preordering* in subsequent discussion. Although fill entries may accumulate during the tridiagonalization of the permuted matrix $P^T AP$, the R-S algorithm uses bulge-chasing transformations to constrain them to the band.

Limited advantage of sparsity is taken if we treat the band of the preordered matrix as dense during the tridiagonalization stage of the extended algorithm. In this case complete reliance is placed upon GPS to exploit sparsity, but for many problems the band of the preordered matrix is relatively sparse prior to reduction. (See Table 2.) To take advantage of band zeros, three modifications could be made to the basic band-preserving tridiagonalization algorithm.

1. Avoid constructing and applying transformations to eliminate band or bulge entries that are already zero.

2. Exploit *zeroing entries* (see §2.1) that are zero by performing row and column exchanges instead of using the general form of the Givens transformation.

3. Apply each nontrivial transformation to only those lower triangular entries whose column(row) index is in the unioned sparsity structure of the two modified rows(columns).

Both the Schwarz code [20] and EISPACK BANDR [8] check if the bulge or band entry is already zero before performing an elimination. These codes, however, are

TABLE 2
Band filling characteristics of sparse R-S.

Problem	n	Bandwidth	Initial off-diagonal band density	No. of columns (rows) eliminated before band is full.
5-pt problems	b^2	b	$\sim 2/b$	b-1
PLAT1919	1919	80(GPS)	10.1%	31
NOS3	960	65(GPS)	12.4%	19

primarily intended for densely banded matrices and neither incorporates the second or third modification. Enhancing the two-stage tridiagonalization algorithm by all three sparsity modifications produces a new algorithm subsequently referred to as the sparse Rutishauser–Schwarz algorithm or simply sparse R-S.

Unfortunately, the unreduced portion of a typical sparse matrix's band still fills quickly during band-preserving tridiagonalization. As an example, consider applying sparse R-S to five-point problems created by discretizing partial differential equations on a uniform $b \times b$ square grid with Dirichlet boundary conditions. The problems are ordered with a standard lexicographic labeling, minimizing bandwidth at b. After $b - 1$ columns of a five-point problem have been reduced to tridiagonal form, the remainder of the band is completely filled in. The preponderance of fill within the band is largely due to the sequence of bulge-chasing rotations required by each band nonzero's elimination. Once the matrix's band has been filled, there is no further opportunity to exploit sparsity beyond the densely banded form of the remaining submatrix. Consequently, the flop and transformation requirements of a five-point problem reduction are identical in the highest order terms to $F^{\mathrm{G}}_{\text{R-S}}$ and $T_{\text{R-S}}$ for a densely banded matrix of equivalent order and bandwidth.

The speed with which the band of a five-point problem fills is typical of most large sparse symmetric problems. Table 2 provides fill data for two additional sparse problems from the Harwell–Boeing test matrix collection [6] as tridiagonalized by sparse R-S. Despite starting with relatively sparse bands, the unreduced portion of the band of both problems is completely filled well before b columns are reduced to tridiagonal form.

Because of high levels of band fill, typically very little advantage can be taken of sparsity in the band. In general, the application of sparse R-S is only slightly superior to the original band-preserving R-S tridiagonalization approach. In the next section we introduce an alternative to sparse R-S that more successfully utilizes band sparsity.

3. Bandwidth contraction. The previous section demonstrated the inability of sparse R-S to take good advantage of internal band sparsity. This section presents an alternative approach to sparse tridiagonalization. It also uses bandwidth-reducing preorderings and band-preserving reduction techniques, but reorders the elimination sequence to more fully exploit band sparsity.

3.1. Motivation. The sparse tridiagonalization techniques explored in this section are motivated by the following observation. A bandwidth-reducing preordering frequently produces a permuted matrix whose profile consists of varying length *spikes* of nonzeros extending from the main diagonal. The longest spike defines the bandwidth of the permuted matrix as shown in Fig. 5. For many practical problems, the spikes of the permuted matrix are of dramatically different lengths. The new sparse tridiagonalization approach attempts to exploit this characteristic. Although fill cannot be avoided, the bandwidth of the matrix could be significantly reduced

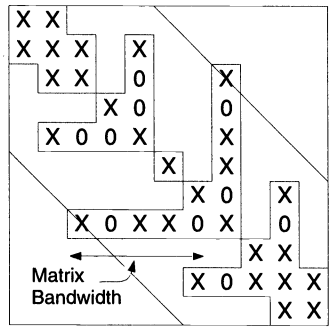


FIG. 5. *Matrix bandwidth and spike length.*

with relatively few transformations if the ends of the longest spikes could be clipped off at low cost before the contracted band becomes dense.

3.2. The sparse bandwidth contraction algorithm. There are a number of ways that the ends of the profile's longest spikes could be clipped off with Givens transformations. One way to approach the clipping process is to rearrange the elimination order of a band-preserving tridiagonalization so that the matrix is reduced to tridiagonal form diagonal by diagonal (outermost diagonal first), rather than column by column as in sparse R-S. A diagonally oriented band-preserving tridiagonalization for densely banded symmetric matrices has been previously considered [19], [21], but was superseded on sequential machines by the Rutishauser–Schwarz column-oriented reduction [18], [20]. (See §4.1 for a detailed comparison of algorithm complexity.) To our knowledge, however, no one has considered the relative merits of the two reduction paradigms extended for general application to sparse symmetric matrices. (For densely banded matrices, the LAPACK [1] project considered the relative merits of these two algorithms implemented on vector machines.)

As shown in Fig. 6, our bandwidth contraction algorithm uses the diagonally oriented spike-clipping process to completely tridiagonalize a sparse symmetric matrix.

1. $A := P^T A P$, where P is a bandwidth-reducing permutation matrix.
2. $b :=$ bandwidth(A)
3. FOR $\widehat{b} := b$ DOWNTO 2 DO /*Tridiagonalize A.*/
   FOR col := 1 TO $n - \widehat{b}$ DO
   IF $A_{\text{col}+\widehat{b},\text{col}} \neq 0$ THEN /*Zero $A_{\text{col}+\widehat{b},\text{col}}$.*/
   IF $A_{\text{col}+\widehat{b}-1,\text{col}} = 0$ THEN
   Exchange rows/columns $(\text{col} + \widehat{b})$ and $(\text{col} + \widehat{b} - 1)$ in A.
   ELSE
   $A := G(\text{col} + \widehat{b}, \text{col} + \widehat{b} - 1, \theta)^T \, A \, G(\text{col} + \widehat{b}, \text{col} + \widehat{b} - 1, \theta)$
   (Exploit band sparsity of modified rows and columns.)
   IF bandwidth(A) $> \widehat{b}$ THEN
   Chase bulges with additional adjacent Givens
   transformations or row/column exchanges.
   ENDIF /*Outermost IF*/

FIG. 6. *The bandwidth contraction algorithm.*

To properly exploit band zeros, the three modifications made to the basic Rutishauser–Schwarz algorithm in §2.4 are also incorporated into the bandwidth contraction algorithm.

We begin with the matrix symmetrically permuted to reduce bandwidth. Then, starting with $A_{1+b,1}$, the band's outermost diagonal is scanned for its first nonzero entry. This entry is eliminated using an adjacent Givens transformation or a row/column exchange. If a nonzero entry is created beyond the current bandwidth, the bulge is chased off the end of the matrix as described in §2.3. The scanning and reduction of the outermost diagonal continues until all nonzeros have been eliminated. At this point the current bandwidth of the matrix is reduced by one and the reduction process continues with the next diagonal.

As for the sparse Rutishauser–Schwarz algorithm, bandwidth contraction uses adjacent transformations exclusively, avoiding the creation of multiple entry bulges and the concomitant extra bulge-chasing transformations. Triangular bulges not only increase computational costs, they also accelerate the introduction of fill entries into the band. An added complication for a diagonally oriented tridiagonalization is that nonadjacent transformations may reintroduce fill entries in previously zeroed positions of the diagonal. This discussion, however, is not intended to completely rule out the use of nonadjacent transformations. There are special sparsity patterns for which nonadjacent transformations are beneficial. Future study will explore the potential role of nonadjacent transformations in more sophisticated bandwidth contraction algorithms.

3.3. A demonstration of bandwidth contraction. To illustrate the potential effectiveness of bandwidth contraction, we provide a small contrived example in Fig. 7. The top matrix in Fig. 7 shows the original sparse matrix A with its nonzero entries indicated by an X and the numbered diagonal. It is assumed that A has already been permuted to reduce its bandwidth to 6. Two additional matrices, C and D, illustrate A after a partial reduction by sparse R-S or bandwidth contraction. In both C and D a 0 marks the positions of eliminated band nonzeros. Finally, reported flop counts assume that both reductions employ fast Givens transformations.

Matrix C illustrates A after sparse R-S has reduced its first three columns to tridiagonal form. Despite the highly sparse nature of the original problem, the remainder of the band is almost completely filled. The entire tridiagonalization uses eight row/column exchanges and 132 nontrivial transformations, requiring a total of 7232 flops.

Matrix D illustrates A after bandwidth contraction has eliminated the three outermost nonzero diagonals and contracted the bandwidth to 3. Although the elimination of nonzeros once again produces fill entries within the band relatively quickly, the algorithm is able to efficiently exploit the sparsity of the band away from the main diagonal. For example, bandwidth contraction eliminates the entire sixth and fifth subdiagonals of the band at the relatively low cost of 216 flops, using seven row/column exchanges and four nontrivial transformations. The complete tridiagonalization uses 12 row/column exchanges and 163 nontrivial transformations, requiring a total of 6537 flops. For this example, the computational requirements of tridiagonalization with bandwidth contraction, as measured by flop counts, are approximately 9.6% lower than for the sparse R-S approach.

It is important to note that the number of nontrivial transformations used by a tridiagonalization is a misleading metric of algorithm performance. Bandwidth contraction requires more transformations, but generally fewer nonzeros are modified by each nontrivial transformation, permitting a lower total flop count.

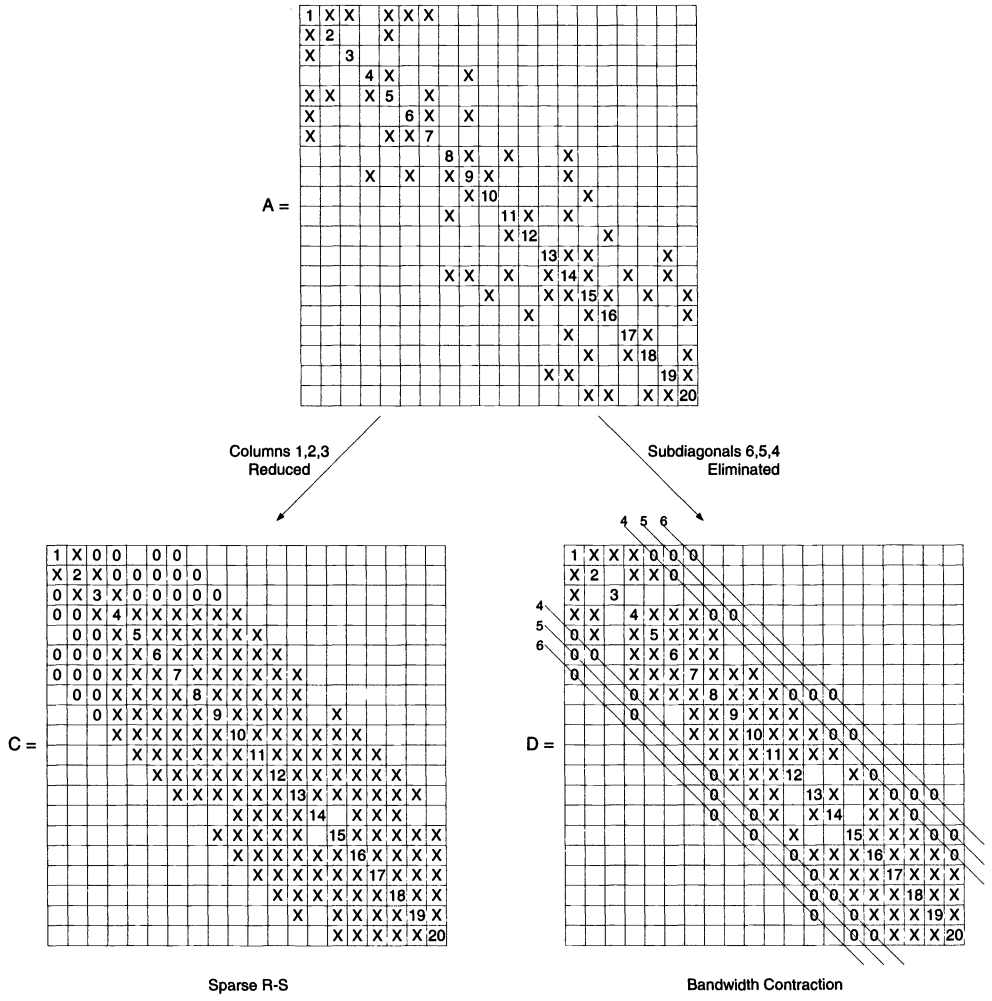


FIG. 7. *A partial tridiagonalization, sparse* R-S *versus bandwidth contraction.*

The key to the success of bandwidth contraction is the elimination of the outermost diagonals at low cost. As the result of several contributing factors, bandwidth contraction prolongs the advantages of sparsity in the outermost subdiagonals. First, when nontrivial adjacent transformations are required, they are often applied to well-separated pairs of rows and columns, insulating the effects of fill from one transformation to the next. In contrast, the sparse R-S initial band zeroing transformations and associated bulge-chasing transformations are applied to groups of neighboring rows and columns. These initial transformations produce a cascade of fill entries, typically not observed for bandwidth contraction, which quickly fills the band despite a matrix's initial sparsity. Second, the sparsity of the outermost diagonals permits bandwidth contraction to cheaply eliminate many of the band nonzeros and associated bulges, without fill, using row/column exchanges. In addition, the sparsity of the outermost diagonals often reduces the number of transformations required by bulge-chasing sequences. Finally, as the nonzeros of the outermost diagonal are eliminated, bulge-chasing sequences must shorten, while the length of the sequences used by sparse

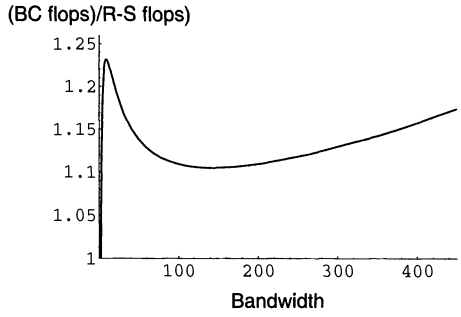


FIG. 8. *The flop requirements of* BC *relative to* R-S *for a densely banded matrix,* $n = 1000$.

R-S remains relatively constant as it reduces the first few columns. Consequently, the initial stage of bandwidth contraction produces fewer band fill entries. Thus, the partial tridiagonalization often significantly contracts the matrix's bandwidth before producing a densely banded intermediate matrix.

In general, the relative success of the sparse tridiagonalization algorithms depends on problem-specific sparsity structures. The extensive experimental analysis of §5 confirms the relative advantage experienced by bandwidth contraction for many practical sparse problems.

4. A hybrid tridiagonalization algorithm.

4.1. Motivation. Two metrics of tridiagonalization algorithm cost are flop and transformation counts F and T. As demonstrated in the previous section, the bandwidth contraction algorithm may be able to significantly reduce the bandwidth of a sparsely banded matrix at relatively low cost. Consequently, for sparsely banded matrices, bandwidth contraction flop counts are smaller than for sparse R-S, despite larger transformation counts. If the band of a matrix is dense, however, the Rutishauser and Schwarz column-oriented band-preserving tridiagonalization is superior in both measures of work.

Table 3 provides formulas for the tridiagonalization costs for the fast Givens bandwidth contraction variant applied to a densely banded, symmetric matrix of bandwidth b. The analysis assumes that $b < (n+1)/2$ and that an equal proportion of the two types of fast Givens transformations are employed. The analysis ignores the potential cost of the periodic rescaling required by fast Givens transformations. C_{BC} is the nonanalytic term $\text{Mod}(n,k)(k - \text{Mod}(n,k))$. When $b \ll n$, the $\text{Mod}(n,k)$ terms

TABLE 3
Tridiagonalization costs of the bandwidth contraction algorithm for a densely banded matrix.

$F_{\text{BC}}^{\text{FG}}$	T_{BC}
$\left(4b - 4 + 10\sum_{k=2}^{b}\left(\frac{1}{k}\right)\right)n^2 + \left(22 - 16b - 3b^2\right)n$ $+\frac{2b^3}{3} + \frac{5b^2}{2} + \frac{11b}{6} - 5 + \sum_{k=2}^{b}\left(\left(\frac{10}{k} + 2\right)C_{\text{BC}}\right)$	$\frac{n^2\sum_{k=2}^{b}\left(\frac{1}{k}\right) - (b-1)n + \sum_{k=2}^{b}\frac{C_{\text{BC}}}{k}}{2}$

can be safely ignored without incurring significant errors. Comparison of $F_{\text{BC}}^{\text{FG}}$ and $F_{\text{R-S}}^{\text{FG}}$, from (1) in §2.3, shows that the flop requirements of bandwidth contraction are

larger than those for the R-S algorithm applied to the same problem. To demonstrate the potential difference in tridiagonalization costs, Fig. 8 plots the flop requirements of bandwidth contraction, normalized by $F^{\mathrm{FG}}_{\mathrm{R-S}}$, against bandwidth for a densely banded matrix.

While $F^{\mathrm{FG}}_{\mathrm{BC}}$ is typically 10–25% larger than $F^{\mathrm{FG}}_{\mathrm{R-S}}$, for problems with nontrivial bandwidth, the difference between T_{BC} and $T_{\mathrm{R\text{-}S}}$ can be much greater. At first glance there may seem to be an inconsistency in our analysis. As mentioned in §3.3, however, transformation counts are generally a misleading metric of tridiagonalization costs. $F^{\mathrm{FG}}_{\mathrm{R\text{-}S}}$ and $F^{\mathrm{FG}}_{\mathrm{BC}}$ are closer than predicted by T_{BC} and $T_{\mathrm{R\text{-}S}}$ because, as bandwidth contraction reduces a matrix's bandwidth, the number of nonzeros modified by each transformation generally declines. The computational effort of applying later transformations is reduced, while for the R-S algorithm the cost of each transformation remains relatively constant.

As an aside to our discussion of sequential algorithms, vector machines complicate the relative efficiency analysis of tridiagonalization algorithms. In general, vector machines put more weight on T relative to F. During the development of the LAPACK [1] replacement for BANDR, SSBTRD, several band-preserving tridiagonalization algorithms were tested on vector machines [4]. In very general terms, for small n (less than 50) or matrices with moderate bandwidth ($20 \leq b < 50$), it was found that vectorized code based on a diagonally oriented elimination is the fastest approach. For other densely banded matrices, variants of a column-oriented tridiagonalization are more efficient. Emphasizing the importance of good performance for large n and small bandwidth, SSBTRD is based on the column-oriented, vectorized algorithm of Kaufman [16].

4.2. The hybrid tridiagonalization algorithm. The observations of the previous subsection suggest a hybrid tridiagonalization algorithm. While the band of the resulting matrix remains sparse, the hybrid algorithm employs the bandwidth contraction scheme. When some measure of band density or "fullness" exceeds a specified threshold, the reduction switches to sparse R-S to complete the tridiagonalization. To avoid the redundant elimination of band nonzeros, the hybrid algorithm always completes the reduction of a nonzero diagonal before switching to sparse R-S. Vector machines complicate the transition decision. This paper concentrates on the sequential algorithm.

The most sensitive design issue for this hybrid algorithm is the selection of a metric for measuring band fullness and the choice of a threshold value. There are many ways to determine when the contracted band is dense or nearly so. The most obvious choice is to directly monitor the number of nonzero entries in the band. Instead, we suggest regulating the transition between bandwidth contraction and sparse R-S by a threshold on the number of nonzero entries in the outermost nonzero subdiagonal. The transition is made when the number of nonzeros is greater than some fraction of the subdiagonal's length. As shown below, monitoring the number of nonzeros in the next subdiagonal can be integrated cheaply into bandwidth contraction. Equally important, this transition regulation technique is a good approximation of band density. As shown in §5, sparsity within the band is best exploited at the transformation level, which is controlled by the zeros in the outermost diagonal. In addition, once the outermost diagonal is full, or nearly so, each successive subdiagonal eliminated will have a similar density and the band will quickly fill.

The pseudocode in Fig. 9 describes the hybrid tridiagonalization algorithm. Let *threshold* be the fraction of the outermost diagonal that must be nonzero before the

1. $A := P^T AP$, where P is a bandwidth-reducing permutation matrix.
2. $b :=$ bandwidth(A)
3. /*Initialize *nzcnt* for the outermost diagonal.*/
   nzcnt := 0
   FOR i:= 1 TO $n - b$ DO
      IF $A_{i+b,i} \neq 0$ THEN *nzcnt*:= *nzcnt*+1
4. (a) $\widehat{b} := b$
   (b) /*While the matrix is not tridiagonal and the threshold has not been*/
   /*met, eliminate the outermost nonzero diagonal.*/
   WHILE ($(\widehat{b} \geq 2)$ AND (*nzcnt* < (threshold * $(n - \widehat{b})$))) DO
      i. *nzcnt* := 0
      ii. FOR col := 1 TO $n - \widehat{b}$ DO
         IF $A_{\text{col}+\widehat{b},\text{col}} \neq 0$ THEN /*Zero $A_{\text{col}+\widehat{b},\text{col}}$.*/
            IF $A_{\text{col}+\widehat{b}-1,\text{col}} = 0$ THEN
               Exchange rows/columns $(\text{col} + \widehat{b})$ and $(\text{col} + \widehat{b} - 1)$ in A.
            ELSE
               $A := G(\text{col} + \widehat{b}, \text{col} + \widehat{b} - 1, \theta)^T \; A \; G(\text{col} + \widehat{b}, \text{col} + \widehat{b} - 1, \theta)$
                  (Exploit band sparsity of modified rows and columns.)
            IF bandwidth(A) > $\widehat{b}$ THEN
               Chase bulges with additional adjacent Givens
               transformations or row/column exchanges.
         ENDIF /*Outermost IF*/
         IF $A_{\text{col}+\widehat{b}-1,\text{col}} \neq 0$ THEN *nzcnt*:= *nzcnt*+1
      iii. IF $A_{n,n-\widehat{b}+1} \neq 0$ THEN *nzcnt*:= *nzcnt*+1
      iv. $\widehat{b} := \widehat{b} - 1$
   (c) IF $\widehat{b} > 1$ THEN complete tridiagonalization with sparse R-S.

FIG. 9. *The hybrid tridiagonalization algorithm.*

transition to sparse R-S is made and let *nzcnt* be the number of nonzeros in the next subdiagonal. The algorithm is able to check the nonzero status of entry $A_{\text{col}+\widehat{b}-1,\text{col}}$ after the elimination of entry $A_{\text{col}+\widehat{b},\text{col}}$ because the entries of row $(\widehat{b} + \text{col} - 1)$ will not be modified again during the reduction of the current outermost diagonal. The resource requirements of the band density metric are minimal—one additional integer variable, and during each diagonal's reduction $n - \widehat{b}$ comparisons and at most $n - \widehat{b} + 1$ integer operations.

4.3. Performance of the hybrid tridiagonalization algorithm. Consider the application of the hybrid tridiagonalization algorithm to matrix A of Fig. 7 with a *threshold* of 0.85. In the first stage of the tridiagonalization, bandwidth contraction reduces the three outermost nonzero subdiagonals, producing matrix D of Fig. 7. The hybrid tridiagonalization algorithm then transfers control to sparse R-S to complete the reduction to tridiagonal form. Table 4 summarizes the computational requirements of all three sparse tridiagonalization algorithms, assuming the use of fast Givens transformations. The hybrid algorithm requires approximately 19% and 10% fewer flops than sparse R-S and bandwidth contraction, respectively. It is interesting to note that the total number of nontrivial transformations required by the hybrid tridiagonalization algorithm is significantly lower than for bandwidth contraction and

TABLE 4
Tridiagonalization summary for a small sparse example.

Method	Row/Column exchanges	Nontrivial Transformations	Flops
Sparse R-S	8	132	7232
Bandwidth contraction	12	163	6537
Hybrid tridiagonalization	12	136	5880

only marginally higher than for sparse R-S.

As an additional example, the five-point problem, with a standard lexicographic ordering, produces matrices for which the bandwidth contraction algorithm is able to take little advantage of band sparsity. The hybrid algorithm, however, detects the inability of a partial bandwidth contraction to exploit matrix sparsity and immediately switches to sparse R-S. The extra costs incurred over a direct application of sparse R-S are insignificant, except for very small problems. Generally, even in the worst case, the hybrid tridiagonalization algorithm is always comparable to sparse R-S.

The test problem of Fig. 7 is obviously a trivial example. The experiments described in the following section, however, show that the hybrid tridiagonalization algorithm dramatically reduces the computational requirements of sparse tridiagonalization for a wide range of sparse problems.

5. Experimentation. This section describes extensive experimentation with implementations of the bandwidth contraction and hybrid tridiagonalization algorithms. After briefly describing the implementations, testing environment, and suite of test problems, we analyze test results comparing the implementations to the EISPACK BANDR.

5.1. Implementation. The implementation of bandwidth contraction BANDCON was created by rewriting the EISPACK Fortran routine BANDR (an R-S code) to perform a sparse, diagonally oriented, band-preserving tridiagonalization. Using this new routine, BANDHYB implements the hybrid tridiagonalization algorithm by augmenting BANDCON with the threshold strategy described and a transition to a modified version of BANDR that omits initializations. The hybrid algorithm switches to a column-oriented scheme when the band is dense or nearly so. Given the speed with which a sparse band fills during an R-S reduction (see §2.4), using a sparse R-S code for this portion of the BANDHYB code is not warranted. Otherwise BANDCON and BANDHYB closely follow the algorithms in §§3.2 and 4.2 with one exception. That is, based on the outcome of the following study, we implement bandwidth contraction with only two of the three sparse algorithm modifications listed in §2.4.

Unlike the first two sparsity modifications listed in §2.4, exploiting the sparsity of a pair of rows or columns during the application of a transformation requires significant overhead. To determine if the potential savings were worthy of the increased overhead, experiments with a symbolic reduction code were conducted. Assuming no cancellation, the program estimates the flop requirements of different tridiagonalization algorithms by manipulating matrix sparsity structures. For 15 larger problems, this code compared the flop requirements of a hybrid tridiagonalization algorithm that fully exploits sparsity with one that treats the band as dense while applying a transformation. Accounting procedures differ between the two simulations, but the sequence of sparsity structures encountered is identical.

Going from dense to sparse transformations, savings of 12–22% in the bandwidth

contraction portion of the reduction are observed for three problems; however, for the remaining matrices, savings are less than 5%. Considering the cost of the entire hybrid tridiagonalization, the potential savings of sparse transformations in the bandwidth contraction are very small. The largest reduction is 1.2% and for the remaining problems the potential savings are 1% or lower. Considering the storage and computational overhead required by a sparse data structure, performing sparse transformations is not beneficial to BANDCON or BANDHYB performance and will not be pursued by the sparse tridiagonalization implementations described in this paper. In future work, however, sparse transformations will be reevaluated for the special case in which a partial bandwidth contraction is the end goal.

As a consequence of the preceding study, both BANDCON and BANDHYB keep the densely banded data structure of BANDR, storing each subdiagonal of the band's lower triangular portion in a separate column of an $n \times (b+1)$ double precision array. As a result, the storage requirements of the three routines are essentially identical and the analysis of §5.3 concentrates on the CPU requirements of each routine.

To improve efficiency, BANDR uses fast Givens transformations instead of classical Givens transformations. Unfortunately, to avoid overflow problems when a tridiagonalization requires a large number of transformations, periodic rescaling is necessary. In BANDCON and the bandwidth contraction portion of BANDHYB, the BANDR rescaling strategies are completely reformulated. This aspect of the implementation will not be given in detail here; in general, the bandwidth contraction algorithm needs more rescaling than the column-oriented tridiagonalization. Fortunately, the computational requirements for rescaling in either approach was insignificant for all large problems tested.

5.2. Test problems and the testing environment. To compare the computational requirements of BANDCON and BANDHYB to BANDR, all three routines are applied to 115 symmetric problems from the Harwell–Boeing sparse matrix collection [6]. The problems range in size from a 24-node problem to the BCSSTK24 problem with $n = 3562$. When nonzero values are not provided by the collection, a random value in the range $(0.0, 1.0]$ is assigned to each nonzero entry. Unless otherwise specified, each problem is preordered to reduce bandwidth using the Lewis implementation of GPS [17].

All testing was conducted on a SUN SPARCstation 2. The reported CPU second timings, produced using the system routine `etime`, include both user and system time. In these experiments the transition to a column-oriented tridiagonalization in BANDHYB is made when the outermost subdiagonal is full. In our experience, *threshold* values less than 1.0 can improve the performance of BANDHYB for some sparse problems. Our research of effective *threshold* selection techniques, however, is inconclusive, so we chose to work with a value of 1.0 in this paper.

5.3. Numerical results. BANDCON and especially BANDHYB are very successful relative to the EISPACK BANDR. For 98 of the 115 problems tested, the hybrid tridiagonalization algorithm significantly reduced CPU requirements. For this group of problems, reductions in CPU time range from a low of 6.6% to a high of 63.3%. For the 70 problems tested with more than 400 nodes, BANDHYB required on average 31.1% fewer CPU seconds than BANDR. The first 20 entries of Table 5 summarize timings of test problems for which BANDHYB is especially successful. For this group BANDHYB exhibited a mean reduction in CPU time of 44.2%. Of the 17 test problems for which BANDHYB shows little or no improvement, 14 are very small problems and three matrices have between 400 and 1000 nodes.

TABLE 5
Selected tridiagonalization timings.

Name	n	GPS BW	Tridiagonalization times (sec) BANDR	BANDCON	BANDHYB	% CPU reduction BANDR→BANDHYB
685 BUS	685	78	67.7	26.4	26.0	61.6
GR 30 30	900	49	77.9	59.2	57.9	25.7
NOS3	960	65	128.6	85.8	83.1	35.4
DWT 1005	1005	106	209.1	88.7	86.8	58.5
CAN 1054	1054	112	237.5	151.7	147.6	37.8
CAN 1072	1072	156	331.9	193.7	189.2	43.0
BCSSTK09	1083	95	234.3	158.5	150.8	35.7
1138 BUS	1138	126	305.3	111.7	112.2	63.3
ERIS1176	1176	100	291.3	118.9	120.1	58.8
DWT 1242	1242	91	278.6	149.2	142.6	48.8
BCSPWR07	1612	103	604.4	233.5	232.8	61.5
BCSPWR09	1723	116	730.4	312.4	313.1	57.1
PLAT1919	1919	80	706.7	440.0	424.6	39.9
BCSSTK26	1922	245	1863.7	1018.0	1001.1	46.3
DWT 2680	2680	65	1103.3	678.7	664.9	39.7
ZENIOS	2873	30	1.25	0.81	0.86	31.2
SSTMODEL	3345	82	1449.8	856.3	857.8	40.8
LSHP3466	3466	61	1790.7	1448.1	1441.8	19.5
BCSSTK24	3562	312	8990.3	5417.2	5328.9	40.7
BCSSTK28	4410	323	14643	9235.5	9118.1	37.7
DWT 361	361	14	3.29	3.94	3.14	4.6

Although the performance of BANDCON is similar to BANDHYB for most of the problems listed in Table 5, for some problems BANDCON is slower than BANDR. As an example, consider the tridiagonalization times of DWT 361 given at the bottom of Table 5. BANDCON requires significantly more CPU time than does BANDR. Such performance degradation is understandable from the theoretical analysis for the tridiagonalization of densely banded matrices provided in §4.1. The hybrid routine BANDHYB, however, always has comparable CPU requirements to BANDR in the worst case; for the majority of sparse problems it is significantly faster.

For 40 of the 115 problems tested, BANDHYB, with a *threshold* of 1.0, uses bandwidth contraction for the entire tridiagonalization process. The band of many of these problems becomes quite dense towards the end of the reduction and the tridiagonalization might have benefitted by an earlier switch to BANDR. This suggests that although the transition regulating criterion works well for most problems, there is room for improvement using lower threshold values or a more sophisticated transition criterion.

Although BANDHYB implements efficient techniques for clipping longer spikes, it is important to reiterate that the primary objective of preordering must be to reduce bandwidth. It is not beneficial to search for preorderings with long spikes or with a wide variation in spike length as the first priority. For example, nested dissection [10] permutations of sparse matrices often produce spikes of widely varying length, with the longest spikes towards the bottom of the matrix. But the bandwidth of such preorderings is much larger than for GPS. Although BANDHYB takes good advantage of the increased sparsity away from the main diagonal for nested dissection preorderings, Table 6 demonstrates that it is much better to choose bandwidth-reducing preorderings. More direct examples of banded-like ordering are given by considering

TABLE 6
Sparse tridiagonalization, GPS *versus nested dissection.*

Name n	GPS bw (transbw)	ND bw (transbw)	Tridiagonalization times (sec) BANDR GPS	BANDR ND	BANDHYB GPS	BANDHYB ND
BCSSTK09 1083	95 (57)	980 (33)	234.3	964.4	150.8	375.6
DWT 1007 1007	34 (12)	894 (22)	72.2	715.3	50.2	137.4
PLAT1919 1919	80 (10)	1891 (72)	706.7	6877.0	424.6	2729.0
BCSSTK27 1224	45 (25)	778 (31)	140.4	1145.7	113.7	264.6

TABLE 7
The effects of a poor preordering on DWT 878.

Preordering method	Bandwidth	BANDR time	BANDHYB time
GPS	27	43.1	34.4
RCM	46	70.0	36.3
GK	40	59.8	34.7

the tridiagonalization of DWT 878 using GPS, RCM [11], and GK [12], [17] preorderings summarized in Table 7. GK and RCM are likely to have longer spikes (higher bandwidth) but better profile than GPS. They are exactly the kinds of orderings one might use to get longer spikes. Switching from a GPS ordering to either RCM or GK, the CPU requirements of BANDR closely mirror the large increase in bandwidth. The "spike clipping" process of BANDHYB, however, is able to efficiently exploit the increased band sparsity presented by RCM and GK, resulting in only marginal increases in CPU requirements.

6. Conclusions and future study. This paper has introduced novel sequential methods for the tridiagonalization of symmetric sparse matrices. We began by describing the difficulties and limitations associated with existing direct methods extended for use with sparse matrices. The most successful of these approaches combined GPS and an enhanced version of the Rutishauser–Schwarz algorithm. Unfortunately, the Rutishauser–Schwarz algorithm was designed primarily for densely banded matrices and proved unable to take advantage of the band sparsity of matrices permuted to reduce bandwidth. Alternatively, working with adjacent transformations and bulge-chasing techniques, the elimination sequence of band entries was modified to produce a hybrid tridiagonalization algorithm, which is shown to more fully exploit matrix sparsity. Compared to the EISPACK BANDR, BANDHYB dramatically reduced the CPU requirements of tridiagonalization without an increase in storage. Even in the worst case, BANDHYB was always comparable to BANDR.

Although substantial progress has been made, there is room for improvement and future research will address many interesting areas. The hybrid algorithm presented by this paper is one approach of many in a large algorithm space. The following list outlines a few of the different alternatives defining the space of band-oriented, sparse tridiagonalization algorithms.

1. Chase all bulge entries between band eliminations or attempt to delay or avoid bulge-chasing transformations.

2. Eliminate a single entry or multiple entries with each transformation.
3. Use adjacent transformations exclusively or permit the use of nonadjacent transformations as well.
4. Target the algorithm for serial, vector, or parallel computing environments.
5. Zero the nonzero entries using alternative elimination sequences.

For densely banded matrices, a recent paper by Bischof and Sun [2] investigates time and space tradeoffs associated with avoiding or delaying bulge chasing and eliminating multiple-band entries with a single transformation. These techniques are not appropriate for all stages of a sparse tridiagonalization. However, although multiple elimination is inappropriate while the band of the matrix remains sparse, future work will consider its incorporation into the second stage of the hybrid tridiagonalization algorithm. In addition, both techniques are of special interest for parallel sparse tridiagonalization algorithms under investigation.

Future research will also explore the potential role of nonadjacent transformations in sparse tridiagonalization algorithms using a symbolic model of fill under development. New elimination sequences that permit additional exploitation of band sparsity will be sought. The transition strategies of BANDHYB will be explored in further detail, seeking more advantageous *threshold* values or more sophisticated transition techniques. Finally, the extension of sparse tridiagonalization techniques to bidiagonalization is being considered.

A future paper will compare eigenvalue methods based on a sparse tridiagonalization with other eigenvalue methods, in particular Lanczos-type algorithms, used in the solution of specific sparse symmetric eigenvalue problems.

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, AND D. SORENSEN, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[2] C. H. BISCHOF AND X. SUN, *A framework for symmetric band reduction and tridiagonalization*, Technical Report MCS-P298-0392, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, July 1992.

[3] E. CUTHILL, *Several strategies for reducing the bandwidth of matrices*, in Sparse Matrices and Their Applications, D. J. Rose and R. A. Willoughby, eds., Plenum Press, New York, 1972, pp. 157–166.

[4] J. DU CROZ, Personal communication, October 1992.

[5] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.

[6] I. S. DUFF, R. GRIMES, J. LEWIS, AND B. POOLE, *Sparse matrix test problems*, ACM SIGNUM News., 17 (1982), p. 22.

[7] I. S. DUFF AND J. K. REID, *On the reduction of sparse matrices to condensed forms by similarity transformations*, J. Instit. Math. Appl., 15 (1975), pp. 217–224.

[8] B. S. GARBOW, J. M. BOYLE, J. J. DONGARRA, AND C. B. MOLER, *Matrix Eigensystem Routines—EISPACK Guide Extension*, Vol. 51, Lecture Notes in Computer Science, Springer-Verlag, New York, Berlin, 1977.

[9] W. M. GENTLEMAN, *Least squares computations by Givens transformations without square roots*, J. Instit. Math. Appl., 12 (1973), pp. 329–336.

[10] A. GEORGE AND J. W. H. LIU, *An automatic nested dissection algorithm for irregular finite element problems*, SIAM J. Numer. Anal., 15 (1978), pp. 1053–1069.

[11] J. A. GEORGE AND J. W. H. LIU, *User guide for SPARSPAK: Waterloo sparse linear equations package*, Research Report CS-78-30, Department of Computer Science, University of Waterloo, Ontario, 1978.

[12] N. E. GIBBS, *A hybrid profile reduction algorithm*, ACM TOMS, 2 (1976), pp. 378–387.

[13] N. E. GIBBS, W. G. POOLE JR., AND P. K. STOCKMEYER, *An algorithm for reducing the bandwidth and profile of a sparse matrix*, SIAM J. Numer. Anal., 13 (1976), pp. 236–250.

[14] N. E. GIBBS, W. G. POOLE JR., AND P. K. STOCKMEYER, *A comparison of several bandwidth and profile reduction algorithms*, ACM TOMS, 2 (1976), pp. 322–330.

[15] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.

[16] L. KAUFMAN, *Banded eigenvalue solvers on vector machines*, ACM TOMS, 10 (1984), pp. 73–86.

[17] J. G. LEWIS, *Implementation of the Gibbs–Poole–Stockmeyer and Gibbs–King algorithms*, ACM TOMS, 8 (1982), pp. 180–189.

[18] H. RUTISHAUSER, *On Jacobi rotation patterns*, in Experimental Arithmetic, High Speed Computing and Mathematics, Vol. 15, Proceedings of Symposia in Applied Mathematics, AMS, Providence, RI, April 1963, pp. 219–239.

[19] H. R. SCHWARZ, *Reduction of a symmetric bandmatrix to triple diagonal form*, Comm. ACM, 6 (1963), pp. 315–316.

[20] ———, *Tridiagonalization of a symmetric band matrix*, in Linear Algebra, J. H. Wilkinson and C. Reinsch, eds., Vol. II, Handbook for Automatic Computation, Springer-Verlag, New York, Berlin, 1971, pp. 273–283.

[21] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.