SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 1-22, January 1990

SUBSTRUCTURING METHODS FOR COMPUTING THE NULLSPACE OF EQUILIBRIUM MATRICES*

R. J. PLEMMONS† AND R. E. WHITE‡

Abstract. Equations of equilibrium arise in numerous areas of engineering. Applications to electrical networks, structures, and fluid flow are elegantly described in *Introduction to Applied Mathematics*, Wellesley Cambridge Press, Wellesley, MA, 1986 by Strang. The context in which equilibrium equations arise may be stated in two forms:

Constrained Minimization Form: $\min(x^T Ax - 2x^T r)$ subject to $Ex = s$,

Lagrange Multiplier Form: $EA^{-1}E^T\lambda = s - EA^{-1}r$ and $Ax = r - E^T\lambda$.

The Lagrange multiplier form given above results from block Gaussian elimination on the 2×2 block matrix system for the constrained minimization form. Here A is generally some symmetric positive-definite matrix associated with the minimization problem. For example, A can be the element flexibility matrix in the structures application. An important approach (called the force method in structural optimization) to the solution to such problems involves dimension reduction nullspace schemes based upon computation of a basis for the nullspace for E. In our approach to solving such problems we emphasize the *parallel computation* of a basis for the nullspace of E and examine the applications to structural optimization and fluid flow. Several new block decomposition and node ordering schemes are suggested and reanalysis computations are investigated. Comparisons of these schemes are made with those of Storaasli et al. for structures and Hall et al. for fluids.

Key words. substructuring, force method, discplacement method, structural optimization, fluid flow, dual variables, reanalysis, parallel algorithms

AMS(MOS) subject classifications. 65F05, 76D05

1. Introduction.

The purpose of this paper is to develop some parallel schemes for computing a basis of the nullspace of an equilibrium matrix with m rows and n columns, having full row rank. Upon aggregation and then scaling, an *equilibrium matrix* (*or incidence matrix*) E can generally be assumed to have entries 0 and ± 1. Such matrices arise in a variety of applications in science and engineering (Strang [20], [21]). Methods of finding a sparse or structured basis of the nullspace of E has been the subject of extensive study over the past few years. Our objective here is to consider parallel algorithms for such computations.

In general, there exists a product of elementary matrices G such that

$$GE = [R_1, R_2] = R_1[I_n, R_1^{-1}R_2] \tag{1}$$

where R_1 is nonsingular. Consequently, the nullspace of GE, and hence E, is generated by the columns of the block matrix

$$B = \begin{bmatrix} R_1^{-1}R_2 \\ -I_{n-m} \end{bmatrix}. \tag{2}$$

We will emphasize the parallel computation of a basis for the column space of B and how it is then used in the solution of problems associated with equilibrium equations.

* Received by the editors November 28, 1988; accepted for publication (in revised form) May 12, 1989.

† Departments of Mathematics and Computer Science, North Carolina State University, Raleigh, North Carolina 27695-8205. This research was supported by U.S. Air Force grants AFOSR-83-0255 and AFOSR-88-0285, and by National Science Foundation grant DMS-85-21154.

‡ Department of Mathematics, North Carolina State University, Raleigh, North Carolina 27695-8205. This research was supported by U.S. Air Force grant AFOSR-83-0255.

An excellent general discussion of equilibrium matrices can be found in Strang [20], where applications to electric networks, structures, and fluid flow are described in detail. The context in which equilibrium matrices arise may be stated in two forms.

Constrained Minimization Problem:

$$\min(x^T Ax - 2x^T r) \quad \text{subject to } Ex = s. \tag{3}$$

Lagrange Multiplier Problem:

$$\begin{bmatrix} A & E^T \\ E & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} r \\ s \end{bmatrix}. \tag{4}$$

All matrices and vectors considered here and elsewhere in this paper are real. The matrix A is generally some symmetric nonnegative definite matrix associated with the minimization problem. For example, A is the element flexibility matrix in the structures application.

In this paper we will examine the applications to structural analysis and fluid flow computations. The structures problem of computing the system forces, displacements, and associated stresses and strains is usually formulated as minimization of potential energy of the elements in the structure, leading to a constrained minimization problem of the form (3). In this case $r = 0$, s is the vector of external loads, x is the system force vector, and $-\lambda$ is the displacement vector associated with (3). Here A is symmetric and block diagonal where each block is associated with an element of the structure and has relatively small dimension (see McGuire and Gallagher [16] and Huston and Passerello [12]).

In the fluid flow problems x is the vector of velocity components, A has block structure, but is not symmetric or block diagonal. Here r and s represent the imposed boundary conditions and λ is the pressure. The equilibrium (or incidence matrix) is a discretization of the conservation of mass equation

$$u_x + v_y = 0$$

where u is the velocity in the x-direction and v is the velocity in the y-direction. As contrasted to the structures case, the fluid flow problem is formulated in terms of the Navier-Stokes equations, and when appropriately discretized, they give the Lagrange multipliers problem (4) (see Hall [11] and Hall, Porsching, and Dougall [10]).

The existence and uniqueness of solutions to problems (3) and (4) are generally given by two sets of assumptions leading to well-known theorems. The first theorem is relevant to the structures problem and the second is important in fluid flow computations. Discussions of the first theorem can be found in Dyn and Ferguson [7] and in Hadley [9].

THEOREM 1.1. *If*

(i) *A is symmetric and nonnegative definite,*

(ii) *E has full row rank, and*

(iii) *A and E have no common null vector,*

then problems (3) *and* (4) *are equivalent and have a unique solution* $\begin{bmatrix} x \\ \lambda \end{bmatrix}$, *where x solves* (3) *and*

$$\lambda = (EE^T)^{-1}E(r - Ax).$$

The second theorem is established in Hall [11] and does not require A to be symmetric. When the Navier-Stokes equations are discretized by the semi-implicit time discretization, by the upwind discretization of the advection terms, and by the finite difference of the viscous terms, then the assumptions of the following theorem are true, as shown by Hall, Porsching, and Dougall [10]. Here problems (3) and (4) are not necessarily equivalent.

THEOREM 1.2. *If*

(i) *A has positive diagonal elements,*

(ii) *A is both row and column diagonally dominant and is strictly diagonally dominant in the rows or columns, and*

(iii) *E has full row rank,*

then the linear system (4) *has a unique solution* $\begin{bmatrix} x \\ \lambda \end{bmatrix}$. *Moreover, if B is any matrix whose columns form a basis of the nullspace of E, then B^TAB is nonsingular.*

There are two methods generally used to calculate the solution of (3) or (4), the displacement method and the force (or dual variable) method.

Displacement Method. Consider (4) and assume A is invertible and E has full row rank. Block elimination in (4) yields the steps:

(i) Solve $EA^{-1}E^T\lambda = EA^{-1}r - s$,

(ii) Solve $Ax = r - E^T\lambda$.

This approach is called the displacement method because for structures λ represents the displacements of the nodes. Here x is the system force vector and is recovered after λ is computed. On the other hand, the force method for structures (dual variable method for fluids) involves calculating x first.

Force Method. Consider (4) and assume that Theorem 2 holds, so that B^TAB is invertible where B is a matrix whose columns form a basis of the nullspace of E.

(i) Solve $Ex_p = s$, where x_p is any particular solution to $Ex = s$.

(ii) Find a basis of the nullspace of E, given by the columns of B, and solve

$$B^TABx_o = B^T(r - Ax_p).$$

(iii) Set $x = x_p + Bx_o$.

(iv) Solve $(EE^T)\lambda = E(r - Ax)$.

The relative merits of the two approaches have been the topic of some debate [11], [14]. Essentially, the force method may be preferable when: (1) B is readily computable, and (2) the row and column dimensions m and n for the equilibrium matrix E are such that $n - m << m$. Then since B^TAB has order $n - m$ while $EA^{-1}E^T$ has order m, the force method is a dimension reduction scheme.

The work in [4], [5], [6], [15], and [18] is a graph theoretic approach to the computation of a sparse nullspace basis. In these papers cycle bases and bipartite graphs are used to form a nullspace basis with as few nonzero components as possible. Although this approach is not used here, we do utilize graph theoretic ideas in what we call proper partitioned structures. In the last section of this paper we examine the nullspace for a simple incidence (or equilibrium) matrix from incompressible fluid flow. The sparseness of the nullspace for the cycle basis approach and our approach are similar. We show that the nullspace calculation (forming B) can often be done by appropriate ordering of the nodes and elements, extending certain results in [2], [8], and [19]. This ordering yields an equilibrium matrix with a great deal of structure which can be exploited by multiprocessing computers in forming B. Furthermore, we will show that in the context of problems (3) and (4), the force method is particularly

useful in the reanalysis of the problem at hand, and in the nonlinear analysis when the components of A depend upon (x, λ).

The outline of the paper is as follows. In §2 we consider a distinguished portion of a structure and show how it generates a given block structure of E. This results in the computation of B as in (2), where B is shown to have a useful block structure. The third section contains the introduction of the concept of a "proper" partition of a structure. This allows us to develop, in some cases, a very nice block structure of E and, accordingly, B. Several examples are presented and the computation of B on multiprocessing computers is discussed. The last two sections deal with the applications of interest here. In §4 an application to the reanalysis of structures is given. The final section contains an application to incompressible fluid flow.

2. Equilibrium matrices with partitioned structure. In this section we examine equilibrium matrices that have a certain block structure which is intimately associated with a partition of a network or undirected graph. In this regard we will use the parlance of finite-element models of physical structures; however, the concept to be developed can be applied to other applications such as electrical networks and fluids (see §5).

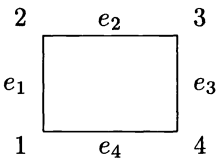
For a given undirected graph G with node set $\mathcal{N}$ and set of edges $\mathcal{E}$, we consider two distinguished, disjoint sets of nodes which we call $\mathcal{N}_{\text{fix}}$ and $\mathcal{N}_{\text{free}}$. Writing $S = (\mathcal{N}, \mathcal{E})$, we call S a *structure* if the graph G is connected. We may think of edges, (i, j), as elements connecting the nodes i and j. A pair $S_1 = (\mathcal{N}_1, \mathcal{E}_1)$ is a *substructure* of S if $\mathcal{N}_1$ and $\mathcal{E}_1$ are nonempty subsets of $\mathcal{N}$ and $\mathcal{E}$, respectively, and S_1 is itself a structure.

DEFINITION 1. Let $S = (\mathcal{N}, \mathcal{E})$ be a structure, where $\mathcal{N}_{\text{free}}$ has cardinality m and $\mathcal{E}$ has cardinality n. An *equilibrium matrix of* S is a $m \times n$ matrix $E = (e_{ij})$, where

$$e_{ij} = \begin{cases} 1 & i \in \mathcal{N}_{\text{free}} \text{ and } j = (i,k) \in \mathcal{E} \text{ for some } k \in \mathcal{N}, \\ 0 & \text{otherwise.} \end{cases}$$

We remark that entries 1 may be replaced by entries -1 in Definition 1 for directed graphs. This situation is illustrated by Example 6 later in §5 on fluids. Also 1 may be replaced by $\pm I$, where I is an $\ell \times \ell$ identity matrix and ℓ is the dimension of an appropriate diagonal block of A. Another possibility is to replace 1 by an $\ell \times \ell$ nonsingular matrix with the sines and cosines of the angles formed by the elements and the coordinate system (see Kaneko, Lawo, and Thierauf [13]). The first situation is illustrated in the examples to follow.

Example 1. (a) $\mathcal{N} = \mathcal{N}_{\text{free}} = \{1, 2, 3, 4\}$ and edge set $\mathcal{E} = \{e_1, e_2, e_3, e_4\}$ where the graph is given by

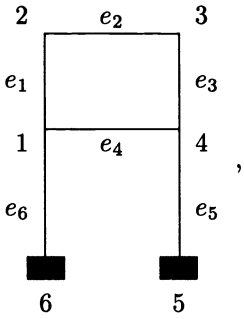


Here the equilibrium matrix is 4×4 and given by

$$E = \begin{bmatrix} 1 & & & 1 \\ 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \end{bmatrix}.$$

Note that $\operatorname{rank}(E) = 3$. In physical terms, this means that the structure is unstable since E does not have full row rank. We may attach two more elements with fixed nodes, resulting in the following example.

(b) $\mathcal{N} = \mathcal{N}_{\text{free}} \cup \mathcal{N}_{\text{fix}}$ where $\mathcal{N}_{\text{free}} = \{1,2,3,4\}$, $\mathcal{N}_{\text{fix}} = \{5,6\}$ and $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6\}$.



$$E = \begin{bmatrix} 1 & & & 1 & & 1 \\ 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & 1 & \end{bmatrix}.$$

If the nodes are rigid, then there are three forces at each node (horizontal, vertical, moment). In this case the 1's represent 3×3 identity matrices and E is in fact 12×18. The equilibrium matrix E in this case has rank 12.

In general, we call a structure with equilibrium matrix E *stable* if E has full row rank. A stable structure always has a basis matrix B for its nullspace which can be expressed in the form (2). However, it is not always clear how to effectively perform the computations in (2).

For notation purposes in what follows, the fixed nodes will either be listed last or deleted from the set of nodes.

Example 2. This is an example of a pin-jointed-truss with 14 nodes and 39 elements. In this case each node has two associated forces and consequently the 1's represented in E are 2×2 identity matrices.

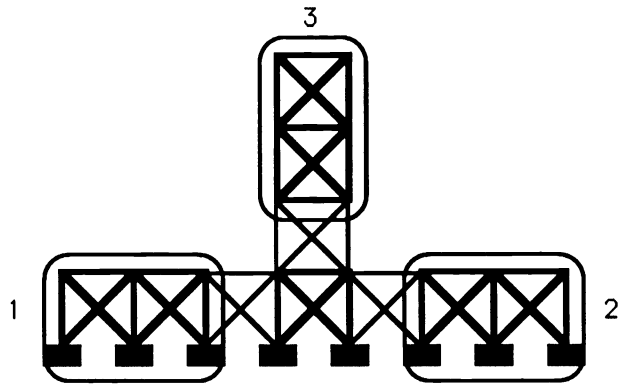


FIG. 1. *Pin-jointed-truss.*

The substructures 1 and 2 in Fig. 1 are stable, whereas 3 is not stable. The connecting elements are indicated by the light lines. If the connecting elements are attached to substructures 1, 2, and 3, then the resulting equilibrium matrix has the block form shown in Fig. 2.

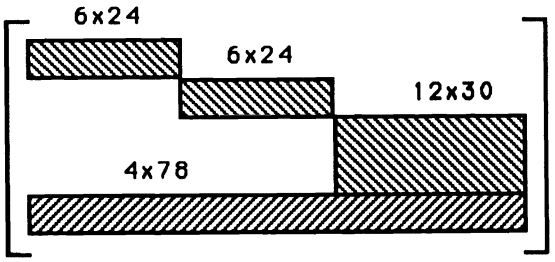


FIG. 2. *Equilibrium matrix: first form.*

However, it turns out that the nullspace basis matrix is easier to compute by our techniques if the connecting elements are associated with the last block of nodes. In this case the equilibrium matrix takes the form in Fig. 3.

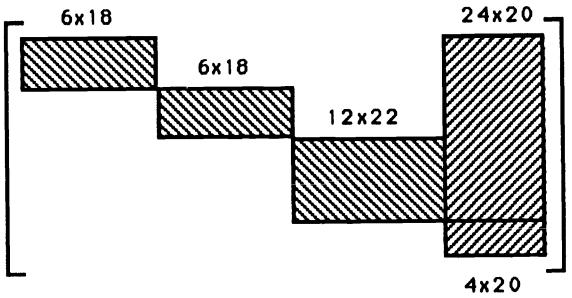


FIG. 3. *Equilibrium matrix: second form.*

The first three blocks for this second form for E are associated with substructures and their nodes, and elements are disjoint from each other. The disjoint substructures correspond to the first three diagonal blocks in the equilibrium matrix. This type of substructuring is common, and therefore, we formalize this in the following definition.

DEFINITION 2. Let $S = (\mathcal{N}, \mathcal{E})$ be a structure and consider the collection of pairs

$$\{(\mathcal{N}_k, \mathcal{E}_k) : \mathcal{N}_k \subseteq \mathcal{N},\ \mathcal{E}_k \subseteq \mathcal{E},\ 1 \leq k \leq K+1\}.$$

The collection is called a *partition* of S if

(i) $\mathcal{N} = \cup_{k=1}^{K+1} \mathcal{N}_k$ is a disjoint union,

(ii) $\mathcal{E} = \cup_{k=1}^{K+1} \mathcal{E}_k$ and the first K sets $\mathcal{E}_k$ are disjoint, and

(iii) $(\mathcal{N}_k, \mathcal{E}_k)$ are substructures for $1 \leq k \leq K$.

We remark that the equilibrium matrix E resulting from a partition of S can be assembled into the general block form shown in Fig. 4.

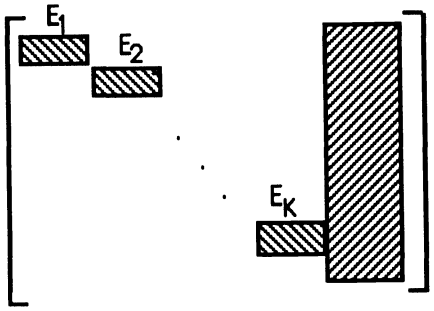


FIG. 4. *Block angular form.*

Here the matrices E_k are the equilibrium matrices associated with the substructures given by $(\mathcal{N}_k, \mathcal{E}_k)$, $1 \leq k \leq K$.

We are now ready to describe an effective algorithm for computing B given by (2), where E has the block form in Fig. 4.

THEOREM 2.1. *Let $S = (\mathcal{N}, \mathcal{E})$ be a stable structure with an associated partition. Then with the equilibrium matrix E assembled into the form in Fig. 4 there is a basis matrix B of the nullspace of E such that for some permutation matrix P, PB has the block form given by Fig. 5.*

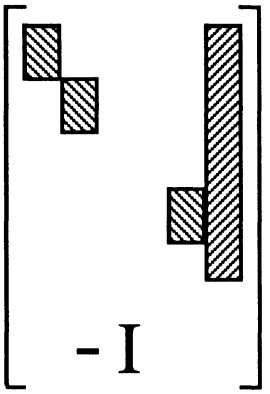


FIG. 5. *Matrix PB.*

Proof. We derive PB in Fig. 5 for $K = 2$ in Definition 2. The proof for $K > 2$ follows in a similar manner. By Definition 2, E can be assembled into the form shown in Fig. 6.

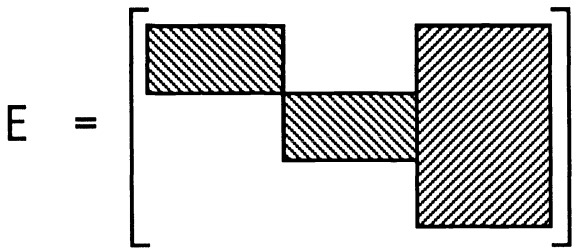


FIG. 6. *Matrix E for $K = 2$.*

Then the use of either elementary row operations or orthogonal transformations will result in a transformation of E into the form shown in Fig. 7.

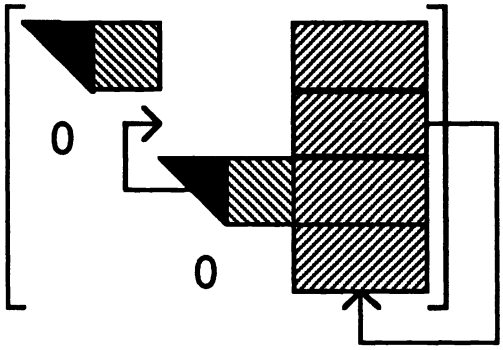


FIG. 7. *Matrix* G_1E.

Now by using row interchanges, as indicated by the arrows in Fig. 7, to move any resulting rows with all zeros in the first two diagonal blocks to the last block, we have the reduced form shown in Fig. 8.

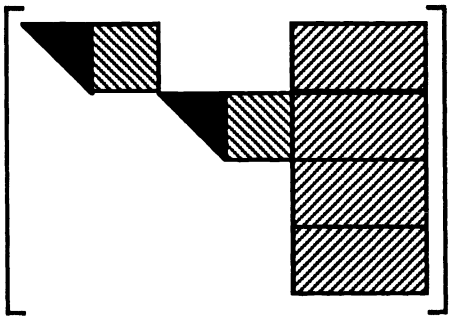


FIG. 8. *Matrix* P_1G_1E.

By further reduction on the bottom block there results a further form as shown in Fig. 9.

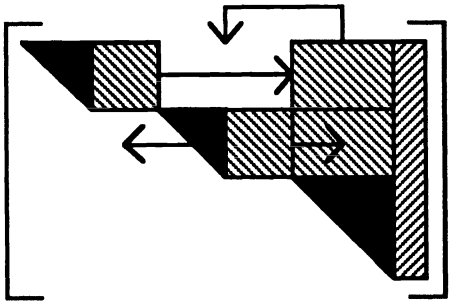


FIG. 9. *Matrix* $G_2P_1G_1E$.

Since E is of full row rank, the triangular matrices shaded in black must be invertible.

Now, moving the column blocks between the triangular blocks to last column, as indicated by the arrows in Fig. 9, results in the basic reduced form in Fig. 10. In Fig. 10 R_1 is the left square region and $[R_1, R_2] = G_2P_1G_1EP_2$.

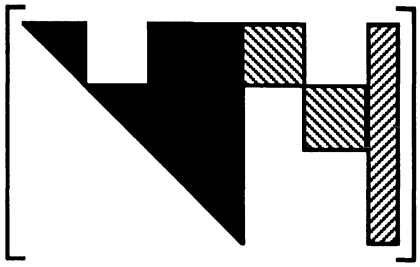


FIG. 10. *Matrix* $G_2P_1G_1EP_2$.

By using either block back substitution or block elementary matrices we obtain $R_1^{-1}R_2$, and note that it has the same form as R_2. Thus, the nullspace basis matrix of

$$EP_2 = (G_2P_1G_1)^{-1}[R_1, R_2] = (G_2P_1G_1)^{-1}R_1[I, R_1^{-1}R_2]$$

is given by

$$B = P_2^T \begin{bmatrix} R_1^{-1}R_2 \\ -I \end{bmatrix}.$$

Thus P_2B has the block form in Fig. 5, completing the proof of the theorem for the case $K = 2$. □

In order to illustrate this form, consider the matrix E from Example 2 given by Fig. 3. Each 6×18 block has full row rank 6, the rank of the 12×22 block is 10, and the 4×20 block has rank 4. Thus the form of $[R_1, R_2]$ is given in Fig. 11.

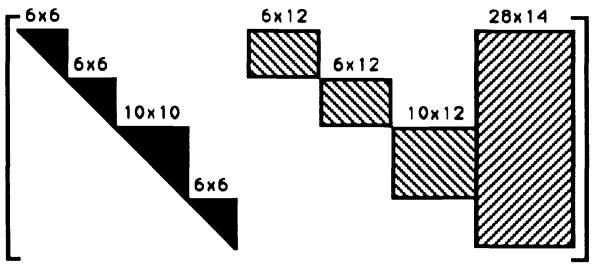


FIG. 11. *Matrix* $[R_1, R_2]$ *in Example* 2.

Because of the block structure of E associated with the finite-element model of the physical structure, certain steps in the calculation of R_1, R_2, and B can be done concurrently. This is summarized as follows.

Parallel Computation of B for a Stable Structure with Partition.

Step 1 In parallel, reduce each diagonal block of E to upper triangular form, where by row permutations the zero rows are last.

Step 2 Reorder the rows so that the form of P_1G_1E in Fig. 8 is obtained for the general case.

Step 3 Reduce the last row block to obtain $G_2P_1G_1E$ of the general form given in Fig. 9.

Step 4 Reorder the columns to obtain the general form of $G_2P_1G_1EP_2 = [R_1, R_2]$ in Fig. 10.

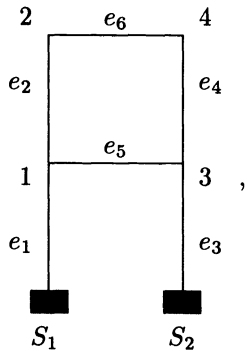
Step 5 In parallel, compute $R_1^{-1}R_2$ for the $K+1$ blocks of rows.

Step 6 Form

$$B = P_2^T \begin{bmatrix} R_1^{-1}R_2 \\ -I \end{bmatrix}.$$

In the next section we show that in some cases a proper reordering of the nodes and elements will allow us to avoid Steps 1-3.

3. Equilibrium matrices and proper partitioned structures. As motivation for the definition for a proper partition we reconsider Example 1 with the following ordering of nodes and elements:



$$E = \begin{bmatrix} I & I & & & I & \\ & I & & & & I \\ & & I & I & I & \\ & & & I & & I \end{bmatrix}.$$

Notice the form of the equilibrium matrix E. The diagonal blocks have inverses and correspond to stable substructures given by

$$\begin{aligned} S_1 &= (\mathcal{N}_1, \mathcal{E}_1) = (\{1,2\},\ \{e_1, e_2\}), \\ S_2 &= (\mathcal{N}_2, \mathcal{E}_2) = (\{3,4\},\ \{e_3, e_4\}). \end{aligned}$$

The remaining elements, e_5 and e_6, connect these stable substructures.

DEFINITION 3. Let $\{S_k = (\mathcal{N}_k, \mathcal{E}_k);\ k = 1, \cdots, K+1\}$ be a partition of $S = (\mathcal{N}, \mathcal{E})$. A partition is called *proper* if

(i) $\mathcal{N}_{K+1}$ is empty.

(ii) $\mathcal{N}_k$ and $\mathcal{E}_k$ have the same cardinalities for $1 \le k \le K$.

(iii) S_k are stable for $k = 1, \cdots, K$; that is, each block E_k of E has an inverse.

We remark that the equilibrium matrix E for a structure with a proper partition must have the form given in Fig. 12.

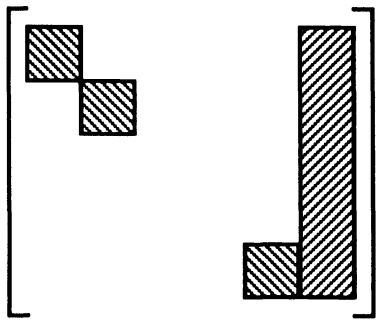


FIG. 12. *Matrix* $E = [R_1, R_2]$.

Here we define $R_1 \equiv \text{diag}(E_1, E_2, \cdots, E_K)$. The elements of E_{K+1} are often called the redundant elements of the structure. Since the diagonal blocks of R_1 are square and nonsingular, the structure must be stable. The following examples will illustrate that R_2 often has a great deal of structure. In each example we indicate the parallel portions of the computation of

$$B = \begin{bmatrix} R_1^{-1} R_2 \\ -I \end{bmatrix}.$$

Example 3. Consider Example 2 with the eight disjoint stable substructures given by the dark lines (Fig. 13). The connecting elements E_9 are indicated by light lines.

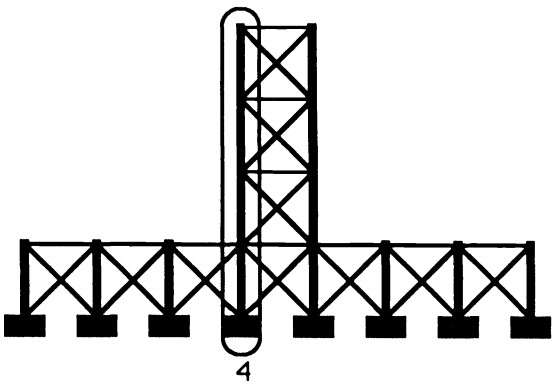


FIG. 13. *Example* 2 *with eight disjoint stable substructures.*

Since this is pin-jointed truss, each 1 in the equilibrium matrix is a 2×2 identity matrix and E is 28×88.

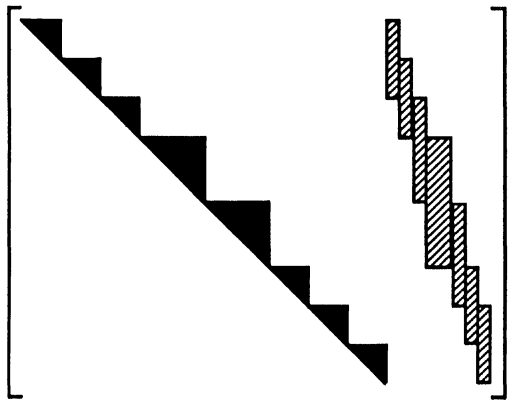


FIG. 14. *Equilibrium matrix form for Fig.* 13.

The diagonal blocks in Fig. 14 are

$$E_k = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} = I \quad \text{for} \quad k = 1,2,3,6,7,8 \quad \text{and}$$

$$E_4 = E_5 = \begin{bmatrix} I & I & & \\ & I & I & \\ & & I & I \\ & & & I \end{bmatrix}, \quad I = 2 \times 2 \text{ identity.}$$

Note the "diagonal" structure of R_2 which is a result of ordering the connecting elements from the left to the right. The parallel computation of $R_1^{-1}R_2$ is clear. In this case the work to be done on each substructure is not equal. So, we might want to consider three groups of substructures $\{S_1, S_2, S_3\}$, $\{S_4, S_5\}$, and $\{S_6, S_7, S_8\}$.

Example 4. Consider a rigid frame which models a wheel with eight spokes (Fig. 15). Each spoke is a stable substructure and together they form a proper partition. The connecting elements are indicated by the light lines $e_{33}, \cdots, e_{40}$.

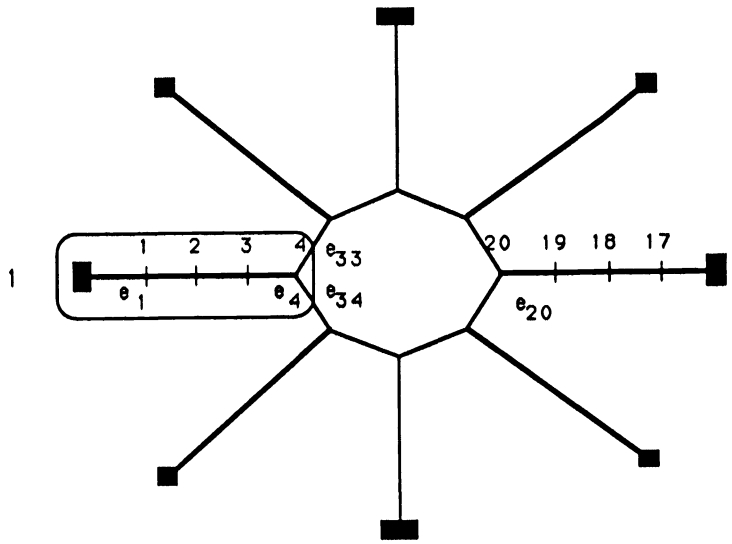


FIG. 15. *Rigid frame that models a wheel.*

There are 32 nodes with three forces at each node. Thus, E will be a 96×120 matrix of the form

$$E = \begin{bmatrix} E_1 & & & & & & & f & f & & & \\ & E_2 & & & & & & & f & f & & \\ & & E_3 & & & & & & & f & f & \\ & & & E_4 & & & & & & & f & f \\ & & & & E_5 & & & & & & f & f \\ & & & & & E_6 & & & & & f & f \\ & & & & & & E_7 & & & & f & f \\ & & & & & & & E_8 & f & & & f \end{bmatrix}$$

where

$$f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ I \end{bmatrix}, \quad E_k = \begin{bmatrix} I & I & & \\ & I & I & \\ & & I & I \\ & & & I \end{bmatrix}, \quad E_k^{-1} = \begin{bmatrix} I & -I & I & -I \\ & I & -I & I \\ & & I & -I \\ & & & I \end{bmatrix} \quad \text{for } k = 1, \cdots, 8.$$

Then $R_1^{-1}R_2$ is easily computed and B is the 120×24 matrix given by

$$B = \begin{bmatrix} R_1^{-1}R_2 \\ -I \end{bmatrix} = \begin{bmatrix} g & g & & & & & & \\ & g & g & & & & & \\ & & g & g & & & & \\ & & & g & g & & & \\ & & & & g & g & & \\ & & & & & g & g & \\ & & & & & & g & g \\ g & & & & & & & g \\ -I & & & & & & & \\ & -I & & & & & & \\ & & -I & & & & & \\ & & & -I & & & & \\ & & & & -I & & & \\ & & & & & -I & & \\ & & & & & & -I & \\ & & & & & & & -I \end{bmatrix}.$$

g is defined by $E_k^{-1}f$. The above calculations can be done concurrently.

Example 5. This example of a pin-jointed structure illustrates how a nested

structure (Fig. 16) of the elements will yield a very nice structure of R_2, and consequently, B. This gives a more complicated form of R_2 in $E = [R_1, R_2]$, where a proper partition is identified.

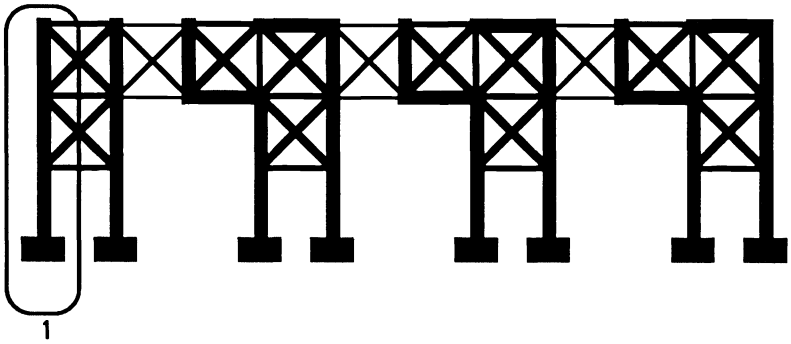


FIG. 16. *Pin-jointed-truss with nested structure of the elements.*

The eight stable substructures are given by the dark lines. The dark and regular lines indicate four disjoint substructures. The light lines are elements which connect these four substructures. There are 30 nodes and 79 elements and two forces at each node. Therefore, the equilibrium matrix E is 60×158 and has the form shown in Fig. 17.

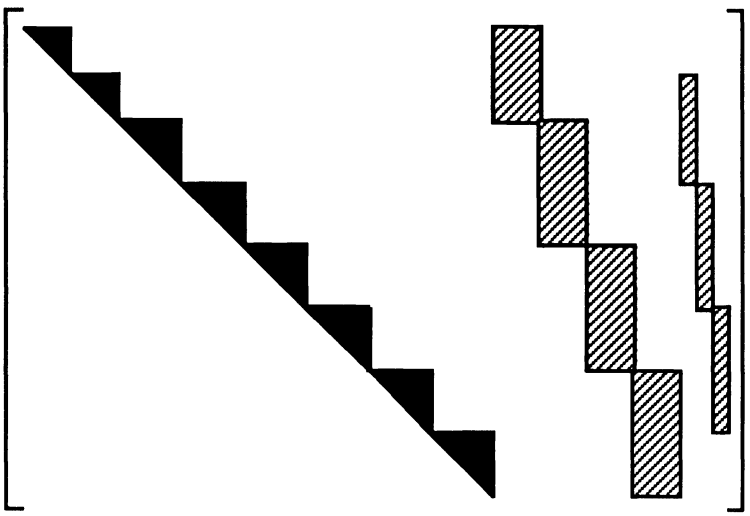


FIG. 17. *Equilibrium matrix form for Fig.* 16.

The blocks of R_2 can be grouped to match those in R_1. This allows us to implement the computation of $R_1^{-1}R_2$ on a multiprocessing computer.

4. Reanalysis of structures and the force method. In this section we apply the force method and the above structure of the nullspace matrix to the reanalysis of a structure. Reanalysis refers to the analysis of a structure which has been only slightly

modified. We will assume that only one element has been modified and that the equilibrium matrix remains unchanged. We assume the equilibrium matrix has full row rank. Thus, in (4) $r = 0$, $A = \operatorname{diag}(A_k)$, where each A_k is an $n_k \times n_k$ symmetric positive-definite matrix. A will be modified by changing one A_k to $A_k + \delta_k \delta_k^T$ where the δ_k are $n_k \times n_k$; we may assume the perturbation of A_k may be written in the form $\delta_k \delta_k^T$ (see Batt and Gellin [1]).

Since A is symmetric positive definite, A has a Cholesky factorization $A = G^T G$. Step (ii) of the force method may be viewed as the normal equation of a least squares problem

$$B^T G^T G B x_o = -B^T G^T G x_p. \tag{5}$$

Thus, let $GB = QR$ be the QR factorization and find x_o by solving

$$R x_o = -Q^T G x_p. \tag{6}$$

Here, R^{-1} exists because B has full column rank.

The advantage of the force method is that we can use the QR factorization of the unperturbed problem (5) to solve the perturbed problem

$$B^T \left(A + e_k \delta_k \delta_k^T e_k^T\right) B(x_o + \Delta x_k) = -B^T \left(A + e_k \delta_k \delta_k^T e_k^T\right) x_p. \tag{7}$$

Here only the kth block of A has been perturbed by $\delta_k \delta_k^T$. We have used the notation e_k for a $n \times n_k$ matrix

$$e_k = \begin{bmatrix} 0 \\ \vdots \\ I \\ \vdots \\ 0 \end{bmatrix} \leftarrow k\text{th block, } I = n_k \times n_k \text{ identity and}$$

A is $n \times n$,
E is $m \times n$,
B is $n \times (n-m)$,

$B^T e_k$ is $(n-m) \times n_k$ matrix consisting of the kth block column of B^T.

The key formula in the proof of the following theorem is the Sherman-Morrison-Woodbury formula (see Ortega and Rheinboldt [17]):

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1}. \tag{8}$$

Here $(A + UV^T)^{-1}$ exists if and only if $(I + V^T A^{-1} U)^{-1}$ exists.

THEOREM 4.1. *Let x_o be the solution of* (5) (*or equivalently* (6)). *Let $GB = QR$ where $A = G^T G$ is symmetric positive definite and B has full column rank. Then the solution of* (7) *is given by $x_o + \Delta x_k$, where*

$$\Delta x_k = \left[R^{-1}R^{-T} - R^{-1}R^{-T}U_k(I + U_k^T R^{-1} R^{-T} U_k)^{-1} U_k^T R^{-1} R^{-T}\right] f_k, \tag{9}$$

$$\begin{aligned} U_k &= B^T e_k \delta_k \qquad (n-m) \times n_k \text{ matrix}, \\ I &= n_k \times n_k \quad \text{identity matrix}, \\ f_k &= -B^T e_k \delta_k \delta_k^T e_k^T (Bx_o + x_p). \end{aligned}$$

Proof. Consider (7) where x_o satisfies (5)

$$B^T(A + e_k\delta_k\delta_k^T e_k^T)B(x_o + \Delta x_k) = -B^T(A + e_k\delta_k\delta_k^T e_k^T)x_p,$$

$$B^T ABx_o + B^T AB\Delta x_k + B^T e_k\delta_k\delta_k^T e_k^T B(x_o + \Delta x_k) = -B^T Ax_p - B^T e_k\delta_k\delta_k^T e_k^T x_p.$$

Since (5) holds and $A = G^T G$,

$$(B^T G^T GB + B^T e_k\delta_k\delta_k^T e_k^T B)\Delta x_k = -B^T e_k\delta_k\delta_k^T e_k^T(Bx_o + x_p) \equiv f_k.$$

Apply the Sherman–Morrison–Woodbury formula (8) with the following substitutions

$$\begin{aligned} A &= B^T G^T GB = R^T Q^T QR = R^T R, \\ A^{-1} &= R^{-1}R^{-T}, \\ U &= U_k = B^T e_k \delta_k, \\ V &= U. \end{aligned}$$

This completes the proof of the theorem. □

Note, $I + U_k^T R^{-1} R^{-T} U_k$ in (9) is an $n_k \times n_k$ symmetric positive-definite matrix, and therefore, its inverse exists and is simple to compute for small n_k.

The appearance of the e_k and e_k^T matrices in (9) reduces the amount of computation. The n_k usually ranges from one to ten. Also, B is usually sparse. The calculation of the QR factorization of GB often can be done in parallel. Consider Example 4 where B has a block structure. The Givens transformations can be used in parallel by concurrently working on the eight column blocks. The $-I$ in row block 33 and column block 1 can be used to annihilate the components in row blocks 1-4 and column block 1. At the same time the terms in row blocks 5-8 and column 2 can be annihilated by the $-I$ in row block 34 and column block 2. The remainder of the top 32 row blocks can be annihilated concurrently in a similar manner.

5. Applications to incompressible fluid flow. In this section we consider an application of the force method, the proper partition of the finite difference grid (structure), and the time induced reanalysis to incompressible fluid flow (see [3] or [11]). As noted in Hall [11], an appropriate discretization of the Navier-Stokes equations will yield a sequence of problems of the form (4). The matrix A will change a little from one time step to the next. As E reflects the conservation of mass equation, it remains fixed. In this section we illustrate how we can approximate the solution of

$$B^T(A + \Delta A)B(x_o + \Delta x) = B^T(r - (A + \Delta A)x_p) \tag{10}$$

where ΔA represents a change in A because the velocity has changed from one time step to the next time step(s). We want to make use of the solution process when $\Delta A = 0$ and $\Delta x = 0$, that is, the LU factorization of $B^T AB$ is known. The perturbation ΔA

of A, unlike the perturbation for structures, may change every row of A. However, the magnitude of ΔA, $\|\Delta A\|$, may be small for suitably small changes in time. Also, A is not symmetric, but $B^T AB$ is invertible (see Theorem 1.2).

Before we consider the details, we review an example given in Burkardt, Hall, and Porsching [4] and Hall [11]. Consider the incompressible fluid flow about an obstacle with no-slip boundary conditions at the walls (Fig. 18).

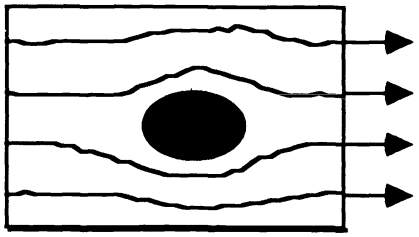


FIG. 18. *Fluid flow about an obstacle.*

A finite difference grid with 14 cells has 21 unknown velocity components given by the following vectors in Fig. 19.

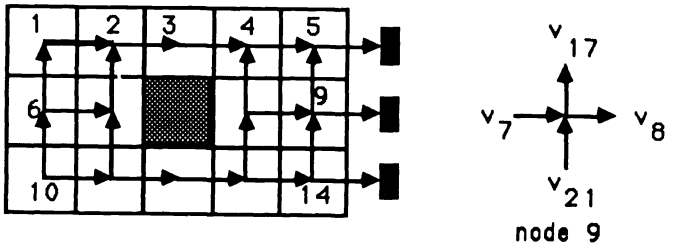


FIG. 19. *Finite difference grid for Fig.* 18.

(A full numbering of this network is given in Hall [11].) Each cell is analogous to a free node and each vector component is analogous to an element. The nodes to the right of vectors v_5, v_8, and v_{13} are fixed nodes. The connected graph is now directed and the corresponding equilibrium matrix has entries 0 and ± 1, called an incidence matrix in [11]. For example, row 9 in the incidence matrix reflects the conservation of mass for cell 9:

$$\frac{v_8 - v_7}{h} + \frac{v_{17} - v_{21}}{h} = 0,$$

where $h = \Delta x = \Delta y$. Except for cells 6 and 7, each horizontal row of cells is similar to a stable substructure. The incidence matrix is almost in the form given by a proper partition. It fails to be a proper partition because the cells 6 and 7 with the vector v_6 is not a stable structure. The incidence matrix E is given:

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

Execute the following elementary row operations:

add row 6 to row 7,
interchange row 7 and row 14,
add row 14 to row 1,
subtract row 14 from row 6 .

Representing these by a matrix G we have the following matrix GE:

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

The matrix GE has the form which is associated with a proper partition. Consequently, the nullspace of E is given by the span of the columns of B which is a 21×7 matrix:

$$B = \begin{bmatrix} R_1^{-1}R_2 \\ -I \end{bmatrix} = \begin{bmatrix}
1 & & & -1 & -1 & & \\
0 & & & -1 & -1 & & \\
0 & & & -1 & -1 & & \\
0 & -1 & & -1 & -1 & & \\
0 & -1 & -1 & -1 & -1 & & \\
-1 & & & 0 & 1 & 0 & 0 \\
0 & 1 & & 0 & 0 & -1 & 0 \\
0 & 1 & 1 & 0 & 0 & -1 & -1 \\
 & & & 1 & 0 & & \\
 & & & 1 & 1 & & \\
 & & & 1 & 1 & & \\
 & & & 1 & 1 & 1 & \\
 & & & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & -1 & -1 & 0 & 0 \\
-1 & & & & & & \\
 & -1 & & & & & \\
 & & -1 & & & & \\
 & & & -1 & & & \\
 & & & & -1 & & \\
 & & & & & -1 & \\
 & & & & & & -1
\end{bmatrix}.$$

It is interesting to compare this nullspace basis with the basis $\bar{B}$ obtained in Hall [11] by using graph theoretic ideas. In [11], $\bar{B}$ is a little more sparse, but $\bar{B}$ may not be constructed by utilizing the properties of a proper partition.

In the above, the incidence matrix was initially not in the form associated with a proper partition of a structure. We can reorder the nodes so that the corresponding incidence matrix does have the desired form. This is indicated by the following example where the three substructures are given by the dark lines.

Example 6. Consider the following proper partition of a finite difference grid (structure) (Fig. 20).

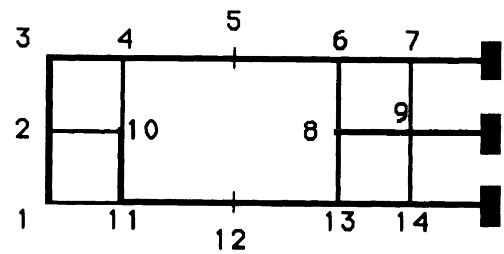


FIG. 20. *Proper partition of a finite difference grid.*

Then the associated equilibrium matrix E has the following form:

$$
\begin{bmatrix}
1 & & & & & & & & & & & & & & 1 & & & & & & \\
-1 & 1 & & & & & & & & & & & & & & 1 & & & & & \\
 & -1 & 1 & & & & & & & & & & & & & & & & & & \\
 & & -1 & 1 & & & & & & & & & & & & & -1 & & & & \\
 & & & -1 & 1 & & & & & & & & & & & & & & & & \\
 & & & & -1 & 1 & & & & & & & & & & & & -1 & & & \\
 & & & & & -1 & 1 & & & & & & & & & & & & -1 & & \\
 & & & & & & & 1 & & & & & & & & & & 1 & & -1 & \\
 & & & & & & & -1 & 1 & & & & & & & & & & 1 & & -1 \\
 & & & & & & & & & 1 & & & & & & -1 & 1 & & & & \\
 & & & & & & & & & -1 & 1 & & & & -1 & & & & & & \\
 & & & & & & & & & & -1 & 1 & & & & & & & & & \\
 & & & & & & & & & & & -1 & 1 & & & & & & & 1 & \\
 & & & & & & & & & & & & -1 & 1 & & & & & & & 1
\end{bmatrix}.
$$

The resulting $R_1^{-1}R_2$ has the form

$$
R_1^{-1}R_2 = \begin{bmatrix}
1 & & & & & & \\
1 & 1 & & & & & \\
1 & 1 & & & & & \\
1 & 1 & -1 & & & & \\
1 & 1 & -1 & & & & \\
1 & 1 & -1 & -1 & & & \\
1 & 1 & -1 & -1 & -1 & & \\
 & & & 1 & & -1 & \\
 & & & 1 & 1 & -1 & -1 \\
 & -1 & 1 & & & & \\
-1 & -1 & 1 & & & & \\
-1 & -1 & 1 & & & & \\
-1 & -1 & 1 & & & 1 & \\
-1 & -1 & 1 & & & 1 & 1
\end{bmatrix}.
$$

Then B has the form

$$\begin{bmatrix} R_1^{-1}R_2 \\ -I \end{bmatrix} = \begin{bmatrix} 1 & & & & & & \\ 1 & 1 & & & & & \\ 1 & 1 & & & & & \\ 1 & 1 & -1 & & & & \\ 1 & 1 & -1 & & & & \\ 1 & 1 & -1 & -1 & & & \\ 1 & 1 & -1 & -1 & -1 & & \\ & & & 1 & & -1 & \\ & & & 1 & 1 & -1 & -1 \\ & -1 & 1 & & & & \\ -1 & -1 & 1 & & & & \\ -1 & -1 & 1 & & & & \\ -1 & -1 & 1 & & & 1 & \\ -1 & -1 & 1 & & & 1 & 1 \\ -1 & & & & & & \\ & -1 & & & & & \\ & & -1 & & & & \\ & & & -1 & & & \\ & & & & -1 & & \\ & & & & & -1 & \\ & & & & & & -1 \end{bmatrix}.$$

Consider the perturbed problem (10), and assume the solution for (10) with $\Delta A = 0$ is given by x_o. Theorem 2 gives conditions on A and B that are applicable to fluid flow problems. Suppose that $B^TAB = LU$ has a known LU factorization. Then the solution of equation (10) can be approximated for suitably small ΔA.

THEOREM 5.1. *Let* $(B^TAB)^{-1}$ *exist and* x_o *satisfy* (10) *with* $\Delta A = 0$. *Then for suitably small* ΔA,

$$(\Delta x)^{m+1} = (B^TAB)^{-1}\left[-B^T\Delta AB(\Delta x)^m - B^T\Delta A(Bx_o + x_p)\right] \tag{11}$$

converges to Δx *and* $x_o + \Delta x$ *satisfies* (10).

Proof. Write (10) in expanded form:

$$B^TABx_o + B^TAB\Delta x + B^T\Delta ABx_o + B^T\Delta AB\Delta x = B^T(r - Ax_p) - B^T\Delta Ax_p.$$

Since $B^TABx_o = B^T(r - Ax_p)$,

$$B^TAB\Delta x + B^T\Delta AB\Delta x = -B^T\Delta A(Bx_o + x_p).$$

Or, as $(B^TAB)^{-1}$ exists,

$$\Delta x = (B^TAB)^{-1}(-B^T\Delta AB)\Delta x - (B^TAB)^{-1}\left(B^T\Delta A(Bx_o + x_p)\right).$$

So, if $\rho\left((B^TAB)^{-1}(-B^T\Delta AB)\right) < 1$, then the iterative scheme (11) must converge. Since

$$\|(B^TAB)^{-1}(-B^T\Delta AB)\| \leq \|(B^TAB)^{-1}\|\,\|B^T\Delta AB\|,$$

we have the desired result for suitably small ΔA. □

As already mentioned, we must solve a sequence of problems of the form (4) where A is changing with each time step. It is not necessary to compute the LU factorization for each B^TAB. In particular, we may use the following scheme for solving (4) with A replaced by A_ℓ = the value of A at the ℓth time step:

(i) Factor $B^T A_\ell B = L_\ell U_\ell$ and solve (4) for $x_o = x_\ell =$ the value of x at the ℓth time step.

(ii) Approximate Δx_k in $B^T A_{\ell+k} B(x_\ell + \Delta x_k) = B^T(r - A_{\ell+k} x_p)$ by using line (11) in Theorem 5 with $\Delta A = A_{\ell+k} - A_\ell$ and $1 \leq k \leq K$.

(iii) Repeat (i) and (ii) with ℓ replaced by $\ell + K$. The size of K will be determined by ΔA and the magnitude of $\rho\left((B^T A B)^{-1}(B^T \Delta A B)\right)$ in the iterative scheme (11).

There are several advantages of using the force method to solve the full system in (4). The force method is a variable reduction scheme which involves solution of reduced linear systems with $B^T A_\ell B$. Theorem 5 shows that $B^T A_\ell B$ does not require a new LU factorization for each ℓ. The matrix B is a nullspace basis matrix for an incidence matrix E; where, upon appropriate ordering of the cells, E has the features of a proper partitioned equilibrium matrix. Consequently, B is easily computed and can have a useful structure of its own which can further simplify the solution of linear systems with $B^T A_\ell B$. □

REFERENCES

[1] J.R. BATT AND S. GELLIN, *Rapid reanalysis by the force method*, Comput. Methods Appl. Mech. Engrg., 53 (1985), pp. 105-117.

[2] M.W. BERRY, M.T. HEATH, I. KANEKO, M. LAWO, R.J. PLEMMONS, AND R.C. WARD, *An algorithm to compute a sparse basis of the nullspace*, Numer. Math., 47 (1985), pp. 483-504.

[3] U. BULGARELLI, G. GRAZIANI, D. MANSUTTI, AND R. PIVA, *A reduced scheme via discrete stream function for unsteady Navier-Stokes equations in general curvilinear coordinates*, presented at VI GAMM Conference, Gottingen, Germany, Lecture Notes on Numerical Fluid Mechanics, Vol. 13, Springer-Verlag, Berlin, 1985.

[4] J. BURKARDT, C. HALL, AND T. PORSCHING, *The dual variable method for the solution of compressible fluid flow problems*, SIAM J. Algebraic Discrete Methods, 7(1986), pp. 476-483.

[5] T.F. COLEMAN AND A. POTHEN, *The nullspace 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. 544-561.

[7] N. DYN AND W.E. FERGUSON, JR., *The numerical solution of equality-constrained quadratic programming problems*, Math. Comp., 41(1983), pp. 165-170.

[8] J.R. GILBERT AND M.T. HEATH, *Computing a sparse basis for the nullspace*, SIAM J. Algebraic Discrete Methods, 8(1987) pp. 446-459.

[9] G. HADLEY, *Nonlinear and Dynamic Programming*, Addison-Wesley, Reading, MA, 1964.

[10] C.A. HALL, T.A. PORSCHING, AND R.S. DOUGALL, *Numerical methods for thermally expandable two-phase-flow computational techniques for steam generator modeling*, Report NP-1416, Electric Power Research Institute, Palo Alto, CA, May 1980.

[11] C.A. HALL, *Numerical solution of Navier-Stokes problems by the dual variable method*, SIAM J. Algebraic Discrete Methods, 6(1985), pp. 220-236.

[12] R.L. HUSTON AND C.E. PASSERELLO, Finite Element Methods, Marcel Dekker, New York, Basel, 1984.

[13] I. KANEKO, M. LAWO, AND G. THIERAUF, *On computational procedures for the force method*, Internat. J. Numer. Methods Engrg., 18(1982), pp. 1469-1495.

[14] I. KANEKO AND R.J. PLEMMONS, *Minimum norm solutions to linear elastic analysis problems*, Internat. J. Numer. Methods Engrg., 20(1984), pp. 983-998.

[15] A. KAVEH, *A combinational optimization problem; optimal generalized cycle bases*, Comput. Methods Appl. Mech. Engrg., 20(1979), pp. 39-51.

[16] W. MCGUIRE AND R.H. GALLAGHER, *Matrix Structural Analysis*, John Wiley, New York, 1979.

[17] J.M. ORTEGA AND W.C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[18] A. POTHEN, *Sparse null basis computations in structural optimization*, Tech. Report CS-88-27, Department of Computer Science, Penn. State University, University Park, PA, July 1988.

[19] O. STORAASLI AND P. BERGAN, *A nonlinear substructuring method for concurrent processing computers*, presented at AIAA/ASME/ASCE/AAS 27th Structures, Structural Dynamics, and Materials Conference, San Antonio, TX, May 1986.

[20] G. STRANG, *A framework for equilibrium equations*, SIAM Rev., 30(1988), pp. 283-297.

[21] , *Introduction to Applied Mathematics*, Wellesley Cambridge Press, Wellesley, MA, 1986.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 23–41, January 1990

STABILITY ANALYSIS OF ALGORITHMS FOR SOLVING CONFLUENT VANDERMONDE-LIKE SYSTEMS*

NICHOLAS J. HIGHAM†

Abstract. A confluent Vandermonde-like matrix $P(\alpha_0, \alpha_1, \cdots, \alpha_n)$ is a generalisation of the confluent Vandermonde matrix in which the monomials are replaced by arbitrary polynomials. For the case where the polynomials satisfy a three-term recurrence relation algorithms for solving the systems $Px = b$ and $P^Ta = f$ in $O(n^2)$ operations are derived. Forward and backward error analyses that provide bounds for the relative error and the residual of the computed solution are given. The bounds reveal a rich variety of problem-dependent phenomena, including both good and bad stability properties and the possibility of extremely accurate solutions. To combat potential instability, a method is derived for computing a "stable" ordering of the points α_i; it mimics the interchanges performed by Gaussian elimination with partial pivoting, using only $O(n^2)$ operations. The results of extensive numerical tests are summarised, and recommendations are given for how to use the fast algorithms to solve Vandermonde-like systems in a stable manner.

Key words. Vandermonde matrix, orthogonal polynomials, Hermite interpolation, Clenshaw recurrence, forward error analysis, backward error analysis, stability, iterative refinement

AMS(MOS) subject classifications. primary 65F05, 65G05

C. R. classification. G.1.3

1. Introduction.

Let $\{p_k(t)\}_{k=0}^n$ be a set of polynomials, where p_k is of degree k, and let $\alpha_0, \alpha_1, \cdots, \alpha_n$ be real scalars, ordered so that equal points are contiguous, that is,

$$\alpha_i = \alpha_j \quad (i<j) \quad \Rightarrow \quad \alpha_i = \alpha_{i+1} = \cdots = \alpha_j. \tag{1.1}$$

We define the *confluent Vandermonde-like matrix*

$$P = P(\alpha_0, \alpha_1, \cdots, \alpha_n) = [q_0(\alpha_0), q_1(\alpha_1), \cdots, q_n(\alpha_n)] \in \mathbf{R}^{(n+1)\times(n+1)},$$

where the vectors $q_j(t)$ are defined recursively by

$$q_j(t) = \begin{cases} [p_0(t), p_1(t), \cdots, p_n(t)]^T & \text{if } j = 0 \text{ or } \alpha_j \neq \alpha_{j-1}, \\ \dfrac{d}{dt} q_{j-1}(t), & \text{otherwise.} \end{cases}$$

In the case of the monomials, $p_k(t) = t^k$, this definition yields the well-known confluent Vandermonde matrix [9], [4]. When the points α_i are distinct we can write $P = (p_i(\alpha_j))_{i,j=0}^n$, and P is referred to as a nonconfluent Vandermonde-like matrix [12]. For all polynomials and points, P is nonsingular; this follows from the derivation of the algorithms in § 2.

Various applications give rise to confluent or nonconfluent Vandermonde or Vandermonde-like systems

$$Px = b \quad \text{(primal)} \tag{1.2}$$

and

$$P^Ta = f \quad \text{(dual)}. \tag{1.3}$$

* Received by the editors December 7, 1987; accepted for publication (in revised form) March 22, 1989.
† Department of Mathematics, University of Manchester, Manchester M13 9PL, United Kingdom (NA.NHIGHAM@NA-NET.STANFORD.EDU).

Three examples are the construction of quadrature formulae [2], [14], [15], rational Chebyshev approximation [1], and the approximation of linear functionals [3], [22].

For the monomials, with distinct points α_i, efficient algorithms for solving the primal and dual Vandermonde systems are given in [5]. These algorithms have been generalised in two ways: in [4] they are extended to confluent Vandermonde matrices, and in [12] they are extended to nonconfluent Vandermonde-like matrices, under the assumption that the polynomials $p_k(t)$ satisfy a three-term recurrence relation. In § 2 we blend these two extensions, obtaining algorithms for solving (1.2) and (1.3), which include those in [5], [4], [12] as special cases. We also show how to compute the residual vector of the dual system efficiently using a generalisation of the Clenshaw recurrence.

In § 3 we present an error analysis of the algorithms of § 2. The analysis provides bounds for both the forward error and the residual of the computed solutions. It makes no assumptions about the ordering or signs of the points α_i, and thus extends the error analysis in [11].

To interpret the analysis we compare the error bounds with appropriate "ideal" bounds. This leads, in § 4, to pleasing stability results for certain classes of problem, but also reveals grave instabilities in some other cases. The instabilities can be interpreted as indicating that the natural, increasing ordering of the points can be a poor one. In § 5 we derive a technique for computing a more generally appropriate ordering. The method is based on a connection derived between the stability of the fast algorithms and the stability of Gaussian elimination. As a means for restoring stability, the re-ordering approach has several advantages over iterative refinement in single precision, which was used in [12] and [21].

Numerical experiments are presented in § 6. Finally, in § 7 we offer recommendations on the use of the fast algorithms for solving Vandermonde-like systems in a stable manner.

2. Algorithms. Assume that the polynomials $p_k(t)$ satisfy the three-term recurrence relation

$$p_{j+1}(t) = \theta_j(t-\beta_j)p_j(t) - \gamma_j p_{j-1}(t), \qquad j \geqq 1, \tag{2.1a}$$

with

$$p_0(t) = 1, \qquad p_1(t) = \theta_0(t-\beta_0)p_0(t), \tag{2.1b}$$

where $\theta_j \neq 0$ for all j. Algorithms for solving the systems (1.2) and (1.3) can be derived by using a combination of the techniques in [4] and [12]. Denote by $r(i) \geqq 0$ the smallest integer for which $\alpha_i = \alpha_{i-1} = \cdots = \alpha_{r(i)}$. Considering, first, the dual system (1.3), we note that

$$\phi(t) = \sum_{i=0}^{n} a_i p_i(t) \tag{2.2}$$

satisfies

$$\phi^{(i-r(i))}(\alpha_i) = f_i, \qquad 0 \leqq i \leqq n.$$

Thus ϕ is a Hermite interpolating polynomial for the data $\{\alpha_i, f_i\}$, and our task is to obtain its representation in terms of the basis $\{p_i(t)\}_{i=0}^{n}$. As a first step, following [4], we construct the divided difference form of ϕ:

$$\phi(t) = \sum_{i=0}^{n} c_i \prod_{j=0}^{i-1} (t-\alpha_j). \tag{2.3}$$

The (confluent) divided differences $c_i = f[\alpha_0, \alpha_1, \cdots, \alpha_i]$ may be generated using the recurrence relation [4], [20, p. 55]

(2.4)

$$f[\alpha_{j-k-1}, \cdots, \alpha_j] = \begin{cases} \dfrac{f[\alpha_{j-k}, \cdots, \alpha_j] - f[\alpha_{j-k-1}, \cdots, \alpha_{j-1}]}{\alpha_j - \alpha_{j-k-1}}, & \alpha_j \neq \alpha_{j-k-1}, \\ \dfrac{f_{r(j)+k+1}}{(k+1)!}, & \alpha_j = \alpha_{j-k-1}. \end{cases}$$

Now we need to generate the a_i in (2.2) from the c_i in (2.3). We can use the recurrences in [12], which are unaffected by confluency; these are derived by expanding (2.3) using nested multiplication, and using the recurrence relations (2.1) to express the results as a linear combination of the polynomials p_j.

In the following algorithm Stage I computes the confluent divided differences. We use an implementation of (2.4) from [6, pp. 68–69], in preference to the more complicated version in [4]. Stage II is identical to the corresponding part of the dual algorithm in [12].

ALGORITHM 2.1 (Dual, $P^T a = f$). Given parameters $\{\theta_j, \beta_j, \gamma_j\}_{j=0}^{n-1}$, a vector f, and points $\{\alpha_i\}_{i=0}^{n}$ satisfying (1.1), this algorithm solves the dual system $P^T a = f$.

Stage I: Set $c = f$
For $k = 0$ to $n - 1$
  $clast = c_k$
  For $j = k + 1$ to n
    If $\alpha_j = \alpha_{j-k-1}$ then
      $c_j = c_j/(k + 1)$
    else
      $temp = c_j$
      $c_j = (c_j - clast)/(\alpha_j - \alpha_{j-k-1})$
      $clast = temp$
    endif
  endfor j
endfor k

Stage II: Set $a = c$
$a_{n-1} = a_{n-1} + (\beta_0 - \alpha_{n-1})a_n$
$a_n = a_n/\theta_0$
For $k = n - 2$ to 0 step -1
  $a_k = a_k + (\beta_0 - \alpha_k)a_{k+1} + (\gamma_1/\theta_1)a_{k+2}$
  For $j = 1$ to $n - k - 2$
    $a_{k+j} = a_{k+j}/\theta_{j-1} + (\beta_j - \alpha_k)a_{k+j+1} + (\gamma_{j+1}/\theta_{j+1})a_{k+j+2}$
  endfor j
  $a_{n-1} = a_{n-1}/\theta_{n-k-2} + (\beta_{n-k-1} - \alpha_k)a_n$
  $a_n = a_n/\theta_{n-k-1}$
endfor k □

In the algorithm the vectors c and a have been used for clarity; in fact both can be replaced by f, so that the right-hand side is transformed into the solution without using any extra storage. Assuming that the values γ_j/θ_j are given (note that γ_j appears only in

the terms γ_j/θ_j) the computational cost of Algorithm 2.1 is $n(2n + 1)M$ and at most $n(5n + 3)/2A$, where M denotes a multiplication or division, and A an addition or subtraction.

An algorithm for solving the primal system can be deduced immediately, using the approach of [4], [5], [12]. We will show in § 3 that the dual algorithm effectively multiplies the right-hand side vector f by P^{-T}, employing a factorisation of P^{-T} into the product of $2n$ triangular matrices. Taking the transpose of this product we obtain a representation of P^{-1}, from which it is easy to write an algorithm for computing $x = P^{-1}b$.

ALGORITHM 2.2 (Primal, $Px = b$). Given parameters $\{\theta_j, \beta_j, \gamma_j\}_{j=0}^{n-1}$, a vector b, and points $\{\alpha_i\}_{i=0}^{n}$ satisfying (1.1), this algorithm solves the primal system $Px = b$.

Stage I: Set $d = b$
For $k = 0$ to $n - 2$
For $j = n - k$ to 2 step -1
$d_{k+j} = (\gamma_{j-1}/\theta_{j-1})d_{k+j-2} + (\beta_{j-1} - \alpha_k)d_{k+j-1} + d_{k+j}/\theta_{j-1}$
endfor j
$d_{k+1} = (\beta_0 - \alpha_k)d_k + d_{k+1}/\theta_0$
endfor k
$d_n = (\beta_0 - \alpha_{n-1})d_{n-1} + d_n/\theta_0$

Stage II: Set $x = d$
For $k = n - 1$ to 0 step -1
$xlast = 0$
For $j = n$ to $k + 1$ step -1
If $\alpha_j = \alpha_{j-k-1}$ then
$x_j = x_j/(k + 1)$
else
$temp = x_j/(\alpha_j - \alpha_{j-k-1})$
$x_j = temp - xlast$
$xlast = temp$
endif
endfor j
$x_k = x_k - xlast$
endfor k □

Algorithm 2.2 has, by construction, the same operation count (to within one addition) as Algorithm 2.1. Values of θ_j, β_j, γ_j for some polynomials of interest in Algorithms 2.1 and 2.2 are given in Table 2.1.

For practical use of Algorithms 2.1 and 2.2 it is important to be able to calculate the residual, in order to test that the algorithms have been coded correctly (for example) and, perhaps, to implement iterative refinement (see § 5). Ordinarily, residual computation for linear equations is trivial, but in this context the coefficient matrix is not given explicitly, and computing the residual turns out to be conceptually almost as difficult, and computationally as expensive, as solving the linear system!

To compute the residual for the dual system we need a means for evaluating $\phi(t)$ in (2.2) and its first $k \leqq n$ derivatives, where $k = \max_i(i - r(i))$ is the *order of confluency*. Since the polynomials p_j satisfy a three-term recurrence relation we can use an extension of the Clenshaw recurrence formula. The following algorithm implements the appropriate

TABLE 2.1
Parameters in the three term recurrence (2.1).

Polynomial	θ_j	β_j	γ_j	
Monomials	1	0	0	
Chebyshev	2*	0	1	$*\theta_0 = 1$
Legendre*	$\frac{2j+1}{j+1}$	0	$\frac{j}{j+1}$	$*p_j(1) \equiv 1$
Hermite	2	0	$2j$	
Laguerre	$-\frac{1}{j+1}$	$2j+1$	$\frac{j}{j+1}$	

recurrences, which are given in [18]; we note that an alternative derivation to that in [18] is to differentiate repeatedly the original Clenshaw recurrence and to rescale so as to consign factorial terms to a "clean-up" loop at the end.

ALGORITHM 2.3 (Extended Clenshaw recurrence [18]). This algorithm computes the $k+1$ values $y_j = \phi^{(j)}(x)$, $0 \leqq j \leqq k$, where ϕ is given by (2.2) and $k \leqq n$. It uses a work vector z of order k.

```
Set y_i = z_i = 0        (i = 0, 1, ..., k)
y_0 = a_n
For j = n - 1 to 0 step -1
    temp = y_0
    y_0 = θ_j(x - β_j)y_0 - γ_{j+1}z_0 + a_j
    z_0 = temp
    For i = 1 to min (k, n - j)
        temp = y_i
        y_i = θ_j((x - β_j)y_i + z_{i-1}) - γ_{j+1}z_i
        z_i = temp
    endfor i
endfor j
m = 1
For i = 2 to k
    m = m*i
    y_i = m*y_i
endfor i      □
```

Cost. $3[n + kn - k(k-1)/2](M + A) + 2 \max\{0, k-1\}M$.

Computing the residual using Algorithm 2.3 costs between approximately $3n^2/2(M+A)$ (for full confluency) and $3n^2(M+A)$ (for the nonconfluent case).

The residual for the primal system can be computed in a similar way, using recurrences obtained by differentiating (2.1).

3. Rounding error analysis. In this section we derive bounds for the forward error and the residual of the computed solution obtained from Algorithm 2.1 in floating point arithmetic. Because of the inherent duality between Algorithms 2.1 and 2.2 all the results that we state have obvious counterparts for Algorithm 2.2.

The key to the analysis is the observation that Algorithm 2.1 can be expressed entirely in the language of matrix-vector products (a similar observation drives the analysis

of a related problem in [19]). In Stage I, letting $c^{(k)}$ denote the vector c at the start of the kth iteration of the outer loop, we have

$$c^{(0)} = f, \qquad c^{(k+1)} = L_k c^{(k)}, \quad k = 0, 1, \cdots, n-1. \tag{3.1}$$

We will adopt the convention that the subscripts of all vectors and matrices run from 0 to n. The matrix L_k is lower triangular and agrees with the identity matrix in rows 0 to k. The remaining rows can be described by, for $k+1 \leqq j \leqq n$,

$$e_j^T L_k = \begin{cases} e_j^T/(k+1) & \text{if } \alpha_j = \alpha_{j-k-1}, \\ (e_j^T - e_s^T)/(\alpha_j - \alpha_{j-k-1}) & \text{for some } s<j, \text{ otherwise,} \end{cases}$$

where e_j is column j of the identity matrix. Similarly, Stage II can be expressed as

$$a^{(n)} = c^{(n)}, \qquad a^{(k)} = U_k a^{(k+1)}, \quad k = n-1, n-2, \cdots, 0. \tag{3.2}$$

The matrix U_k is upper triangular, it agrees with the identity matrix in rows 0 to $k-1$ and it has zeros everywhere above the first two superdiagonals.

From (3.1) and (3.2) we see that the overall effect of the Algorithm 2.1 is to evaluate step by step the product

$$a = U_0 \cdots U_{n-1} L_{n-1} \cdots L_0 f \equiv P^{-T} f. \tag{3.3}$$

We adopt the standard model of floating point arithmetic [6, p. 9]:

$$\mathrm{fl}(x \text{ op } y) = (x \text{ op } y)(1+\delta), \quad |\delta| \leqq u, \quad \text{op} = +, -, *, /, \tag{3.4}$$

where u is the unit roundoff. In line with the general philosophy of rounding error analysis we do not aim for the sharpest possible constants in our bounds, and are thus able to keep the analysis quite short.

THEOREM 3.1. *Let Algorithm* 2.1 *be applied in floating point arithmetic to floating point data* $\{\alpha_i, f_i\}_{i=0}^n$. *Provided that no overflows are encountered the algorithm runs to completion, and the computed solution* $\hat{a}$ *satisfies*

$$|\hat{a} - a| \leqq c(n,u)|U_0| \cdots |U_{n-1}||L_{n-1}| \cdots |L_0||f|,$$

where, with $\mu = (1+u)^4 - 1$, $c(n, u) = (1+\mu)^{2n} - 1 = 8nu + O(u^2)$.

Proof. First, note that Algorithm 2.1 must succeed in the absence of overflow, because division by zero cannot occur.

Because of the form of L_k, straightforward application of the model (3.4) to the components of (3.1) yields

$$\hat{c}^{(k+1)} = D_k L_k \hat{c}^{(k)}, \tag{3.5}$$

where $D_k = \mathrm{diag}(d_i)$, with $d_i = 1$ for $0 \leqq i \leqq k$, and $(1-u)^3 \leqq d_i \leqq (1+u)^3$ for $k+1 \leqq i \leqq n$. Thus

$$|D_k - I| \leqq [(1+u)^3 - 1] I,$$

and hence (3.5) may be written in the form

$$\hat{c}^{(k+1)} = (L_k + \Delta L_k)\hat{c}^{(k)}, \qquad |\Delta L_k| \leqq [(1+u)^3 - 1]|L_k|. \tag{3.6}$$

Turning to (3.2), we can regard the multiplication $a^{(k)} = U_k a^{(k+1)}$ as comprising a sequence of three-term inner products. Analysing these in standard fashion, using (3.4), we arrive at the equation

$$\hat{a}^{(k)} = (U_k + \Delta U_k)\hat{a}^{(k+1)}, \qquad |\Delta U_k| \leqq [(1+u)^4 - 1]|U_k|, \tag{3.7}$$

where we have taken into account the rounding errors in forming $u_{i,i+1}^{(k)} = \beta_j - \alpha_k$ and $u_{i,i+2}^{(k)} = \gamma_{j+1}/\theta_{j+1}$ $(i = k + j)$.

Since $\hat{c}^{(0)} = f$, and $\hat{a} = \hat{a}^{(0)}$, (3.6) and (3.7) imply that

$$\hat{a} = (U_0 + \Delta U_0)\cdots(U_{n-1} + \Delta U_{n-1})(L_{n-1} + \Delta L_{n-1})\cdots(L_0 + \Delta L_0)f, \tag{3.8}$$

where, on weakening (3.6), we have

$$|\Delta U_k| \leqq \mu |U_k|, \quad |\Delta L_k| \leqq \mu |L_k|, \quad \mu = (1+u)^4 - 1.$$

Now we make use of the following perturbation result that is easily proved by induction: *For matrices $X_j + \Delta X_j$, if $|\Delta X_j| \leqq \delta |X_j|$ for all j, then*

$$\left| \prod_{j=0}^{m} (X_j + \Delta X_j) - \prod_{j=0}^{m} X_j \right| \leqq [(1+\delta)^{m+1} - 1] \prod_{j=0}^{m} |X_j|.$$

Applying this result to the difference of (3.8) and (3.3), we obtain the desired bound for the forward error. □

In the course of proving Theorem 3.1 we derived (3.8), a form of backward error result. However, (3.8) is of little intrinsic interest because the perturbations it contains are associated with the matrices U_k and L_k, and not in any exploitable way with the original data $\{\alpha_i, f_i\}$ (and, possibly, $\{\theta_j, \beta_j, \gamma_j\}$). The appropriate way to analyse backward error, as we will explain in § 4.2, is to look at the residual, $r = f - P^T a$ (cf. the similar approach taken in a different context in [7]). Rearranging (3.8),

$$f = (L_0 + \Delta L_0)^{-1}\cdots(L_{n-1} + \Delta L_{n-1})^{-1}(U_{n-1} + \Delta U_{n-1})^{-1}\cdots(U_0 + \Delta U_0)^{-1}\hat{a}. \tag{3.9}$$

From the proof of Theorem 3.1 we can show that

$$(L_k + \Delta L_k)^{-1} = L_k^{-1} + E_k, \qquad |E_k| \leqq [(1-u)^{-3} - 1]|L_k^{-1}|.$$

Strictly, an analogous bound for $(U_k + \Delta U_k)^{-1}$ does not hold, since ΔU_k cannot be expressed in the form of a diagonal matrix times U_k. However, it seems reasonable to make a simplifying assumption that such a bound is valid, say,

$$(U_k + \Delta U_k)^{-1} = U_k^{-1} + F_k, \qquad |F_k| \leqq [(1-u)^{-4} - 1]|U_k^{-1}|. \tag{3.10}$$

Then, writing (3.9) as

$$\begin{aligned} f &= (L_0^{-1} + E_0)\cdots(L_{n-1}^{-1} + E_{n-1})(U_{n-1}^{-1} + F_{n-1})\cdots(U_0^{-1} + F_0)\hat{a} \\ &= P^T\hat{a} + \Bigg(\sum_{k=0}^{n-1} L_0^{-1}\cdots L_{k-1}^{-1} E_k L_{k+1}^{-1}\cdots L_{n-1}^{-1} U_{n-1}^{-1}\cdots U_0^{-1} \\ &\qquad + \sum_{k=0}^{n-1} L_0^{-1}\cdots L_{n-1}^{-1} U_{n-1}^{-1}\cdots U_{k+1}^{-1} F_k U_{k-1}^{-1}\cdots U_0^{-1} \Bigg)\hat{a} + O(u^2), \end{aligned}$$

we obtain the following result.

THEOREM 3.2. *Under the assumption* (3.10), *the residual of the computed solution $\hat{a}$ from Algorithm* 2.1 *is bounded by*

$$|f - P^T\hat{a}| \leqq d_n u |L_0^{-1}|\cdots|L_{n-1}^{-1}||U_{n-1}^{-1}|\cdots|U_0^{-1}||\hat{a}| + O(u^2),$$

with $d_n = 7n$. □

In common with most error analyses the one above uses a profusion of triangle and submultiplicative inequalities, and consequently the bounds will usually be unrealistic

error *estimates*. However, as we will see, they are well able to reveal extremes of behaviour, with respect to accuracy and stability.

4. Implications for stability. Now we pursue the implications of the error analysis. To interpret the forward error bound of Theorem 3.1 and the backward error bound of Theorem 3.2 we need to use two different notions of stability. We consider these separately in §§ 4.1 and 4.2, since there is no simple relation between them and each is of independent interest. We will focus attention mainly on the nonconfluent case, making brief comments about the effects of confluency.

4.1. Weak stability. To interpret the forward error bound

$$|\hat{a}-a| \leqq c(n,u)|U_0|\cdots|U_{n-1}||L_{n-1}|\cdots|L_0||f| \tag{4.1}$$

from Theorem 3.1 we need an "ideal" bound with which to compare it. Following the approach of [11, § 4] we consider the effect of a small, element-wise perturbation in f. If $P^T(a+\delta a) = f + \delta f$ with $|\delta f| \leqq u|f|$, then it is easy to show that

$$|\delta a| \leqq u|P^{-T}||f|, \tag{4.2}$$

and that equality is attained for suitable choice of δf. This prompts the informal definition that an algorithm for solving $P^T a = f$ in floating point arithmetic is *weakly stable* if the error in the computed solution is not much larger, in some appropriate measure, than the upper bound in (4.2). A useful way to interpret the definition is that if the machine right-hand side vector is inexact, then a weakly stable algorithm solves the machine problem to as good an accuracy as the data warrants.

By comparing (4.1) and (4.2) we see that Algorithm 2.1 is certainly weakly stable if

$$|U_0|\cdots|U_{n-1}||L_{n-1}|\cdots|L_0| \leqq b_n|P^{-T}| = b_n|U_0\cdots U_{n-1}L_{n-1}\cdots L_0| \tag{4.3}$$

for some small constant $b_n \geqq 1$. This condition requires that there be little subtractive cancellation in the product $U_0\cdots U_{n-1}L_{n-1}\cdots L_0$. Suppose the points are distinct and consider the case $n = 3$. We have

$$\begin{aligned}
P^{-T} &= U_0U_1U_2L_2L_1L_0 \\
&\equiv \begin{bmatrix} 1 & \beta_0-\alpha_0 & \gamma_1/\theta_1 & 0 \\ & \theta_0^{-1} & \beta_1-\alpha_0 & \gamma_2/\theta_2 \\ & & \theta_1^{-1} & \beta_2-\alpha_0 \\ & & & \theta_2^{-1} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ & 1 & \beta_0-\alpha_1 & \gamma_1/\theta_1 \\ & & \theta_0^{-1} & \beta_1-\alpha_1 \\ & & & \theta_1^{-1} \end{bmatrix} \\
&\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & \beta_0-\alpha_2 \\ & & & \theta_0^{-1} \end{bmatrix} \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & 0 & -1/(\alpha_3-\alpha_0) & 1/(\alpha_3-\alpha_0) \end{bmatrix} \\
&\times \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & -1/(\alpha_2-\alpha_0) & 1/(\alpha_2-\alpha_0) & \\ 0 & 0 & -1/(\alpha_3-\alpha_1) & 1/(\alpha_3-\alpha_1) \end{bmatrix} \\
&\times \begin{bmatrix} 1 & & & \\ -1/(\alpha_1-\alpha_0) & 1/(\alpha_1-\alpha_0) & & \\ 0 & -1/(\alpha_2-\alpha_1) & 1/(\alpha_2-\alpha_1) & \\ 0 & 0 & -1/(\alpha_3-\alpha_2) & 1/(\alpha_3-\alpha_2) \end{bmatrix}.
\end{aligned} \tag{4.4}$$

There is no subtractive cancellation in this product as long as each matrix has the alternating sign pattern defined, for $A = (a_{ij})$, by $(-1)^{i+j}a_{ij} \geq 0$. This sign pattern holds for the matrices L_i if the points α_i are arranged in increasing order. The matrices U_i have the required sign pattern provided that (in general)

$$\theta_i > 0, \qquad \gamma_i \geq 0 \quad \text{for all } i, \quad \text{and} \quad \beta_i - \alpha_k \leq 0 \quad \text{for all } i+k \leq n-1. \tag{4.5}$$

Hence we obtain the following result, where we weaken the last condition to $\beta_i = 0$ and $\alpha_i \geq 0$ for all i since $\beta_i = 0$ holds for most of the commonly occurring polynomials.

COROLLARY 4.1. *If* $0 \leq \alpha_0 < \alpha_1 < \cdots < \alpha_n$, *and* $\theta_i > 0$, $\beta_i = 0$ *and* $\gamma_i \geq 0$ *for all* i, *then*

$$|\hat{a} - a| \leq c(n,u)|P^{-T}||f|,$$

where $c(n, u)$ *is defined in Theorem* 3.1, *and hence, under these conditions, Algorithm* 2.1 *is weakly stable.* □

Corollary 4.1 is stated without proof in [12, § 3]. In the special case of the monomials ($\theta_i = 1$, $\beta_i = \gamma_i = 0$) Corollary 4.1 is essentially the same as the main result of [11, Thm. 2.3]. As shown in [11], the bound in the corollary can imply high relative accuracy even when P^{-T} is large. To see this, note that under the conditions of the corollary P^{-T} has the alternating sign pattern, since each of its factors does; thus if $(-1)^i f_i \geq 0$ then $|P^{-T}||f| = |P^{-T}f| = |a|$, and the corollary implies that $\hat{a}$ is accurate essentially to full machine precision.

The nonnegativity condition on the points α_i in Corollary 4.1 is rather restrictive, since points of both signs are likely to occur in practice. Suppose, then, that we alter the conditions of Corollary 4.1 to allow that

$$\alpha_0 < \cdots < \alpha_m < 0 \leq \alpha_{m+1} < \cdots < \alpha_n, \qquad 0 \leq m \leq n-1. \tag{4.6}$$

The matrices L_i retain the alternating sign pattern, as do $U_{m+1}, \cdots, U_{n-1}$. But $U_0, \cdots, U_m$ lose the sign property, and so there is subtractive cancellation within the product $U_0 \cdots U_{n-1} L_{n-1} \cdots L_0$. It is possible to derive an a priori bound for the effect of this cancellation. For example, we have $U_i \geq 0$ for $0 \leq i \leq m$, and so

$$\begin{aligned} |U_0| \cdots |U_{n-1}||L_{n-1}| \cdots |L_0| &= U_0 \cdots U_m |U_{m+1} \cdots U_{n-1} L_{n-1} \cdots L_0| \\ &= B|B^{-1}BU_{m+1} \cdots U_{n-1} L_{n-1} \cdots L_0| \\ &= B|B^{-1}P^{-T}| \\ &\leq B|B^{-1}||P^{-T}|, \end{aligned} \tag{4.7}$$

where $B = U_0 \cdots U_m$. However, in our experience this inequality is quite weak, and to obtain a manageable bound for the term $B|B^{-1}|$ would produce a substantial further weakening. Therefore we adopt an empirical approach.

For various distributions of distinct points $\alpha_i \in [-1, 1]$, ordered according to (4.6), we evaluated for the monomials, and for the Chebyshev polynomials $T_k(t)$, the ratio

$$q_n = \frac{\| |U_0| \cdots |U_{n-1}||L_{n-1}| \cdots |L_0| \|_\infty}{\|P^{-T}\|_\infty} \geq 1. \tag{4.8}$$

This quantity is a norm-wise analogue of b_n in (4.3); we have taken norms because for points satisfying (4.6) inequality (4.3) can fail to hold for any b_n, since P^{-T} can have a zero element while the lower bound matrix in (4.3) has a nonzero in the same position. Note that q_n can be interpreted as a measure of the sensitivity of the factorisation $P^{-T} = U_0 \cdots U_{n-1} L_{n-1} \cdots L_0$ to perturbations in the factors. Loosely, for a particular

problem we would expect Algorithm 2.1 to perform in a weakly stable manner only if q_n is not too large compared to one.

Values of q_n, together with the condition number $\kappa_\infty(P^T) = \|P^T\|_\infty \|P^{-T}\|_\infty$, are presented for two representative point distributions in Figs. 4.1 and 4.2. The q_n values for the monomials are reasonably small, but suggest some potential instability. More seriously, the results indicate severe instability of Algorithm 2.1 for the Chebyshev polynomials; for example, with $n = 30$ and α_i the extrema of T_n, there is a potential loss of up to 14 figures in solving an almost perfectly conditioned linear system (cf. problem (6.3)). Instability of this magnitude was diagnosed in [12], and a heuristic explanation is given there. The present analysis reveals the source of the problem: the matrix factorisation at the heart of Algorithm 2.1 is, in some cases, an unstable one, in the sense that the product is unduly sensitive to small perturbations in the factors.

If the order of confluency k is positive, and the points are in increasing order, then the alternating sign condition fails to hold for at least one of $L_0, \cdots, L_{k-1}$. A result similar to Corollary 4.1 can be obtained using the technique employed in (4.7). For example, if $k = 1$ then the bound in Corollary 4.1 can be replaced by

$$|\hat{a} - a| \leqq c(n,u)|P^{-T}||M||f|,$$

where $M = |L_0^{-1}||L_0|$ is unit lower triangular and satisfies $|m_{ij}| \leqq 2$.

4.2. Backward stability. We turn now to the residual bound in Theorem 3.2:

$$|f - P^T\hat{a}| \leqq d_n u |L_0^{-1}| \cdots |L_{n-1}^{-1}||U_{n-1}^{-1}| \cdots |U_0^{-1}||\hat{a}| + O(u^2). \tag{4.9}$$

For comparison, if $\tilde{a}$ agrees with a to working precision (e.g., $\tilde{a} = \mathrm{fl}(a)$) then

$$a = \tilde{a} + \delta\tilde{a}, \qquad |\delta\tilde{a}| \leqq u|\tilde{a}|,$$

and so

$$|f - P^T\tilde{a}| = |P^T\delta\tilde{a}| \leqq u|P^T||\tilde{a}|. \tag{4.10}$$

We take (4.10), and the norm-wise version

$$\|f - P^T\tilde{a}\|_\infty \leqq u\|P^T\|_\infty\|\tilde{a}\|_\infty, \tag{4.11}$$

as our model bounds for the residual vector. Connections with the usual notion of backward error are that (4.10) is true if and only if, for some E [16], [17],

$$(P^T + E)\tilde{a} = f, \qquad |E| \leqq u|P^T|, \tag{4.12}$$

and (4.11) implies

$$(P^T + F)\tilde{a} = f, \quad \|F\|_\infty \leqq n^{1/2}u\|P^T\|_\infty \quad (F = (f - P^T\tilde{a})\tilde{a}^T/\tilde{a}^T\tilde{a}). \tag{4.13}$$

Thus (4.10) and (4.11) are equivalent to the condition that $\tilde{a}$ is the solution of a linear system obtained from $P^Ta = f$ by slightly perturbing P^T, in the element-wise sense in (4.12), or the norm-wise sense in (4.13). Note, however, that these perturbed matrices are not, in general, Vandermonde-like matrices.

For the monomials, with distinct, nonnegative points arranged in increasing order, the matrices L_i and U_i are bidiagonal with the alternating sign pattern, as we have seen in § 4.1. Thus $L_i^{-1} \geqq 0$ and $U_i^{-1} \geqq 0$, and since $P^T = L_0^{-1}\cdots L_{n-1}^{-1}U_{n-1}^{-1}\cdots U_0^{-1}$, we obtain from (4.9) the following pleasing backward stability result.

COROLLARY 4.2. *Let* $0 \leqq \alpha_0 < \alpha_1 < \cdots < \alpha_n$, *and consider Algorithm* 2.1 *for the monomials. Under the assumption* (3.10), *the computed solution* $\hat{a}$ *satisfies*

$$|f - P^T\hat{a}| \leqq d_n u|P^T||\hat{a}| + O(u^2),$$

with $d_n = 7n$. □

To investigate the general case (permitting confluency) it is useful to approximate the matrix product in (4.9) by its lower bound in

$$|L|\,|U| := |L_0^{-1}\cdots L_{n-1}^{-1}|\,|U_{n-1}^{-1}\cdots U_0^{-1}| \leqq |L_0^{-1}|\cdots|L_{n-1}^{-1}|\,|U_{n-1}^{-1}|\cdots|U_0^{-1}|,$$

where $P^T = LU$ is an unnormalised LU factorisation. In so doing we make the residual bound smaller and so we are still able to draw conclusions from a large value for the bound. The significance of the approximation is that $|L|\,|U|$ is the matrix that appears in the backward error analysis of Gaussian elimination. For example, from [8] the LU factors $\hat{L}$ and $\hat{U}$ computed by Gaussian elimination without pivoting on $A \in \mathbf{R}^{n\times n}$ satisfy

$$\hat{L}\hat{U} = A + E, \qquad |E| \leqq \frac{nu}{1-nu}\,|\hat{L}|\,|\hat{U}|. \tag{4.14}$$

Using our approximation in the bound (4.9), we obtain

$$|f - P^T\hat{a}| \leqq d_n u\,|L|\,|U|\,|\hat{a}| + O(u^2) \qquad (P^T = LU). \tag{4.15}$$

The similarity of (4.14) and (4.15) suggests that the backward stability of Algorithm 2.1 is related to that of Gaussian elimination without pivoting on P^T. (Note that $|L|\,|U| = |LD|\,|D^{-1}U|$ for any diagonal D, so the normalisation of our LU factorisation is unimportant.) For the same polynomials and points as in Figs. 4.1 and 4.2, Figs. 4.3 and 4.4 show values of

$$g_n = \frac{\||L|\,|U|\|_\infty}{\|P^T\|_\infty} \geqq 1 \qquad (P^T = LU). \tag{4.16}$$

Again, the results predict serious instability of Algorithm 2.1 for the Chebyshev polynomials, and, to a somewhat lesser extent, for the monomials.

5. Preventing and curing instability. Although the increasing ordering for the points α_i yields the favourable stability results in Corollaries 4.1 and 4.2, this ordering is not universally appropriate for Algorithm 2.1, as evidenced by the instability for the Chebyshev

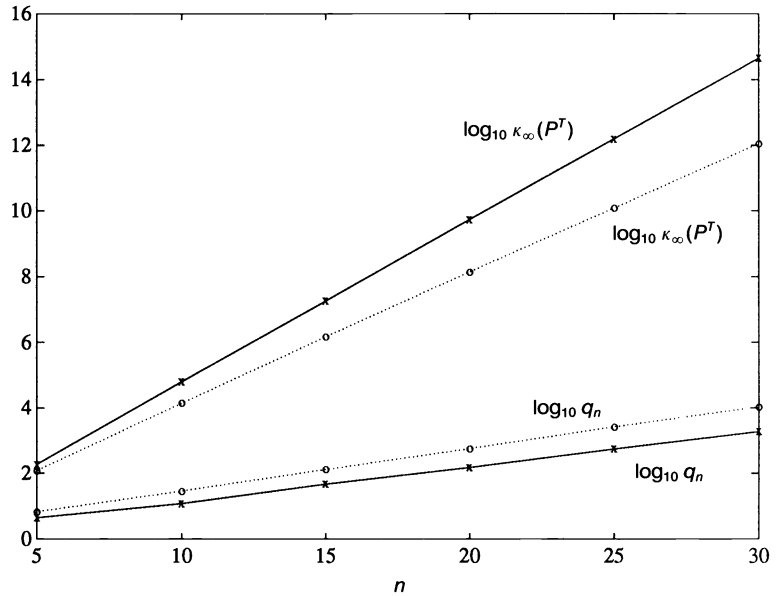


FIG. 4.1. *Monomials.* ×: $\alpha_i = -1 + 2i/n$; ○: $\alpha_{n-i} = \cos(i\pi/n)$ (*extrema of* T_n).

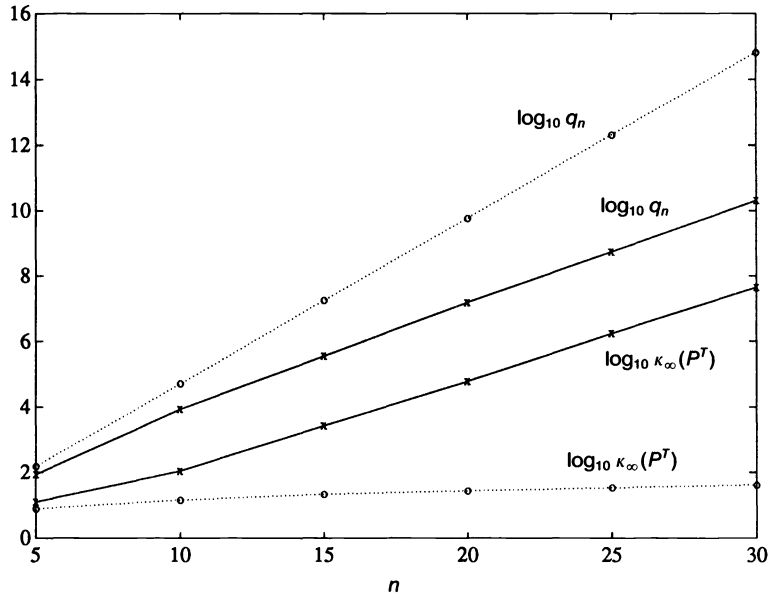


FIG. 4.2. *Chebyshev polynomials.* ×: $\alpha_i = -1 + 2i/n$; ○: $\alpha_{n-i} = \cos(i\pi/n)$ (*extrema of* T_n).

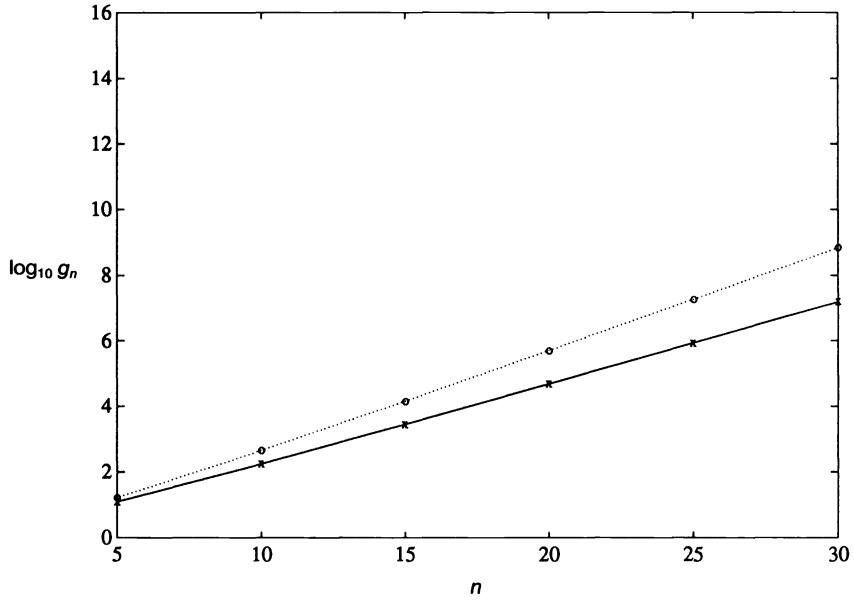


FIG. 4.3. *Monomials.* ×: $\alpha_i = -1 + 2i/n$; ○: $\alpha_{n-i} = \cos(i\pi/n)$ (*extrema of* T_n).

polynomials when there are points of both signs. How, then, in general, can we construct a "good" ordering of the points?

Consider the nonconfluent case. We suggest the following approach that exploits the connection with Gaussian elimination exposed in (4.14) and (4.15). The bound (4.15) suggests that to make Algorithm 2.1 backward stable the points should be ordered

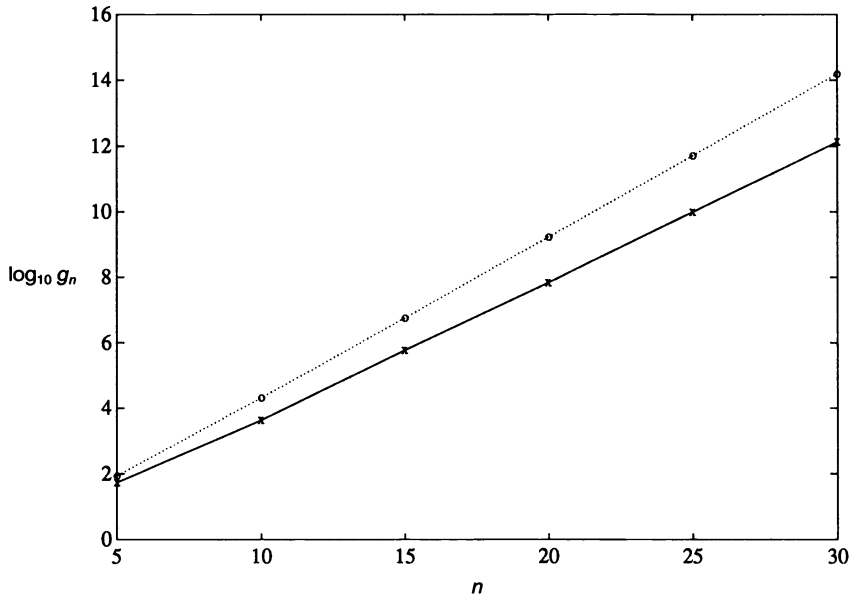


FIG. 4.4. *Chebyshev polynomials*. ×: $\alpha_i = -1 + 2i/n$; ○: $\alpha_{n-i} = \cos(i\pi/n)$ (*extrema of* T_n).

so that g_n in (4.16) is reasonably small. But re-ordering the points is equivalent to permuting the rows of P^T, and as is well known, Gaussian elimination with partial pivoting is a very successful way to obtain a row permutation that keeps g_n small. Now we make the crucial observation that the permutation that would be produced by Gaussian elimination with partial pivoting on P^T can be computed in $O(n^2)$ operations, *without actually performing the elimination*. To see this, recall that $P^T = L_0^{-1} \cdots L_{n-1}^{-1} U_{n-1}^{-1} \cdots U_0^{-1} \equiv LU$, and so if we take L unit lower triangular then (cf. the inverse of (4.4))

$$u_{ii} = h_i \prod_{j=0}^{i-1} (\alpha_i - \alpha_j),$$

where h_i depends only on the θ_i. At the kth stage of Gaussian elimination on P^T the partial pivoting strategy interchanges rows k and r, where $|u_{rk}| = \max_{s \geq k} |u_{sk}|$. Because of the equivalence between interchanges among the rows of P^T and among the points α_i, it follows that r is characterised by

$$\left| \prod_{j=0}^{k-1} (\alpha_r - \alpha_j) \right| = \max_{s \geq k} \left| \prod_{j=0}^{k-1} (\alpha_s - \alpha_j) \right|.$$

This relation forms the basis for the next algorithm.

ALGORITHM 5.1. Given distinct points $\alpha_0, \alpha_1, \cdots, \alpha_n$, this algorithm re-orders the points according to the same permutation that would be produced by Gaussian elimination with partial pivoting applied to $P^T(\alpha_0, \alpha_1, \cdots, \alpha_n)$ (but see below). The permutation is recorded in the vector p.

Swap (α_0, α_j) where $\alpha_j = \min_{i \geq 0} \alpha_i$, $\quad p_0 = j$
Swap (α_1, α_j) where $\alpha_j = \max_{i \geq 1} \alpha_i$, $\quad p_1 = j$
$\pi_i = \alpha_i - \alpha_0 \quad (i = 2, 3, \cdots, n)$

For $k = 2$ to $n - 1$
  $\pi_i = \pi_i * (\alpha_i - \alpha_{k-1})$ $(i = k, \cdots, n)$
  Find j where $\pi_j = \max_{i \geq k} |\pi_i|$
  Swap (α_k, α_j); Swap (π_k, π_j), $p_k = j$
endfor k

Cost. Approximately $n^2/2$ multiplications and comparisons.

In fact, Algorithm 5.1 does slightly more than imitate partial pivoting since it chooses α_0 and α_1, rather than just α_1, to maximise the (1, 1) pivot $\alpha_1 - \alpha_0$. This has the desirable effect of making the output of the algorithm independent of the initial ordering of the points.

If we apply the heuristic that $g_n \approx 1$ for Gaussian elimination with partial pivoting, then from (4.15) we obtain for the ordering of Algorithm 5.1 the approximate residual bound

$$\|f - P^T\hat{a}\|_\infty \leqq d_n u \|P^T\|_\infty \|\hat{a}\|_\infty + O(u^2).$$

Thus, under the several assumptions leading to (4.15), the ordering of Algorithm 5.1 renders Algorithm 2.1 (and similarly Algorithm 2.2) backward stable.

We note that Algorithm 5.1 never produces the increasing ordering, since it sets $\alpha_1 := \max_i \alpha_i$. It is also interesting to note that Algorithm 5.1 is invariant under the linear transformation of the points $\alpha_i := \mu\alpha_i + \lambda$.

An alternative approach to achieving backward stability is to take an arbitrary ordering of the points and to follow Algorithm 2.1 with one step of iterative refinement in single precision. This approach, advocated for general linear equation solvers in [13], was used successfully with the nonconfluent version of Algorithm 2.1, with Chebyshev polynomials, in [12]. However, we have no rigorous forward error bounds or residual bounds for Algorithm 2.1 combined with iterative refinement.

In terms of computational cost the re-ordering strategy is preferable to iterative refinement, since it requires only $5n^2/2$ multiplications in total, compared to the $7n^2$ multiplications required for two invocations of Algorithm 2.1 and a residual vector computation. Moreover, in some applications a sequence of problems with the same, or slightly changed, sequence of points may arise, in which case the re-ordering strategy need be applied only once for the whole sequence.

In the confluent case Algorithm 5.1 can be applied to the *distinct* subset of the points, with groups of equal points interchanged block-wise (since condition (1.1) must be maintained). Note, however, that in this form the algorithm no longer mimics the partial pivoting interchanges, and so the theoretical support is weaker.

6. Numerical experiments. We have carried out a wide variety of numerical experiments to test the analysis of §§ 3–5, and to gain further insight into the behaviour of Algorithm 2.1; we present detailed results for a subset of the tests in this section. The tests were done using Borland Turbo Basic on a PC-AT compatible machine. Turbo Basic uses IEEE-standard single and double precision arithmetic, for which the unit roundoffs are $u_{sp} = 2^{-23} \approx 1.19 \times 10^{-7}$ and $u_{dp} = 2^{-52} \approx 2.22 \times 10^{-16}$, respectively.

We solved each test problem in single precision using each of the following four schemes, which we will refer to by the mnemonics indicated.

(1) Alg: Algorithm 2.1.
(2) Ord: Algorithm 2.1 preceded by Algorithm 5.1.
(3) Sir: Algorithm 2.1 followed by one step of iterative refinement with the residual computed in single precision using Algorithm 2.3.

(4) Gepp: Gaussian elimination with partial pivoting, where P^T is formed in double precision using repeated calls to Algorithm 2.3 (with $x = \alpha_i$, and $a = e_j$ in (2.2)).

In all our test problems the points are in increasing order (of course this is irrelevant for Ord and Gepp). For each computed solution $\hat{a}$ we formed the norm-wise relative error

$$\text{ERR} = \frac{\|\hat{a} - a\|_\infty}{u_{\text{sp}} \|a\|_\infty}$$

and the relative residual

$$\text{RES} = \frac{\|f - P^T\hat{a}\|_\infty}{u_{\text{sp}} \|P^T\|_\infty \|\hat{a}\|_\infty}.$$

Here, $a := \hat{a}_{\text{dp}}$ is the solution computed by Algorithm 2.1 in double precision, and the residual $f - P^T\hat{a}$ is computed using Algorithm 2.3 in double precision. The order n was restricted to ensure that $\hat{a}_{\text{dp}}$ was correct to single precision, thus ensuring a correct value for ERR. Note that ERR and RES are scaled to be "independent of the machine precision"; thus both should be compared with 1 when assessing the accuracy of a computed solution or the size of its residual.

Two further quantities computed were the model bound for ERR, from (4.2),

$$w_n = \frac{\| \, |P^{-T}| \, |f| \, \|_\infty}{\|a\|_\infty},$$

and g_n in (4.16) (for the original, increasing ordering of the points).

The first problem,

$$\text{(6.1)} \qquad \text{Chebyshev polynomials} \quad \alpha_i = \frac{i}{n}, \quad f_i \sim \text{Unif}[-1, 1],$$

illustrates Corollary 4.1 (Unif denotes the uniform random number distribution); see Table 6.1. The excellent accuracy of Algorithm 2.1 is forecast by the corollary since, as is clear from the results, $\| \, |P^{-T}| \, |f| \, \|_\infty \approx \|a\|_\infty$ (a is a large-normed solution). Interestingly, the favourable forward error properties are seen to be lost in the process of iterative refinement, as has been observed in [12].

Next, we consider the monomials on problems with points of both signs. We tried a variety of problems, aiming to generate the instability that the analysis of § 4 predicts may occur for the monomials. In most problems, including all those from [5] and [11], Algorithm 2.1 performed in both a weakly stable and a backward stable manner, yielding

TABLE 6.1
Results for problem (6.1). *All values except n are logs to base* 10.

			ERR				RES					
n	$\kappa_\infty(P^T)$	$\|a\|_\infty$	Alg	Ord	Sir	Gepp	Alg	Ord	Sir	Gepp	w_n	g_n
10	8.6	6.5	−0.2	0.8	5.8	6.7	−1.6	−1.4	−1.5	−1.3	0.9	2.1
15	13.1	11.4	−0.3	0.4	10.3	6.9	−1.3	−1.6	−1.6	−1.3	0.1	3.5
20	17.6	13.9	1.0	1.6	14.7	6.9	−1.4	−1.8	−1.5	−1.7	2.1	5.2
25	22.1	19.6	0.2	0.8	19.3	6.9	−1.7	−1.6	−1.6	−0.9	0.8	6.4

TABLE 6.2
Results for problem (6.2). *All values except n are logs to base* 10.

			ERR				RES					
n	$\kappa_\infty(P^T)$	$\|a\|_\infty$	Alg	Ord	Sir	Gepp	Alg	Ord	Sir	Gepp	w_n	g_n
10	4.8	0.0	1.1	1.8	0.9	−0.3	−0.2	−1.5	−1.7	−3.6	1.3	2.2
15	7.3	0.0	2.2	3.9	2.0	1.2	1.6	−1.1	−1.2	−2.9	2.3	3.4
20	9.7	0.0	3.8	6.4	2.8	1.9	3.1	−1.5	−1.9	−3.1	3.3	4.8
25	12.2	0.0	5.1	9.2	5.5	3.4	4.4	−2.1	−0.9	−3.0	4.4	5.2
30	14.7	0.0	6.1	11.6	9.3	4.4	5.4	−1.8	1.2	−2.9	5.5	6.1

TABLE 6.3
Results for problem (6.3). *All values except n are logs to base* 10.

			ERR				RES					
n	$\kappa_\infty(P^T)$	$\|a\|_\infty$	Alg	Ord	Sir	Gepp	Alg	Ord	Sir	Gepp	w_n	g_n
10	1.0	0.0	3.8	1.1	0.3	0.2	3.5	0.8	0.0	−0.2	0.4	4.2
15	1.2	0.0	6.5	1.0	0.3	0.5	5.9	0.3	0.1	−0.1	0.4	6.6
20	1.3	0.0	8.8	1.7	2.3	0.5	6.4	1.2	1.8	−0.1	0.4	9.1
25	1.4	0.0	10.9	2.1	6.5	1.9	6.5	1.4	5.8	0.1	0.5	10.3

ERR $\leqq w_n$, and RES = $O(1)$. On examining the error analysis we selected the problem

$$\text{monomials} \quad \alpha_i = -1 + \frac{2i}{n}, \quad f = P^T e_n, \tag{6.2}$$

reasoning that $a = e_n$ might "pick out" large elements in the matrix product in (4.9). The results, summarised in Table 6.2, do indeed display instability, principally in the residual, and they match well the predictions of the analysis, as can be seen by comparing the values of RES (for Alg) and g_n.

The problem

$$\text{Chebyshev polynomials} \quad \alpha_{n-i} = \cos\left(\frac{(i+\frac{1}{2})\pi}{n+1}\right), \quad f = P^T e, \tag{6.3}$$

in which the points are the zeros of T_{n+1}, illustrates the instability of Algorithm 2.1 for the Chebyshev polynomials when there are points of both signs; the results are in Table 6.3. The re-ordering strategy successfully stabilises Algorithm 2.1, as does iterative refinement except at $n = 25$ (at this value even using double precision to compute the residual brought no further improvement). Note that because P is well conditioned [10], a small residual implies a small forward error in this problem.

Finally, we present two confluent problems. In these the order of confluency is four and the distinct points $\{\lambda_i\}_{i=0}^{d}$ occur in groups of successive sizes 4, 3, 2, 1, 4, 3, $\cdots$, where the obvious pattern repeats. The two problems are:

(6.4a) monomials
(6.4b) Chebyshev polynomials

$$\Big\} \lambda_{d-i} = \cos\left(\frac{i\pi}{d}\right), \quad i = 0, 1, \cdots, d, \quad f = e_n.$$

In Table 6.4 we see that both iterative refinement and the re-ordering approach behave very unstably on (6.4a) in the sense of weak stability; in our experience this instability

TABLE 6.4
Results for problem (6.4a). *All values except n are logs to base* 10.

			ERR				RES					
n	$\kappa_\infty(P^T)$	$\|a\|_\infty$	Alg	Ord	Sir	Gepp	Alg	Ord	Sir	Gepp	w_n	g_n
9	6.4	−0.3	0.3	3.7	0.3	3.5	−1.1	−0.9	−1.0	−1.2	0.0	1.0
19	12.3	2.3	1.0	6.4	6.1	6.9	−1.7	−1.8	−1.7	−1.5	0.0	3.0
29	17.4	5.7	2.8	10.6	10.9	6.9	−0.6	−2.3	0.4	−1.8	0.0	4.8

TABLE 6.5
Results for problem (6.4b). *All values except n are logs to base* 10.

			ERR				RES					
n	$\kappa_\infty(P^T)$	$\|a\|_\infty$	Alg	Ord	Sir	Gepp	Alg	Ord	Sir	Gepp	w_n	g_n
9	6.6	−0.6	0.5	2.5	0.5	1.8	−3.7	−2.8	−2.5	−2.5	0.0	1.3
19	9.4	−0.9	4.9	4.2	2.1	4.6	−0.8	−2.4	−3.6	−2.3	0.0	4.1
29	11.3	−1.1	9.9	8.2	9.8	5.7	0.7	−0.8	1.0	−2.3	0.0	5.8

is unusual for the latter scheme. Table 6.5 demonstrates clearly that weak stability is not implied by backward stability.

The complete set of test results contain several more features worth noting.

(1) The results for confluent problems were similar in most respects to those for nonconfluent ones; the behaviour of Algorithm 2.1 seems to be minimally affected by confluency. Test results for the Legendre polynomials were very similar in almost every respect to those for the Chebyshev polynomials.

(2) The growth quantity g_n for Gaussian elimination without pivoting is sometimes many orders of magnitude bigger than RES for Alg, but approximate equality can be attained, as in problem (6.2). This behaviour confirms our expectations—see the comment at the end of § 3.

(3) For the monomials our experience is that the forward error from Alg is usually similar to, or smaller than, the forward error from Ord.

(4) Unlike in the tests of [12], in which $u_{sp} \approx 10^{-15}$, we found that iterative refinement in single precision does not always yield a small residual (see Table 6.3, for example). This does not appear to be due to errors in computing the single precision residual via Algorithm 2.3, but seems to indicate that in order to guarantee the success of iterative refinement in single precision a certain level of precision is required relative to the degree of instability (indeed this is implied by the results in [13]).

(5) All our tests support the following heuristic, for which theoretical backing is easily given:

> The computed solution $\hat{x}$ from Gaussian elimination with partial pivoting applied to a linear system $Ax = b$ usually satisfies $\|\hat{x}\|_\infty \leqq u^{-1}\|b\|_\infty / \|A\|_\infty$, where u is the unit roundoff.

Thus, although Gaussian elimination with partial pivoting is guaranteed to produce a small residual, it is unable to solve accurately Vandermonde problems with a very large solution, such as problem (6.1). (Indeed, merely forming the machine matrix $\mathrm{fl}(P^T)$ may be enough to force $\|a\|_\infty \leqq u^{-1}\|f\|_\infty / \|P^T\|_\infty$ for the machine problem!)

7. Conclusions. To conclude, we offer some brief guidelines on the numerical solution of Vandermonde and Vandermonde-like systems. First, we caution that construction of algorithms that involve the solution of a Vandermonde-like system is not generally to be recommended. The tendency for Vandermonde matrices to be extremely ill-conditioned may render such an approach inherently unstable, in the sense that the "ideal" forward error bound (4.2) is unacceptably large; furthermore, as n increases the solution components may soon exceed the largest representable machine number, producing overflow. Despite these problems we have seen that many Vandermonde systems can be solved to surprisingly high accuracy using Algorithms 2.1 and 2.2. A useful rule of thumb is that it is those Vandermonde systems with a large-normed solution—one that reflects the size of P^{-1}—that are solved to high accuracy.

Our experience shows that of the four solution methods considered in § 6 (Alg, Ord, Sir, Gepp), none consistently produces the smallest forward error or the smallest relative residual. Nevertheless, the error analysis and the test results point to some clear recommendations for the choice of solution method. Recall that Alg denotes Algorithm 2.1 (or Algorithm 2.2) with the points arranged in increasing order, and Ord denotes Algorithm 2.1 (or Algorithm 2.2) preceded by Algorithm 5.1.

Monomials. Nonnegative points: Use Alg. In the nonconfluent case Corollaries 4.1 and 4.2 guarantee both weak and backward stability.

Points of both signs: (i) Use Alg. This usually behaves in a weakly stable and a backward stable manner. (ii) If it is vital to obtain a small residual use Ord, perhaps after first trying Alg. Note, however, that the forward error for Ord is usually no smaller, and sometimes larger, than that for Alg (see Tables 6.2 and 6.4).

Other polynomials. Nonnegative points: Use Alg. In the nonconfluent case Corollary 4.1 guarantees weak stability if $\theta_i > 0$, $\beta_i = 0$, and $\gamma_i \geqq 0$ in (2.1), as for the Chebyshev, Legendre, and Hermite polynomials.

Points of both signs: Use Ord (Alg is unstable).

If the points are all nonpositive then in both cases Alg should be used with the points in *decreasing* order (appropriate analogues of Corollaries 4.1 and 4.2 can be derived for this situation).

Acknowledgments. I am grateful to Professor Charles Clenshaw for pointing out reference [18], and to Des Higham for valuable comments on the manuscript.

REFERENCES

[1] M. ALMACANY, C. B. DUNHAM, AND J. WILLIAMS, *Discrete Chebyshev approximation by interpolating rationals*, IMA J. Numer. Anal., 4 (1984), pp. 467–477.

[2] 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, Dordrecht, the Netherlands, 1987, pp. 53–60.

[3] C. BALLESTER AND V. PEREYRA, *On the construction of discrete approximations to linear differential expressions*, Math. Comp., 21 (1967), pp. 297–302.

[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] S. D. CONTE AND C. DE BOOR, *Elementary Numerical Analysis*, 3rd ed., McGraw-Hill, Tokyo, 1980.

[7] G. CYBENKO, *The numerical stability of the Levinson–Durbin algorithm for Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 1 (1980), pp. 303–319.

[8] C. DE BOOR AND A. PINKUS, *Backward error analysis for totally positive linear systems*, Numer. Math., 27 (1977), pp. 485–490.

[9] W. GAUTSCHI, *On inverses of Vandermonde and confluent Vandermonde matrices*, Numer. Math., 4 (1962), pp. 117–123.

[10] ———, *The condition of Vandermonde-like matrices involving orthogonal polynomials*, Linear Algebra Appl., 52/53 (1983), pp. 293–300.

[11] N. J. HIGHAM, *Error analysis of the Björck–Pereyra algorithms 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] M. JANKOWSKI AND H. WOŹNIAKOWSKI, *Iterative refinement implies numerical stability*, BIT, 17 (1977), pp. 303–311.

[14] J. KAUTSKY AND S. ELHAY, *Calculation of the weights of interpolatory quadratures*, Numer. Math., 40 (1982), pp. 407–422.

[15] 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, Dordrecht, the Netherlands, 1987, pp. 17–33.

[16] 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.

[17] R. D. SKEEL, *Scaling for numerical stability in Gaussian elimination*, J. Assoc. Comput. Mach., 26 (1979), pp. 494–526.

[18] F. J. SMITH, *An algorithm for summing orthogonal polynomial series and their derivatives with applications to curve-fitting and interpolation*, Math. Comp., 19 (1965), pp. 33–36.

[19] G. W. STEWART, *Error analysis of the algorithm for shifting the zeros of a polynomial by synthetic division*, Math. Comp., 25 (1971), pp. 135–139.

[20] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

[21] W. P. TANG AND G. H. GOLUB, *The block decomposition of a Vandermonde matrix and its applications*, BIT, 21 (1981), pp. 505–517.

[22] 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. 11, No. 1, pp. 42–68, January 1990

003

BALANCED APPROXIMATION OF STOCHASTIC SYSTEMS*

K. S. ARUN† AND S. Y. KUNG‡

Abstract. The state of a linear system is an information interface between past inputs and future outputs, and system approximation (even identification) is essentially a problem of approximating a large-dimensional interface by a low-order partial state. Balanced Model Reduction [*IEEE Trans. Automat. Control*, 26 (1981), pp. 17–31], the Fujishige–Nagai–Sawaragi Model Reduction Algorithm [*Internat. J. Control*, 22 (1975), pp. 807–819], and the Principal Hankel Components Algorithm for system identification [Proc. 12th Asilomar Conference on Circuits Systems and Computers, Pacific Grove, CA, November 1978] approximate this input-output interface by its principal components. First generalizations of balanced model reduction to the stochastic system approximation problem are presented. Then the ideas of principal components to the problem of approximating the information interface between two random vectors are generalized; this leads to three approximate stochastic realization methods based on singular value decomposition. These methods and their relationship to the different kinds of balanced stochastic model reduction are discussed.

Key words. balancing, model reduction, stochastic realization, system identification, principal components, singular value decomposition, canonical correlations, mutual information, predictive efficiency

AMS(MOS) subject classifications. 93E12, 62M10, 62H25, 60G25, 93B30

1. Introduction.

This paper addresses the problem of identifying a linear, rational, discrete-time system driven by second-order white noise, given estimates of the covariance lags of the output process. The approach adopted in this paper is that of balanced model reduction. In general, the state of a system is an information interface between the past and the future, and the dimension of this interface is equal to the minimal order of the system and the minimal size of its state vector. However, perturbations in the covariance lags increase the apparent dimension of this interface to much more than the true system order. Then the problem is one of constructing a reduced-order model whose state is an adequate approximation of this apparently large-dimensional interface between the past and the future. This *partial state* must be constructed from the significant components of the information interface. The yardstick that we will use to measure the significance of a candidate state component is the one that is used in balanced model reduction [1], and in the deterministic identification algorithm of [3]. We will show that a partial state chosen using such a criterion, has the highest predictive efficiency for the future.

The key idea is to put the full-order state in internally balanced coordinates, because in such a coordinate system, the elements of the state vector are uncorrelated, and their variances measure their individual contributions to the input-output behavior of the system. Then, the partial state may be constructed from those elements of the full-order state that have the largest variances. In this paper, we will indicate how the variances of the different elements of the balanced full-order state can be determined directly from the covariances via singular value decomposition, without actually constructing the full-order model.

Section 2 develops a stochastic definition of system state, and demonstrates the phase ambiguity in covariance information. Section 3 describes the many kinds of system balancing that have been proposed in the context of stochastic model reduction. Section 4 discusses three approaches to approximate stochastic realization, and brings out their

* Received by the editors August 10, 1987; accepted for publication (in revised form) January 18, 1989.
† Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801 (arun@uicsl.csl.uiuc.edu).
‡ Department of Electrical Engineering, Princeton University, Princeton, New Jersey 08544.

connections to the different balancing schemes discussed in § 3. The theme of § 4 is principal components approximation of the past-future interface, which is also the state space of the innovations representation (minimum-phase model) corresponding to the covariance data.

2. Preliminaries.

2.1. The model. In state-space notation, a discrete-time, linear, shift-invariant, rational, pth order system is

$$\mathbf{x}(t+1)=\mathbf{F}\mathbf{x}(t)+\mathbf{T}v(t), \qquad y(t)=\mathbf{h}\mathbf{x}(t)+v(t) \tag{1}$$

where $v(t)$ and $y(t)$ are the input and output sequences, respectively, $\mathbf{x}(t)$ is a $p \times 1$ state vector process, and $\mathbf{F}$, $\mathbf{T}$, and $\mathbf{h}$ are constant parameter-matrices of sizes $p \times p$, $p \times 1$, and $1 \times p$, respectively. Henceforth, boldface, italic, and upper case Greek letters will be used to denote matrices and vectors, and the transpose operator will be denoted by a superscript t.

In terms of the state-space parameters, the transfer function of the system is given by

$$H(z)=\mathbf{h}(z\mathbf{I}-\mathbf{F})^{-1}\mathbf{T}+1,$$

the poles of the model are the eigenvalues of $\mathbf{F}$, while the zeros are the eigenvalues of the matrix $(\mathbf{F}-\mathbf{Th})$. It can be seen that the impulse response of the model, in terms of the state-space parameters is

$$i(k)=\begin{cases}1, & k=0,\\ \mathbf{h}\mathbf{F}^{k-1}\mathbf{T}, & k>0.\end{cases}$$

For any invertible $p \times p$ matrix $\mathbf{Q}$, the transformed parameter-triple $(\mathbf{Q}^{-1}\mathbf{FQ}, \mathbf{Q}^{-1}\mathbf{T}, \mathbf{hQ})$ has the same impulse response and transfer function; and it corresponds to a new coordinate system for the state. The new state is $\mathbf{Q}^{-1}\mathbf{x}$ instead of $\mathbf{x}$.

The particular state-coordinate system we are interested in, is the so-called internally balanced coordinate system [1]. The internally balanced realization is a special case of the principal-axis realization introduced in [4]. The principal-axis realization is characterized by both the observability grammian $\mathbf{W}$ and the controllability grammian $\mathbf{K}$ being diagonal. In general, these grammians are the solutions of the two following $p \times p$ Lyapunov equations [5]:

$$\mathbf{K}=\mathbf{FKF}^t+\mathbf{TT}^t, \qquad \mathbf{W}=\mathbf{F}^t\mathbf{WF}+\mathbf{h}^t\mathbf{h},$$

and are also explicitly given by

$$\mathbf{K}=[\mathbf{T}\ \mathbf{FT}\ \mathbf{F}^2\mathbf{T}\ \mathbf{F}^3\mathbf{T}\ \cdots]\begin{bmatrix}\mathbf{T}^t\\ \mathbf{T}^t\mathbf{F}^t\\ \mathbf{T}^t\mathbf{F}^{t^2}\\ \mathbf{T}^t\mathbf{F}^{t^3}\\ \vdots\end{bmatrix}=\boldsymbol{CC}^t,$$

$$\mathbf{W}=[\mathbf{h}^t\ \mathbf{F}^t\mathbf{h}^t\ \mathbf{F}^{t^2}\mathbf{h}^t\ \mathbf{F}^{t^3}\mathbf{h}^t\ \cdots]\begin{bmatrix}\mathbf{h}\\ \mathbf{hF}\\ \mathbf{hF}^2\\ \mathbf{hF}^3\\ \vdots\end{bmatrix}=\boldsymbol{O}^t\boldsymbol{O}.$$

In linear systems terminology, the infinite-sized, p-dimensional operators $\boldsymbol{O}$ and $\boldsymbol{C}$ are known as the extended observability matrix and the extended controllability matrix, respectively. Note that these matrices and the two grammians are not unique for a given transfer-function, instead they change with the state coordinates. A transformation of the state from $\mathbf{x}$ to $\mathbf{Q}^{-1}\mathbf{x}$ changes the extended observability matrix to $\boldsymbol{O}\mathbf{Q}$ and the extended controllability matrix to $\mathbf{Q}^{-1}\boldsymbol{C}$, while changing the grammians to $\mathbf{Q}^{-1}\mathbf{K}\mathbf{Q}^{t^{-1}}$ and $\mathbf{Q}^{t}\mathbf{W}\mathbf{Q}$. A transformation $\mathbf{Q}$ that simultaneously diagonalizes both the grammians can always be found, and a principal-axis realization always exists [4]. In fact, for any given transfer-function, many such principal-axis realizations exist, and the balanced realization is one of them.

A realization is said to be internally balanced [1] if the grammians $\mathbf{K}$ and $\mathbf{W}$ are not only diagonal, but also equal to each other:

$$\mathbf{K} = \mathbf{W} = \Sigma, \quad \text{where } \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \cdots, \sigma_p).$$

Though the operators $\boldsymbol{O}$ and $\boldsymbol{C}$ and the corresponding grammians $\mathbf{W}$ and $\mathbf{K}$ depend on the coordinates of the state, the eigenvalues of the product $\mathbf{WK}$ are coordinate-invariant, and in fact, are equal to $\sigma_1^2, \sigma_2^2, \cdots, \sigma_p^2$. Therefore, the elements of the balanced grammians are invariant parameters of the system, and a model-reduction criterion based on these elements will depend on the system's input-output behavior, and not on the state-coordinates. In balanced model reduction [1], [6], the full-order system is first balanced, and then the partial state for the reduced-order system is constructed from the elements of the balanced full-order state with the largest $\sigma_k - s$.

2.2. The notion of state. Intuitively, the state of a (minimal-order) system is a summary of the information in the past input history that is both necessary and sufficient to predict the future output. In fact, from the state-transition equation of the model:

$$\mathbf{x}(t+1) = \mathbf{F}\mathbf{x}(t) + \mathbf{T}v(t),$$

we can see that the state is a linear combination of the past inputs:

$$\mathbf{x}(t) = \mathbf{T}v(t-1) + \mathbf{F}\mathbf{T}v(t-2) + \mathbf{F}^2\mathbf{T}v(t-3) + \mathbf{F}^3\mathbf{T}v(t-4) + \cdots = \boldsymbol{C}\mathbf{V}^{-}(t)$$

where $\boldsymbol{C}$ is the extended controllability matrix defined earlier, and $\mathbf{V}^{-}$ is the following vector of past inputs:

$$\mathbf{V}^{-}(t) = [v(t-1)\ v(t-2)\ v(t-3)\ v(t-4)\ \cdots]^t.$$

Moreover, from the output equation:

$$y(t) = \mathbf{h}\mathbf{x}(t) + v(t),$$

we can see that if the future input is zero, i.e., if $v(k) = 0$ for all $k \geqq t$, then the present and future outputs are completely determined by the present state as $y(t+k) = \mathbf{h}\mathbf{F}^k\mathbf{x}(t)$ for all $k \geqq 0$, or

$$\mathbf{Y}^{+}(t) = \boldsymbol{O}\mathbf{x}(t)$$

where $\boldsymbol{O}$ is the extended observability matrix defined earlier, and $\mathbf{Y}^{+}$ is the following vector of present and future outputs.[1]

$$\mathbf{Y}^{+}(t) = [y(t)\ y(t+1)\ y(t+2)\ y(t+3)\ \cdots]^t.$$

Hence, the extended controllability matrix $\boldsymbol{C}$ maps the past input $\mathbf{V}^{-}$ into the state $\mathbf{x}$, and the extended observability matrix $\boldsymbol{O}$ maps the state vector into the future output

[1] The future-input vector $\mathbf{V}^{+}$ and past-output $\mathbf{Y}^{-}$ are defined just as are $\mathbf{Y}^{+}$ and $\mathbf{V}^{-}$, respectively.

$\mathbf{Y}^+$. Together, the composition $\boldsymbol{H} = \boldsymbol{OC}$ is an operator from the past input to the future output. The same conclusion may be arrived at by noting that the (m, n)th element of $\boldsymbol{H} = \boldsymbol{OC}$ is $\mathbf{h}\mathbf{F}^{m-1}\mathbf{F}^{n-1}\mathbf{T} = i(m+n-1)$ so that the composition $\boldsymbol{H}$ is the Hankel matrix that appears in the following equation:

$$\begin{bmatrix} y(t) \\ y(t+1) \\ y(t+2) \\ y(t+3) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} i(1) & i(2) & i(3) \cdots \\ i(2) & i(3) & i(4) \cdots \\ i(3) & i(4) & i(5) \cdots \\ i(4) & i(5) & i(6) \cdots \\ \cdot & \cdot & \cdot \cdots \\ \cdot & \cdot & \cdot \cdots \\ \cdot & \cdot & \cdot \cdots \end{bmatrix} \begin{bmatrix} v(t-1) \\ v(t-2) \\ v(t-3) \\ v(t-4) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

$$+ \begin{bmatrix} i(0) & 0 & 0 \cdots \\ i(1) & i(0) & 0 \cdots \\ i(2) & i(1) & i(0) \cdots \\ i(3) & i(2) & i(1) \cdots \\ \cdot & \cdot & \cdot \cdots \\ \cdot & \cdot & \cdot \cdots \\ \cdot & \cdot & \cdot \cdots \end{bmatrix} \begin{bmatrix} v(t) \\ v(t+1) \\ v(t+2) \\ v(t+3) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

or

$$\mathbf{Y}^+ = \boldsymbol{H}\mathbf{V}^- + \mathbf{L}\mathbf{V}^+. \tag{2}$$

Knowing that the Hankel matrix in (2) can be factored as $\boldsymbol{H} = \boldsymbol{OC}$, (2) can be rewritten as

$$\mathbf{Y}^+ = \boldsymbol{O}\mathbf{x} + \mathbf{L}\mathbf{V}^+ \quad \text{where } \mathbf{x} = \boldsymbol{C}\mathbf{V}^-.$$

Hence, $\boldsymbol{H}$ is a two-stage operator that maps the past input $\mathbf{V}^-$ into the state $\mathbf{x}$, and the state $\mathbf{x}$ into the future $\mathbf{Y}^+$. Consequently, the rank of $\boldsymbol{H}$ is equal to the size of the state vector that in turn, is equal to the model order p.

Let the singular value decomposition (SVD) of $\boldsymbol{H}$ be

$$\boldsymbol{H} = U\Sigma V^t.$$

One choice of factors $\boldsymbol{O}$ and $\boldsymbol{C}$ is $\mathbf{U}\Sigma^{1/2}$ and $\Sigma^{1/2}\mathbf{V}^t$, respectively. Any other choice corresponds to different coordinates for the system state. In the chosen coordinates, both grammians $\mathbf{W}$ and $\mathbf{K}$ are equal to Σ. This establishes that the singular values of $\boldsymbol{H}$ are in fact, equal to the system's coordinate-invariant parameters $\sigma_1, \cdots, \sigma_p$. Hence, the deterministic identification algorithm of [3] that constructs a low-order system from the principal components in the SVD of $\boldsymbol{H}$, uses in effect, the same partial-state selection criterion used in balanced model reduction.

Because of the system's time-invariance, $\boldsymbol{O}$ and $\boldsymbol{C}$ (in any coordinates) have special structure, they satisfy

$$\boldsymbol{O}\mathbf{F} = \boldsymbol{O}\uparrow \quad \text{and} \quad \boldsymbol{C}^t\mathbf{F}^t = \boldsymbol{C}^t\uparrow \tag{3}$$

where $\boldsymbol{O}\uparrow$ is obtained from $\boldsymbol{O}$ simply by deleting the first row, and shifting all other rows one step up.

2.3. The stochastic model. When the input $v(t)$ to the model of (1) is a white random process of variance ρ, the system-invariants σ_k take on a new interpretation. Here, by white, we only mean second-order white, i.e., a sequence of zero-mean, uncor-

related random variables, all with the same variance:

$$\mathrm{E}[v(t)v(t+m)] = \begin{cases} \rho, & m=0, \\ 0, & m\neq 0 \end{cases}$$

where $\mathrm{E}[\cdot]$ denotes the expectation operator. Then the state covariance matrix $\mathbf{P} = \mathrm{E}[\mathbf{x}\mathbf{x}^t]$ satisfies the Lyapunov equation:

$$\mathbf{P} = \mathbf{F}\mathbf{P}\mathbf{F}^t + \rho\mathbf{T}\mathbf{T}^t,$$

and is equal to ρ times the controllability grammian $\mathbf{K}$. The covariance of the output process $y(t)$ is

$$r(m) = \mathrm{E}[y(t)y(t+m)] = \begin{cases} \mathbf{h}\mathbf{P}\mathbf{h}^t + \rho, & m=0, \\ \mathbf{h}\mathbf{F}^{|m|-1}\mathbf{g}, & m\neq 0 \end{cases}$$

where $\mathbf{g} = \mathbf{F}\mathbf{P}\mathbf{h}^t + \rho\mathbf{T} = \mathrm{E}[\mathbf{x}(t+1)y(t)]$. The output power spectrum

$$S(z) = \rho H(z)H(z^{-1}) = \textstyle\sum_{-\infty}^{+\infty} r(m)z^{-m}$$

is given in terms of the system parameters as:

$$S(z) = R(z) + R(z^{-1}) \quad \text{where } R(z) = \mathbf{h}(z\mathbf{I}-\mathbf{F})^{-1}\mathbf{g} + \frac{r(0)}{2}.$$

Since the state-variance $\mathbf{P}$ is ρ times the controllability grammian $\mathbf{K}$, the variance of state-element x_k in internally balanced coordinates is simply $\rho\sigma_k$ because

$$\mathbf{P} = \rho\mathbf{W} = \rho\Sigma;$$

for an internally balanced realization. However, because of phase ambiguity in stochastic systems when only output covariances $\{r(m)\}$ are known, different kinds of balancing have been proposed in the context of the stochastic system approximation problem.

2.4. Phase ambiguity. If we reflect some (or all) system-zeros across the unit circle in the z-plane and rescale the transfer function to make the direct feedthrough term $i(0)$ equal to 1, we get a new system that can still generate the process $y(t)$ when it is driven by a *different* white noise sequence.[2] Thus, when we wish to identify the system from the output process alone (without knowledge of the input process) or from output covariances, there is ambiguity as to the exact location of the system-zeros. The restriction that the system be causally stable constrains all poles to be within the unit circle, but because of the ambiguity about zero locations, there are many causally stable models that have the same poles, and generate the same output covariance sequence $r(m)$ when driven by different white-noise processes.

However, the triple $(\mathbf{F}, \mathbf{g}, \mathbf{h})$ can be determined uniquely (modulo coordinate transformations to $(\mathbf{Q}^{-1}\mathbf{F}\mathbf{Q}, \mathbf{Q}^{-1}\mathbf{g}, \mathbf{h}\mathbf{Q})$) from the output covariance sequence. This means that the various models that generate the same covariance sequence, can each be put in state-coordinates where they all have a common state-feedback matrix $\mathbf{F}$, the same output matrix $\mathbf{h}$, and the same vector $\mathbf{g} = \mathrm{E}[\mathbf{x}(t+1)y(t)]$. However, they differ in the input matrix $\mathbf{T}$, input variance ρ, and state-variance $\mathbf{P}$. We thus have a number of *covariance-equivalent* models of the form:

$$\mathbf{x}_m(t+1) = \mathbf{F}\mathbf{x}_m(t) + \mathbf{T}_m v_m(t), \qquad y(t) = \mathbf{h}\mathbf{x}_m(t) + v_m(t),$$

[2] The two white-noise processes are related by an all-pass system of order p.

all of which have unity feedthrough, and the same $\mathbf{F}$ and $\mathbf{h}$ matrices, but have differing $\mathbf{T}_m$ matrices, and are driven by different white input sequences $v_m(t)$ with different variances ρ_m. Yet they all generate the same output process $y(t)$. Their states and state-variances $\mathbf{P}_m = \mathrm{E}[\mathbf{x}_m \mathbf{x}_m^t]$ are different, but every $\mathbf{P}_m$ satisfies the algebraic Riccati equation:

$$\mathbf{P}_m = \mathbf{F}\mathbf{P}_m\mathbf{F}^t + (\mathbf{g} - \mathbf{F}\mathbf{P}_m\mathbf{h}^t)(r(0) - \mathbf{h}\mathbf{P}_m\mathbf{h}^t)^{-1}(\mathbf{g} - \mathbf{F}\mathbf{P}_m\mathbf{h}^t)^t.$$

This can be verified by replacing $\mathbf{T}_m$ and ρ_m in the Lyapunov equation $\mathbf{P}_m = \mathbf{F}\mathbf{P}_m\mathbf{F}^t + \rho_m\mathbf{T}_m\mathbf{T}_m^t$ by $\mathbf{T}_m = \rho_m^{-1}(\mathbf{g} - \mathbf{F}\mathbf{P}_m\mathbf{h}^t)$ and $\rho_m = r(0) - \mathbf{h}\mathbf{P}_m\mathbf{h}^t$.

In Faurre's pioneering work on stochastic realization [7], [8], he has shown that the state-variance of the minimum-phase model (that has all its zeros inside the unit circle) is the *smallest* solution $\mathbf{P}_{\min}$ of the Riccati equation.[3] It was later established [9]–[11] that the *largest* solution $\mathbf{P}_{\max}$ is the state-variance of the maximum-phase model, having all its zeros outside the unit circle.

2.5. Stochastic definition of state. In this section, we will develop a stochastic definition for the state of a system, along the lines of [12]. For a zero-mean $n \times 1$ random vector $\mathbf{Y} = (y_1, y_2, \cdots, y_n)^t$, Span $(\mathbf{Y})$ will denote the Hilbert space of all random variables that are linear combinations of $\{y_1, y_2, \cdots, y_n\}$. The inner product on this space of zero-mean random variables is the cross correlation, and the dimension of this space (upper bounded by n) is the largest number of mutually uncorrelated random variables in the space. We will use the notation $\mathbf{x}\backslash\mathbf{Y}$ to denote the linear, minimum-variance estimate of zero-mean random vector $\mathbf{x}$ from the zero-mean random vector $\mathbf{Y}$. It is also the orthogonal projection of $\mathbf{x}$ onto the subspace Span $(\mathbf{Y})$. From elementary estimation theory [13], we know that

$$\mathbf{x}\backslash\mathbf{Y} = \mathrm{E}[\mathbf{x}\mathbf{Y}^t](\mathrm{E}[\mathbf{Y}\mathbf{Y}^t])^{-1}\mathbf{x}. \tag{4}$$

When the input is a white-noise process, the past and future inputs are uncorrelated, and as a result, the two components of the future output vector $\mathbf{Y}^+$ from (2)

$$\mathbf{Y}^+ = \boldsymbol{H}\mathbf{V}^- + \mathbf{L}\mathbf{V}^+$$

are orthogonal. Consequently, the orthogonal projection of $\mathbf{Y}^+$ on Span $(\mathbf{V}^-)$ must be $\boldsymbol{H}\mathbf{V}^-$ itself,

$$\text{i.e.,} \quad \mathbf{Y}^+\backslash\mathbf{V}^- = \boldsymbol{H}\mathbf{V}^-.$$

However, we have already seen that this information is completely summarized in the state, since $\boldsymbol{H}\mathbf{V}^- = \boldsymbol{O}\mathbf{x}$, and $\mathbf{x} = \boldsymbol{C}\mathbf{V}^-$. Therefore,

$$\mathbf{Y}^+\backslash\mathbf{V}^- = \boldsymbol{H}\mathbf{V}^- = \boldsymbol{O}\mathbf{x}, \qquad \mathbf{x} = \boldsymbol{C}\mathbf{V}^-, \tag{5}$$

a mathematical restatement of the fact that the state condenses all the information in the past input that is sufficient for predicting the future output.

This input-output notion of state can be further refined to a past-future notion based entirely on the output process. While (5) is satisfied by all the covariance-equivalent models that generate $y(t)$, the following past-to-future definitions of the state will depend on the zero locations of the model.

In each of the covariance-equivalent models $(\mathbf{F}, \mathbf{T}_m, \mathbf{h}, \rho_m)$, the output $y(t)$ is obtained causally from $v_m(t)$, and consequently, $\mathbf{Y}^-$ lies in Span $(\mathbf{V}_m^-)$ for every m. However, only the minimum-phase model has a causally stable inverse, and only $v_{\min}(t)$

[3] A symmetric matrix $\mathbf{A}$ is said to be bigger than another symmetric matrix $\mathbf{B}$ if $\mathbf{A} - \mathbf{B}$ is nonnegative definite.

can be obtained causally from the output $y(t)$.[4] Therefore, $\mathbf{V}^-_{\min}$ lies in Span $(\mathbf{Y}^-)$, but none of the other past inputs $\mathbf{V}^-_m$ lie in Span $(\mathbf{Y}^-)$. As a direct consequence, we have the following equality of spaces:

$$\text{Span}\,(\mathbf{Y}^-) = \text{Span}\,(\mathbf{V}^-_{\min});$$

however, Span $(\mathbf{Y}^-)$ is a *proper* subspace of Span $(\mathbf{V}^-_m)$, the space spanned by the past inputs to every nonminimum phase model. As a result of the above equality, the state of the minimum-phase model can be also interpreted as a summary of the past *output* history (instead of past input history) for predicting the future output.

2.5.1. The minimum-phase model. The minimum-phase model:

$$\mathbf{x}_{\min}(t+1) = \mathbf{F}\mathbf{x}_{\min}(t) + \mathbf{T}_{\min} v_{\min}(t), \qquad y(t) = \mathbf{h}\mathbf{x}_{\min}(t) + v_{\min}(t) \tag{6}$$

has a causally stable inverse obtained by simply rearranging the forward model's equations:

$$\mathbf{x}_{\min}(t+1) = (\mathbf{F} - \mathbf{T}_{\min}\mathbf{h})\mathbf{x}_{\min}(t) + \mathbf{T}_{\min} y(t), \qquad v_{\min}(t) = -\mathbf{h}\mathbf{x}_{\min}(t) + y(t). \tag{7}$$

The minimum-phase property ensures that the zeros of the model of (6) that are the eigenvalues of $(\mathbf{F} - \mathbf{T}_{\min}\mathbf{h})$ lie within the unit circle. But these eigenvalues are precisely the poles of the inverse filter of (7), hence the inverse filter must be stable. Thus the state-process $\mathbf{x}_{\min}(t)$ as well as the input process $v_{\min}(t)$ can be obtained causally from the output $y(t)$ using the above inverse filter.

The state transition equation of the inverse filter indicates that

$$\mathbf{x}_{\min}(t) = (\mathbf{T}_{\min}(\mathbf{F} - \mathbf{T}_{\min}\mathbf{h})\mathbf{T}_{\min}(\mathbf{F} - \mathbf{T}_{\min}\mathbf{h})^2\mathbf{T}_{\min}\cdots)\cdot\begin{bmatrix} y(t) \\ y(t+1) \\ y(t+2) \\ \vdots \end{bmatrix} = \Psi\mathbf{Y}^-(t).$$

Moreover, since Span $(\mathbf{Y}^-)$ = Span $(\mathbf{V}^-_{\min})$, we have

$$\mathbf{Y}^+ \backslash \mathbf{Y}^- = \mathbf{Y}^+ \backslash \mathbf{V}^-_{\min} = \boldsymbol{H}\mathbf{V}^-_{\min} = \boldsymbol{O}\mathbf{x}_{\min}.$$

Combining the last two equations, we get

$$\mathbf{x}_{\min} = \Psi\mathbf{Y}^- \quad \text{and} \quad \mathbf{Y}^+ \backslash \mathbf{Y}^- = \boldsymbol{O}\mathbf{x}_{\min}, \tag{8}$$

meaning that the state of the minimum-phase model summarizes the past *output* history for predicting the future output.

As a footnote, (8) indicates that the projection of $y(t)$ on the past space Span $(\mathbf{Y}^-)$ is nothing but

$$y(t) \backslash \mathbf{Y}^- = \mathbf{h}\mathbf{x}_{\min}(t).$$

Therefore, the part of $y(t)$ that cannot be predicted from the past $\mathbf{Y}^-$ is simply

$$y(t) - \mathbf{h}\mathbf{x}_{\min}(t) = v_{\min}(t).$$

Thus, the input white noise to a minimum-phase model is the innovations process [14] for the output, and consequently, the minimum-phase model is also called the *innovations representation* of the output process [9], [15].

3. Balancing of stochastic systems.

3.1. Balanced model reduction. If the full-order model is completely given, and the question is one of model reduction, we could apply Moore's balanced model reduction

[4] The subscript min on $\mathbf{x}$, $\mathbf{T}$, and v indicate that they refer to the minimum-phase model only.

procedure in a fairly straightforward fashion. As noted earlier, in internally balanced coordinates, $\mathbf{P} = \rho\mathbf{W}$ = diagonal matrix $\rho\Sigma$, whose (k, k) entry $\rho\sigma_k$ is the variance of the element x_k of the balanced full-order state. The partial state for the reduced-order model may be constructed from the elements x_k of the balanced full-order state with the largest variances.

When the full-order model is internally balanced, the different elements x_k of the full-order state are uncorrelated (since $\mathbf{P}$ is diagonal), and their contributions to energy in the future-output, are also decoupled (since $\mathbf{W}$ is also diagonal). In addition, for each state-element x_k, its variance P_{kk} is proportional to W_{kk} that measures its output-energy contribution. Therefore, the significance of x_k can be measured by its variance P_{kk} alone. Hence, balanced model reduction picks out those components of the state space, that have large variances and also make a large contribution to the future output.

It turns out that such a model reduction scheme was in effect, proposed by Fujishige, Nagai, and Sawaragi [2] much before the concept of balancing was introduced. Fujishige, Nagai, and Sawaragi used a least-squares prediction-error criterion to justify their model reduction algorithm.

3.2. Fujishige model reduction. Recall the input-output definition of state in (5):

$$\mathbf{Y}^+ \backslash \mathbf{V}^- = \boldsymbol{H}\mathbf{V}^- = \boldsymbol{O}\mathbf{x}, \qquad \mathbf{x} = \boldsymbol{C}\mathbf{V}^-.$$

The space Span $(\boldsymbol{O}\mathbf{x})$ is coordinate-invariant, and its dimension n is the order of the model. The full-order state is any basis for this space. For model reduction, the partial state has to be obtained from the significant components of this full-order state space. An optimal compression of $\boldsymbol{O}\mathbf{x}$ that retains the maximum information is provided by the principal components in its Karhunen–Loeve (KL) decomposition. Let the eigen-decomposition of its covariance matrix $\mathrm{E}[(\boldsymbol{O}\mathbf{x})(\boldsymbol{O}\mathbf{x})^t] = \boldsymbol{O}\mathbf{P}\boldsymbol{O}^t$ be

$$\boldsymbol{O}\mathbf{P}\boldsymbol{O}^t = \mathbf{U}\Sigma^2\mathbf{U}^t = \sum_{k=1}^{n} \sigma_k^2 \mathbf{u}_k \mathbf{u}_k^t$$

where n is the model order, and the eigenvalues are arranged in nonincreasing order $\sigma_1 \geqq \sigma_2 \geqq \cdots \geqq \sigma_n$. Then, a KL decomposition [16] of $\boldsymbol{O}\mathbf{x}$ is

$$\boldsymbol{O}\mathbf{x} = \sum_{k=1}^{n} (\mathbf{u}_k^t \boldsymbol{O}\mathbf{x})\mathbf{u}_k,$$

and the random variables $\mathbf{u}_k^t \boldsymbol{O}\mathbf{x}$ that are the scalar coefficients in the above expansion, are uncorrelated with each other, while their individual variances are σ_k^2. Let $\mathbf{U}_1$ be composed from the p eigenvectors corresponding to the p largest eigenvalues. Then, a principal components approximation of $\boldsymbol{O}\mathbf{x}$ is (for $p < n$) [17]

$$\sum_{k=1}^{p} (\mathbf{u}_k^t \boldsymbol{O}\mathbf{x})\mathbf{u}_k$$

that is completely summarized in

$$\mathbf{x}_{\text{partial}} = \mathbf{U}_1^t \boldsymbol{O}\mathbf{x}.$$

The Fujishige Model Reduction Algorithm chooses the above partial state because it has the smallest error in predicting the future output. Among all p-sized vectors from Span $(\boldsymbol{O}\mathbf{x})$, the above choice minimizes the least-squares prediction error $\mathrm{E}\|\mathbf{Y}^+ - \mathbf{Y}^+ \backslash \mathbf{x}_{\text{partial}}\|^2$ [18], [19].

3.2.1. Relation to balanced model reduction. Recall that $\boldsymbol{O}\mathbf{x}$ equals $\boldsymbol{H}\mathbf{V}^{-}$. Taking the covariances of both sides, we arrive at

$$\boldsymbol{O}\boldsymbol{P}\boldsymbol{O}^{t} = \rho \boldsymbol{H}\boldsymbol{H}^{t},$$

implying that the eigenvalues of $\boldsymbol{O}\boldsymbol{P}\boldsymbol{O}^{t}$ are ρ times the squared singular values of the infinite Hankel $\boldsymbol{H}$. The singular values in turn, are equal to the system-invariant parameters used in balanced model reduction. Thus, Fujishige's method uses the same partial state selection criterion as balanced model reduction. Furthermore, it can be verified that both methods obtain the same reduced-order model. In effect, balanced model reduction was first proposed by Fujishige, Nagai, and Sawaragi using a *stochastic* justification.

When only output covariances are available, and the full-order model is not given, the situation is very different. Because of phase-ambiguity, there is a whole class of full-order models that could have generated the given covariance sequence, and a balanced model reduction of each of them will lead to not only different phase-responses, but also to different approximated covariance sequences, and different power spectra. The first two kinds of balancing that are described below, work with system-invariant parameters that are common to all the full-order covariance-equivalent models that generate $r(m)$. Neither of the two approaches internally balances any of the covariance-equivalent models in Moore's sense.

3.3. Covariance balancing. Although the states of the covariance-equivalent models

$$\mathbf{x}_m(t+1) = \mathbf{F}\mathbf{x}_m(t) + \mathbf{T}_m v_m(t), \qquad y(t) = \mathbf{h}\mathbf{x}_m(t) + v_m(t)$$

that generate the process $y(t)$ are different, they all have the same correlation with the past output, i.e.,

$$\boldsymbol{G} = \mathrm{E}[\mathbf{x}_m \mathbf{Y}^{-t}]$$

is the same for all covariance-equivalent models, and is in fact,

$$\boldsymbol{G} = [\mathbf{g}, \mathbf{F}\mathbf{g}, \mathbf{F}^2\mathbf{g}, \mathbf{F}^3\mathbf{g}, \cdots]$$

where $\mathbf{g} = \mathbf{F}\mathbf{P}_m\mathbf{h}^t + \rho_m \mathbf{T}_m$ is the same for all models.[5]

The new grammian $\mathbf{J} = \boldsymbol{G}\boldsymbol{G}^t$ that satisfies the Lyapunov equation

$$\mathbf{J} = \mathbf{F}\mathbf{J}\mathbf{F}^t + \mathbf{g}\mathbf{g}^t$$

and is common to all the covariance-equivalent models, is the controllability grammian for the new causal system $(\mathbf{F}, \mathbf{g}, \mathbf{h})$ whose impulse response is $\{r(0)/2, r(1), r(2), r(3), \cdots\}$, and transfer function is $R(z)$. Just as for the other grammians, coordinate transformations effect $\mathbf{J}$, however, the eigenvalues of the product $\mathbf{W}\mathbf{J}$ are coordinate-invariant, and common to the entire class of covariance-equivalent models $(\mathbf{F}, \mathbf{T}_m, \mathbf{h}, \rho_m)$.

We will say that this class of models is *covariance-balanced* if

$$\mathbf{W} = \mathbf{J} = \text{a diagonal matrix } \mathbf{D}.$$

Taking a hint from balanced model reduction, we could try to construct a class of reduced-order approximate models by retaining only those rows and columns of the covariance-balanced $(\mathbf{F}, \mathbf{g}, \mathbf{h})$ matrices corresponding to the p largest entries in this diagonal matrix

[5] This indicates that $\mathbf{x}_m \backslash \mathbf{Y}^{-} = \mathbf{x}_{\min}$ for all m, explaining why $\mathbf{P}_{\min} \leqq \mathbf{P}_m$ for all m.

D. However, there is no guarantee that the approximated pseudocovariance sequence

$$\hat{r}(m) = \begin{cases} r(0), & m = 0, \\ \hat{\mathbf{h}}\hat{\mathbf{F}}^{|m|-1}\hat{\mathbf{g}}, & m \neq 0 \end{cases}$$

is nonnegative definite. As a result, the reduced-order Riccati equation

$$\mathbf{P} = \hat{\mathbf{F}}\mathbf{P}\hat{\mathbf{F}}^t + (\hat{\mathbf{g}} - \hat{\mathbf{F}}\mathbf{P}\hat{\mathbf{h}}^t)(r(0) - \hat{\mathbf{h}}\mathbf{P}\hat{\mathbf{h}})^{-1}(\hat{\mathbf{g}} - \hat{\mathbf{F}}\mathbf{P}\hat{\mathbf{h}}^t)^t$$

may not have *any* positive-definite solution. In other words, we may not be able to find *any* model that generates the pseudo-covariance sequence $\hat{r}(m)$. A simple, ad hoc solution is to add a suitable constant to $r(0)$.

3.4. Desai and Pal stochastic balancing. The state-variances of the minimum-phase and maximum-phase models ($\mathbf{P}_{\min}$ and $\mathbf{P}_{\max}$, respectively) change with coordinate transformations. If $\mathbf{x}_{\min}$ is transformed to $\mathbf{Q}^{-1}\mathbf{x}_{\min}$, then $\mathbf{P}_{\min}$ and $\mathbf{P}_{\max}$ get transformed to $\mathbf{Q}^{-1}\mathbf{P}_{\min}\mathbf{Q}^{-1'}$ and $\mathbf{Q}^{-1}\mathbf{x}_{\max}\mathbf{Q}^{-1'}$. However, the product $\mathbf{P}_{\max}^{-1}\mathbf{P}_{\min}$ undergoes a *similarity* transformation to $\mathbf{Q}'\mathbf{P}_{\max}^{-1}\mathbf{P}_{\min}\mathbf{Q}^{-1'}$. Therefore, the eigenvalues of $\mathbf{P}_{\max}^{-1}\mathbf{P}_{\min}$ are also system-invariants [20], [21], and like the eigenvalues of $\mathbf{WJ}$, are common to the entire class of covariance-equivalent models ($\mathbf{F}$, $\mathbf{T}_m$, $\mathbf{h}$, ρ_m). This class can be coordinate-transformed to make

$$\mathbf{P}_{\max}^{-1} = \mathbf{P}_{\min} = \text{a diagonal matrix } \Lambda.$$

In these coordinates, the class of covariance-equivalent models ($\mathbf{F}$, $\mathbf{T}_m$, $\mathbf{h}$, ρ_m) is said to be *stochastically balanced in the Desai and Pal sense* [21], [22]. The entries of the diagonal matrix Λ are the eigenvalues of $\mathbf{P}_{\max}^{-1}\mathbf{P}_{\min}$ in any coordinate system, and are coordinate-invariant parameters of the covariance-equivalent class.

Desai and Pal suggest constructing a class of reduced-order covariance-equivalent models by retaining only those elements of the stochastically balanced full-order states $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ that correspond to the p largest entries in the diagonal matrix Λ. The justification for such a model reduction is in the fact that the entries of Λ are the canonical correlation coefficients between $\mathbf{Y}^-$ and $\mathbf{Y}^+$, and measure the mutual information between them. A more detailed discussion follows in the next section on approximate stochastic realization.

3.5. Internal balancing of the minimum-phase model. In both the kinds of balancing described above, none of the models ($\mathbf{F}$, $\mathbf{T}_m$, $\mathbf{h}$, ρ_m) is internally balanced, and none of the states $\mathbf{x}_m$ is centered between the input and the output. The problem here is that if we internally balance any one of the covariance-equivalent models, all the others will be *unbalanced*. Since we do not know a priori which model is the correct one, the two above kinds of balancing do not internally balance any of the models. Instead, they work with system-invariants that are common to all of them.

In many applications, the model's phase-response is of no concern, and the only purpose of covariance approximation is to smooth out the perturbations in the covariance estimates, or to obtain a low-order rational spectral estimate. There are other applications where the sole purpose of stochastic modeling is least-squares extrapolation/prediction of $y(t)$ [23]. In general, when only the output process (or its covariances) are known, we cannot hope to approximate the input-output transfer of the underlying system, without any additional knowledge about the system's phase-response or of the input process itself. However, a balanced model reduction of the full-order *minimum-phase* system will ap-

proximate the past-future interface in the output process optimally in an unweighted least-squares sense.

We will show in the next section that an internally balanced approximation of the minimum-phase model provides good covariance approximation, and minimizes the least-squares error in the extrapolation/prediction of $y(t)$. Recall that the minimum-phase model is internally balanced in Moore's sense, if

$$\mathbf{P}_{\min} = \rho_{\min}\, \mathbf{W} = \text{a diagonal matrix.}$$

In these coordinates, the elements of $\mathbf{x}_{\min}$ are uncorrelated, and their variances are $\rho_{\min}\sigma_k$, where $\sigma_k - s$ are Moore's system-invariant parameters. Balanced model reduction of the minimum-phase model (innovations representation) corresponding to the covariances is achieved by retaining only the p largest $\sigma_k - s$.

4. Approximate stochastic realization. The problem addressed in this section is that of approximating a perturbed covariance sequence (that may not be nonnegative definite after the perturbation) by a low-order rational model. Because the perturbed sequence may not be a valid covariance sequence, we cannot hope to first construct a full-order model, and then reduce its order by one of the three balanced approximations discussed above. We will construct reduced-order approximate models directly from the perturbed covariance sequence. In this section, we will indicate how stochastic system approximation based on the three kinds of balancing can be performed directly from the covariance sequence, without constructing a full-order model, via the SVD of certain matrices.

We will now formulate the approximate modeling problem as one of approximating the apparently high-dimensional information interface between $\mathbf{Y}^-$ and $\mathbf{Y}^+$ by a p-dimensional state $\mathbf{x}_{\text{partial}}$.

4.1. Partial state selection. Using (4), we can verify that

$$\mathbf{Y}^+ \backslash \mathbf{Y}^- = \mathbf{H}\mathbf{R}^{-1}\mathbf{Y}^-$$

where $\mathbf{R} = \mathrm{E}(\mathbf{Y}^-\mathbf{Y}^{-t})$ and $\mathbf{H} = \mathrm{E}(\mathbf{Y}^+\mathbf{Y}^{-t})$ are the Toeplitz and Hankel matrices, respectively, formed from the covariance lags of the output process. Combining this equation with (8):

$$\mathbf{x}_{\min} = \Psi\mathbf{Y}^- \quad \text{and} \quad \mathbf{Y}^+ \backslash \mathbf{Y}^- = \mathcal{O}\mathbf{x}_{\min},$$

leads to the following observations.

(1) $\mathbf{H}\mathbf{R}^{-1}$ equals $\mathcal{O}\Psi$. Consequently, $\mathbf{H}\mathbf{R}^{-1}$ must have rank equal to the size of the state vector (i.e., equal to the model order p).

(2) Moreover,

$$\mathbf{H}\mathbf{R}^{-1}\mathbf{Y}^- = \mathcal{O}\mathbf{x}_{\min},$$

which means that the dimension of Span $(\mathbf{H}\mathbf{R}^{-1}\mathbf{Y}^-)$ is equal to the model order p, and that the state $\mathbf{x}_{\min}$ is any basis for this space.

Thus, the stochastic realization problem is, simply stated, the problem of picking a basis for Span $(\mathbf{H}\mathbf{R}^{-1}\mathbf{Y}^-)$ [24].

However, when the covariance lags are estimated from a finite record of the stochastic process or are directly measured, then the perturbations in the lags will distort the rank structure of $\mathbf{H}\mathbf{R}^{-1}$. It will have full rank, making the apparent state size much larger than the true model order. Then the problem is one of constructing a partial state from those components in Span $(\mathbf{Y}^-)$ that contain the most information regarding $\mathbf{Y}^+$. This partial-state must "effectively" summarize the information interface between $\mathbf{Y}^+$ and $\mathbf{Y}^-$. Note

that the problem is one of compressing $\mathbf{Y}^-$ while retaining maximal information not about $\mathbf{Y}^-$, but about $\mathbf{Y}^+$. Hence, principal components analysis of $\mathbf{Y}^-$ will not suffice [16], [23], for the partial-state selection problem.[6] The compression of $\mathbf{Y}^-$ into its principal components is not appropriate because it is based on the selection of components containing the maximum information about $\mathbf{Y}^-$ itself, whereas only specific information about $\mathbf{Y}^+$ is of interest in the partial-state selection problem.

However, there exist in the statistical literature, generalizations of the concept of principal components (of a random vector) to the problem of compressing the information interface between two random vectors (that will henceforth be referred to as the 2-vector problem for the sake of brevity). We will present three approaches to approximate stochastic realization as applications of three such generalizations.

For a zero-mean $n \times 1$ random vector $\mathbf{Y}$, the p principal components of $\mathbf{Y}$

(a) Are maximally correlated with $\mathbf{Y}$,

(b) Have maximum self-information in the Gaussian case,

(c) And retain the maximum reconstruction (prediction) efficiency for $\mathbf{Y}$.

Generalizing these three properties to the 2-vector problem leads to the three methods of this section.

4.2. The principal components of H. Taking a hint from the correlation-maximizing property of the principal components of a single random vector, we could look for a partial-state in Span $(\mathbf{Y}^-)$ that maximizes some measure of its correlation with $\mathbf{Y}^+$. For instance, we could pick

$$(9) \qquad p \times 1 \text{ sized } \mathbf{x}_{\text{partial}} = \Psi \mathbf{Y}^- \text{ to } \underset{\text{constraint:}\Psi\Psi^t = \mathbf{I}_p}{\text{maximize}} \|\mathbf{E}[\mathbf{Y}^+ \mathbf{x}_{\text{partial}}^t]\|_F,$$

where subscript F denotes the Frobenius norm of the matrix. The solution to this is constructed from the principal components of the covariance Hankel matrix $\mathbf{H}$. The rows of Ψ must be the orthonormal singular vectors of $\mathbf{H}$ corresponding to the p largest singular values. If the SVD of $\mathbf{H}$ is

$$\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^t = \mathbf{U}_1\mathbf{D}_1\mathbf{V}_1^t + \mathbf{U}_2\mathbf{D}_2\mathbf{V}_2^t$$

(where the subscript 1 stands for the dominant components corresponding to the p largest singular values) then the solution to the minimization problem of (9) is $\Psi = \mathbf{V}_1^t$. This justifies the principal components approximation of $\mathbf{H}$ [25]–[27], that has been extensively used for approximate stochastic modeling [28], [29]. We will henceforth refer to this approximation as the PC-H approximation.

4.2.1. Relation to covariance balancing. It can be easily verified that the eigenvalues of the full-order $\mathbf{WJ}$ in any coordinate system, are identically the squares of the singular values of the infinite Hankel $\mathbf{H}$. First note that $\mathbf{H}$ factors into the product of $\boldsymbol{O}$ and $\boldsymbol{G}$:

$$\mathbf{H} = \mathbf{E}[\mathbf{Y}^+\mathbf{Y}^{-t}] = \mathbf{E}[\boldsymbol{O}\mathbf{x}_m\mathbf{Y}^{-t}] = \boldsymbol{O}\mathbf{E}[\mathbf{x}_m\mathbf{Y}^{-t}] = \boldsymbol{O}\boldsymbol{G}$$

since $(\mathbf{Y}^+ - \boldsymbol{O}\mathbf{x}_m)$ depends only on the future input $\mathbf{V}_m^+$ that is uncorrelated with the past output $\mathbf{Y}^-$. Hence, the rank of $\mathbf{H}$ is equal to the full order of the model, and one choice of $\boldsymbol{O}$ and $\boldsymbol{G}$ is $\mathbf{U}\mathbf{D}^{1/2}$ and $\mathbf{D}^{1/2}\mathbf{V}^t$. In these coordinates, $\mathbf{W} = \mathbf{J} = \mathbf{D}$, and so the eigenvalues of $\mathbf{WJ}$ (that are coordinate-invariant parameters of the covariance-equivalent class) are squares of the singular values of $\mathbf{H}$. Thus, the same partial state selection criterion is used in covariance-balanced model reduction and in the PC-H method. Hence

[6] Note that the covariance matrix $\mathbf{R}$ is not expected to have rank equal to the model order, even when the lags are exact. Hence, in the perturbed situation, a principal components approximation of $\mathbf{R}$ is not justified.

the PC-H method suffers from the same problem: we can obtain $(\hat{\mathbf{F}}, \hat{\mathbf{g}}, \hat{\mathbf{h}})$ for the class of reduced-order models, but the class may be empty because the pseudo-spectrum $\mathrm{Re}\,[2\hat{\mathbf{h}}(e^{j\omega}\mathbf{I} - \hat{\mathbf{F}})^{-1}\hat{\mathbf{g}} + r(0)]$ may be negative at some frequencies ω. However, for spectral estimation applications, where we are only interested in locating the frequencies of spectral peaks [29], [30], the lack of positivity may not be a serious problem. For more detailed discussion of the quality of the PC-H approximation, the reader is referred to [31].

4.3. The canonical correlations criterion. This criterion was first proposed in statistics by Hotelling [32], and later used for the partial-state selection problem by Akaike [12], [24]. Here, any orthonormal basis $\mathbf{Z}^+$ is found for Span $(\mathbf{Y}^+)$, and the p partial-state components are selected as p orthonormal random variables from Span $(\mathbf{Y}^-)$ that have the maximum correlation with $\mathbf{Z}^+$. The constraint that the partial-state components be orthonormal translates into the constraint

$$\mathrm{E}[\mathbf{x}_{\text{partial}}\,\mathbf{x}_{\text{partial}}^t] = \Psi\mathbf{R}\Psi^t = \mathbf{I}_p.$$

If $\mathbf{R}^{1/2}$ is any square root of $\mathbf{R}$ (i.e., $\mathbf{R} = \mathbf{R}^{1/2}\mathbf{R}^{1/2t}$), and $\mathbf{R}^{-1/2}$ is its inverse, then one choice for $\mathbf{Z}^+$ is $\mathbf{R}^{-1/2}\mathbf{Y}^+$, and so our problem is to

$$\underset{\text{constraint:}\Psi R\Psi^t = I_p}{\text{Maximize}} \|\mathrm{E}[\mathbf{R}^{-1/2}\mathbf{Y}^+\mathbf{x}_{\text{partial}}^t]\|_F = \|\mathbf{R}^{-1/2}\mathbf{H}\Psi^t\|_F.$$

The solution to this constrained optimization problem is constructed from the principal singular vectors of $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2t}$:

$$\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2t} = \mathbf{U}\Lambda\mathbf{V}^t = \mathbf{U}_1\Lambda_1\mathbf{V}_1^t + \mathbf{U}_2\Lambda_2\mathbf{V}_2^t$$

where as before, the subscript 1 denotes the principal components in the SVD, and

$$\Psi = \mathbf{V}_1^t\mathbf{R}^{-1/2}.$$

Though the square root of $\mathbf{R}$ is not unique, different choices of $\mathbf{R}^{1/2}$ will not change the singular values Λ of $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2t}$. Although the singular vectors $\mathbf{U}$ and $\mathbf{V}$ depend on the choice of $\mathbf{R}^{-1/2}$, the composition $\Psi = \mathbf{V}_1^t\mathbf{R}^{-1/2}$ is the same for all choices of the square root.

The singular values of $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2t}$ are the canonical correlation (c.c.) coefficients between the past $\mathbf{Y}^-$ and the future $\mathbf{Y}^+$ [32]. It was shown by [33], [34], that for the Gaussian case, the c.c. coefficients between $\mathbf{Y}^+$ and $\mathbf{Y}^-$ provide a measure of the *mutual information* between $\mathbf{Y}^-$ and $\mathbf{Y}^+$. A heuristic derivation of the formula for the mutual information between $\mathbf{Y}^+$ and $\mathbf{Y}^-$ may be found in [31].

The canonical components of $\mathbf{Y}^-$ (with respect to $\mathbf{Y}^+$) are $\mathbf{x}_k = v_k^t\mathbf{R}^{-1/2}\mathbf{Y}^-$ and the mutual information between each x_k and $\mathbf{Y}^+$ is $-0.5\log(1-\lambda_k^2)$. Thus, the p components from Span $(\mathbf{Y}^-)$ that maximize the mutual information with $\mathbf{Y}^+$ are the p canonical components $x_1, x_2, \cdots, x_p$ with the p largest c.c. coefficients. Just as the principal components of a random vector maximize the self-information content, the canonical-components approximation maximizes the mutual information in the 2-vector problem. Thus, it seems that a natural choice for the components of the partial state are the canonical components of $\mathbf{Y}^-$ that have the largest c.c. coefficients, and consequently, the maximum mutual information (with respect to $\mathbf{Y}^+$). Akaike first suggested the use of c.c. analysis for partial-state selection, and subsequently, many approximate modeling algorithms have been proposed [22], [35], [36], that use such an approximation.

4.3.1. Relation to the Desai and Pal stochastic balancing. Desai and Pal [20]–[22] pointed out that the nonzero c.c. coefficients between $\mathbf{Y}^-$ and $\mathbf{Y}^+$ are precisely the ei-

genvalues of the full-order $\mathbf{P}_{\max}^{-1}\mathbf{P}_{\min}$ in any coordinate system. We have seen that these coordinate-invariant eigenvalues are squares of the nonzero entries of $\mathbf{P}_{\min}$, when the covariance-equivalent class is in the Desai and Pal stochastically-balanced coordinates $\mathbf{P}_{\max}^{-1} = \mathbf{P}_{\min} = \Lambda$. However, we also have the following equalities: $\mathbf{P}_{\max}^{-1} = \boldsymbol{O}^t\mathbf{R}^{-1}\boldsymbol{O}$ and $\mathbf{P}_{\min} = \Psi\mathbf{R}\Psi^t$. Therefore, in the Desai and Pal stochastically balanced coordinates $\mathbf{R}^{-1/2}\boldsymbol{O}$ must equal $\mathbf{U}\Lambda^{1/2}$, and $\Psi\mathbf{R}^{1/2}$ must equal $\Lambda^{1/2}\mathbf{V}^t$, for some $\mathbf{U}$ and $\mathbf{V}$ with orthonormal columns. However, the composition $\boldsymbol{O}\Psi$ equals $\mathbf{H}\mathbf{R}^{-1}$, and so we obtain the following equality

$$\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t} = \mathbf{U}\Lambda\mathbf{V}^t$$

where Λ is a diagonal matrix whose entries are square roots of the eigenvalues of $\mathbf{P}_{\max}^{-1}\mathbf{P}_{\min}$, and $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns. Thus, the singular values of $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ that are the c.c. coefficients between the past and the future are precisely the square roots of the coordinate-invariant eigenvalues of $\mathbf{P}_{\max}^{-1}\mathbf{P}_{\min}$. The c.c. algorithm is therefore, equivalent to model reduction via the Desai and Pal stochastic balancing. For a discussion of the appropriateness of the mutual information criterion to the approximate stochastic realization problem, the reader is referred to [31].

4.3.2. Relation to the phase factor. Recall that the target matrix used in the PC-H method is the Hankel $\mathbf{H}$ constructed from the covariance lags of the process. The covariance lags $r(m)$ are the coefficients in the power-series expansion of $S(z) = \rho H(z)H(z^{-1})$ for the full-order system. The function $S(z)$ is called the magnitude factor of the full-order system $H(z)$, because on the unit circle we have $S(e^{j\omega}) = \rho\,|H(e^{j\omega})|^2$. The information about $H(e^{j\omega})$ missing in $S(e^{j\omega})$ is the phase, and this is available in the all-pass system

$$\Phi(z) = \frac{H(z)}{H(z^{-1})}.$$

In fact, we have

$$\rho(H(z))^2 = S(z)\Phi(z).$$

Hence, $\Phi(z)$ is called the phase-factor of the system $H(z)$. The reason for the nomenclature becomes even more obvious on the unit circle, where we have

$$H(e^{j\omega}) = \left(\frac{1}{\rho}S(e^{j\omega})\Phi(e^{j\omega})\right)^{1/2},$$

$$|\Phi(e^{j\omega})| = 1, \qquad \text{Angle}\,[\Phi(e^{j\omega})] = 2 * \text{Angle}\,[H(e^{j\omega})].$$

Stochastic model reduction based on the above phase factor has also been suggested [37], and it is closely related to c.c. analysis. It turns out that the target matrix $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ used in c.c. analysis is related to the phase factor of the full-order, minimum-phase system that corresponds to the given covariances. If the given covariances correspond to the output of a large-order, minimum-phase system $H_{\min}(z)$ driven by white noise, then the matrix $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ is equal to the Hankel matrix constructed from the impulse-response (causal part only) of its phase factor $\Phi(z)$.

CLAIM. Let the stable impulse response (inverse z-transform) of the phase factor be

$$\frac{H_{\min}(z)}{H_{\min}(z^{-1})} = \sum_{k=-\infty}^{\infty} c_k z^{-k},$$

then there exists a square root $\mathbf{R}^{1/2}$ that makes the composition $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ equal to the Hankel operator

$$\mathbf{C} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \cdots \\ c_2 & c_3 & c_4 & c_5 \cdots \\ c_3 & c_4 & c_5 & c_6 \cdots \\ c_4 & c_5 & c_6 & c_7 \cdots \\ \cdot & \cdot & \cdot & \cdot \cdots \\ \cdot & \cdot & \cdot & \cdot \cdots \\ \cdot & \cdot & \cdot & \cdot \cdots \end{bmatrix},$$

and for any other choice of square root, $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ has the same singular values as $\mathbf{C}$, and it will lead to the same approximation.

The proof of this claim is deferred to the Appendix. We can now state that while the PC-H method works on the Hankel operator corresponding to the magnitude factor $S(z)$, the c.c. approach works on the Hankel operator corresponding to the phase factor $\Phi(z)$. This result is useful in demonstrating the sensitivity of the poles of $H_{\min}(z)$ to perturbations in the matrix $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$. It is shown in Appendix B of [38], that the first-order partial derivative of a pole β_i of the system $H_{\min}(z)$ to the entries c_k in the Hankel matrix $\mathbf{C}$ is

$$\frac{\delta\beta_i}{\delta c_k} = -\beta_i^{k+1}(1-\beta_i^2)\prod_{m=1}^{p}\left(\frac{1-\alpha_m\beta_i^{-1}}{1-\alpha_m\beta_i}\right)\prod_{n\neq i}\left(\frac{1-\beta_n\beta_i}{1-\beta_n\beta_i^{-1}}\right)$$

where α_i, β_i, $i = 1, 2, \cdots, p$ are the zeros and poles, respectively, of $H_{\min}(z)$. Thus when two poles are close together (as in high resolution problems where two close spectral peaks are to be resolved), then the poles are very sensitive to perturbations in the c_k parameters, especially when there are no zeros close to the poles. On the other hand, the poles may not be as sensitive to covariance perturbations, because

$$\frac{\delta\beta_i}{\delta r(m)} = 0.$$

Hence the problem of model estimation from the covariances is numerically well conditioned, but the use of the matrix $\mathbf{C}$ as an intermediate step increases the numerical sensitivity causing finite precision errors to be magnified in the pole estimates.

4.4. The predictive efficiency criterion. The previous two approaches to approximate stochastic modeling (the principal components of $\mathbf{H}$ and the canonical correlations method) were derived by generalizing the correlation-maximizing property and the information-maximizing property of the principal components (of a random vector) to the 2-vector problem. Recall however that the function of the partial state is to predict the future output well. Hence, instead of maximizing its correlation with $\mathbf{Y}^+$ or its mutual information with respect to $\mathbf{Y}^+$, it might be more appropriate to generalize the reconstruction-efficiency property of the principal components approximation to the 2-vector problem.

The principal-components approximation of a random vector provides an optimal data compression that maximizes its ability to reconstruct the full-sized vector. In the partial-state selection problem for approximate modeling, we come across a similar problem, that of compressing $\mathbf{Y}^-$ into a partial state that can best predict $\mathbf{Y}^+$. Taking a hint from the reconstruction-efficiency of the principal components of a random vector, we might wish to compress $\mathbf{Y}^-$ into a partial state that has the smallest error in predicting

$\mathbf{Y}^+$. Our partial-state selection problem is then to pick a partial state $\mathbf{x}_{\text{partial}} = \Psi\mathbf{Y}^-$ to

$$\text{minimize } E[\|\mathbf{Y}^+ - \mathbf{Y}^+\backslash\mathbf{x}_{\text{partial}}\|^2].$$

The inherent constraint here is that Ψ should have only p rows.[7] Such a criterion was first used by Rao in multivariate statistics for the 2-vector problem [19]. Since $\mathbf{x}_{\text{partial}} = \Psi\mathbf{Y}^-$, it can be shown using (4) that

$$\mathbf{Y}^+\backslash\mathbf{x} = \mathbf{H}\Psi^t(\Psi\mathbf{R}\Psi^t)^{-1}\mathbf{x} \tag{10}$$

and the prediction error to be minimized is Trace $(\mathbf{R} - \mathbf{H}\Psi^t(\Psi\mathbf{R}\Psi^t)^{-1}\Psi\mathbf{H}^t)$. Equivalently, we must choose a $p \times \infty$ matrix Ψ that maximizes Trace $((\Psi\mathbf{H}^t\mathbf{H}\Psi^t)(\Psi\mathbf{R}\Psi^t)^{-1})$. The solution to this optimization problem is as follows. The p rows of Ψ must be a basis for the space spanned by the p generalized eigenvectors of the matrix pencil $(\mathbf{H}^t\mathbf{H}, \mathbf{R})$, corresponding to the p largest generalized eigenvalues. If $\mathbf{R}$ is invertible, as is the case when the model is strictly stable, we can obtain Ψ from the eigenvectors of $\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^t$ instead.[8] Let the eigendecomposition (or SVD) of $\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^t$ be

$$\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^t = \mathbf{U}\Sigma^2\mathbf{U}^t = \mathbf{U}_1\Sigma_1^2\mathbf{U}_1^t + \mathbf{U}_2\Sigma_2^2\mathbf{U}_2^t$$

and let subscript "1" denote the principal components, as before. Then, the predictive-efficiency criterion is optimized when

$$\Psi = \mathbf{A}\mathbf{U}_1^t\mathbf{H}\mathbf{R}^{-1}$$

where $\mathbf{A}$ is any $p \times p$ invertible matrix.

Note that this solution is different from Akaike's solution and the PC-H approximation, because under perturbations, $\mathbf{H}$ will be full rank and the principal components of $\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^t$, $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$, and $\mathbf{H}$ are all different. Rao himself states that his generalized principal components analysis for studying the association between two random vectors is different from Hotelling's canonical correlations analysis.

4.4.1. The Unweighted Principal Components (UPC) Algorithm. After choosing the partial-state components using the predictive efficiency criterion, we still must obtain the corresponding parameter estimates. The parameter-estimation step (Step 2) is taken from the deterministic identification algorithm of [3]. It is assumed here that the model order p is estimated (or given) prior to the model parameter estimation. From that point on, the rest of the Unweighted Principal Components (UPC) algorithm is [31], [42]:

Step 1. Perform an eigendecomposition of

$$\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^t = \mathbf{U}\Sigma^2\mathbf{U}^t = \mathbf{U}\Sigma_1^2\mathbf{U}_1^t + \mathbf{U}_2\Sigma_2^2\mathbf{U}_2^t$$

and retain only the principal components (denoted by subscript 1). Now Ψ can be any basis from the row span of $\mathbf{U}_1^t\mathbf{H}\mathbf{R}^{-1}$, i.e.,

$$\Psi = \mathbf{A}\mathbf{U}_1^t\mathbf{H}\mathbf{R}^{-1} \quad \text{for any invertible } p\times p \text{ matrix } \mathbf{A}.$$

Different choices of $\mathbf{A}$ will correspond to different coordinate transformations of the partial state. We choose

$$\Psi = \Sigma_1^{-1/2}\mathbf{U}_1^t\mathbf{H}\mathbf{R}^{-1}.$$

---

[7] Without such a constraint, no size compression is required, and the entire past $\mathbf{Y}^-$ can be used as the state.

[8] When $\mathbf{R}$ is singular, the process is purely sinusoidal, and this solution is the same as the Toeplitz approximation method of [39]–[41].

Then, (10) indicates that

$$\mathbf{Y}^{+}\backslash\mathbf{x}_{\text{partial}} = \mathbf{H}\Psi^{t}(\Psi\mathbf{R}\Psi^{t})^{-1}\mathbf{x}_{\text{partial}},$$

implying that the extended observability matrix estimate is

$$\boldsymbol{O} = \mathbf{H}\mathbf{R}^{-1}\mathbf{H}^{t}\mathbf{U}_1\Sigma_1^{-1/2}(\Sigma_1^{-1/2}\mathbf{U}_1^{t}\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^{t}\mathbf{U}_1\Sigma_1^{-1/2})^{-1} = \mathbf{U}_1\Sigma_1^{1/2}$$

Step 2. But, the partial-state is not a "true state" of a linear time-invariant system, and the $\boldsymbol{O}$ and Ψ matrices do not have the required structure. Hence, as in the deterministic identification algorithm of [3], we resort to a second approximation, and $\mathbf{F}$ is obtained as the least-squares solution of (see (3))

$$\boldsymbol{O}_1\mathbf{F} = \boldsymbol{O}_2$$

where $\boldsymbol{O}_1$ ($\boldsymbol{O}_2$) is formed from $\boldsymbol{O}$ by deleting the last (first) row. Moreover, $\mathbf{h}$ and $\mathbf{T}$ are the first row and column of $\boldsymbol{O}$ and Ψ, respectively. Therefore, the parameter estimates are:

$$\mathbf{h} = \text{1st row of } \boldsymbol{O},$$

$$\mathbf{T} = \text{1st column of } \Psi,$$

$$\mathbf{F} = \boldsymbol{O}_1^{\dagger}\boldsymbol{O}_2$$

where the superscript † stands for the pseudoinverse.

4.4.2. Relation to internal balancing of the minimum-phase model. It can be easily verified that the eigenvalues of $\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^{t}$ are precisely the squares of the state-variances of the internally balanced, full-order, minimum-phase model; and that, consequently, the UPC method effectively performs balanced model reduction of the full-order, minimum-phase system corresponding to the given covariance sequence.

We will first show that the UPC algorithm is a stochastic version of the deterministic identification algorithm of [3]. Recall that $\mathbf{Y}^{+}\backslash\mathbf{Y}^{-} = \boldsymbol{O}\mathbf{x}_{\min}$ that in turn is equal to $\boldsymbol{H}_{\min}\mathbf{V}_{\min}^{-}$ because $\mathbf{x}_{\min} = \boldsymbol{C}_{\min}\mathbf{V}_{\min}^{-}$. Moreover, using (4), we saw that $\mathbf{Y}^{+}\backslash\mathbf{Y}^{-} = \mathbf{H}\mathbf{R}^{-1}\mathbf{Y}^{-}$. Combining the two, we get

$$\mathbf{H}\mathbf{R}^{-1}\mathbf{Y}^{-} = \boldsymbol{H}_{\min}\mathbf{V}_{\min}^{-}.$$

Therefore, the covariance matrices of the two vectors must also be the same. And thus, we come to the rather surprising result:

$$\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^{t} = \rho_{\min}\boldsymbol{H}_{\min}\boldsymbol{H}_{\min}^{t}.$$

Thus, the eigenvalues of $\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^{t}$ are proportional to the singular values of the impulse-response Hankel $\mathbf{H}_{\min}$ of the minimum-phase model. Hence, the UPC method is a stochastic generalization of the deterministic identification algorithm of [3] that works on covariance data instead of impulse-response measurements.

Since the singular values of $\mathbf{H}_{\min}$ are precisely square roots of the coordinate-invariant eigenvalues of $\mathbf{W}\mathbf{K}$ for the minimum-phase model, it implies that the UPC method performs balanced model reduction on the minimum-phase model corresponding to the given covariances.

4.4.3. Some comparisons. In the previous section, it has been shown that the matrix approximated by its dominant singular vectors in the UPC method is $\boldsymbol{H}\boldsymbol{H}^{t}$. It also has been shown that the matrix used in c.c. analysis is equal to the Hankel matrix built from the impulse response of the all-pass system $\Phi(z) = H_{\min}(z)/H_{\min}(z^{-1})$. If we use the notation $\Gamma[\cdot]$ to denote the Hankel matrix constructed from the causal part of the inverse

z-transform of the function "$\cdot$" within the square brackets, then we have

$$\boldsymbol{H}_{\min} = \Gamma[H_{\min}(z)],$$

$$\mathbf{H}\mathbf{R}^{-1}\mathbf{H}^t = \Gamma[H_{\min}(z)] \cdot \Gamma[H_{\min}(z)]^t,$$

$$\mathbf{H} = \Gamma[S(z)],$$

$$\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t} = \Gamma[\Phi(z)]$$

where $S(z) = \rho H_m(z)H_m(z^{-1})$ and $\Phi(z) = H_{\min}(z)/H_{\min}(z^{-1})$.

Thus the PC-H approximation uses the magnitude factor of the full-order system, the c.c. approximation uses the phase factor, and the UPC approximation uses the transfer function of the minimum-phase system. Alternate interpretations of the three methods presented in this section may be found in [31].

4.4.4. Connections to other methods. We have already seen how the UPC method relates to balanced model reduction, to the deterministic identification algorithm of [3], and to Fujishige model reduction. It turns out that the matrix used by the UPC method for SVD is also used in a realization algorithm due to Mullis and Roberts [43]. This realization algorithm uses both output covariances and impulse-response coefficients. When an equal number of covariance lags and impulse-response coefficients are known, they utilize an $(n+1)\times(n+1)$ matrix $\mathbf{K}(n, n)$ (not to be confused with the controllability grammian) constructed from

$$\{r(0), r(1), \cdots, r(n); i(0), i(1), \cdots, i(n)\}$$

whose (j, k)th element is

$$r(|j-k|) - \rho \sum_{l=0}^{\min(k,j)} i(l)i(l+|k-j|), \qquad j,k = 0, 1, \cdots, n.$$

It is apparent that this matrix is the leading submatrix of $\mathbf{R} - \rho\mathbf{L}\mathbf{L}^t$. Taking the covariances of both sides of (2) $\mathbf{Y}^+ = \boldsymbol{H}\mathbf{V}^- + \mathbf{L}\mathbf{V}^+$, we arrive at

$$\mathbf{R} = \rho\boldsymbol{H}\boldsymbol{H}^t + \rho\mathbf{L}\mathbf{L}^t.$$

Therefore, $\mathbf{K}(n, n)$ is the leading submatrix of $\rho\boldsymbol{H}\boldsymbol{H}^t$. In this paper, we have indicated how the matrix $\rho\boldsymbol{H}\boldsymbol{H}^t$ can be obtained from output covariances alone, when the model is minimum-phase.

5. Simulations and concluding remarks. Every system approximation criteria discussed in this paper is based on system parameters that are invariant to coordinate transformations. However, each criterion is different, and measures different quantities. For instance, only internal balancing centers the state between the input and the output, while covariance balancing and the Desai and Pal stochastic balancing do not do so, partly because the input sequence is not uniquely known in the stochastic realization problem. We have shown that an internally balanced approximation of the unique minimum-phase transfer-function corresponding to the given covariances, optimizes predictive efficiency in Rao's sense. We have shown that such an approximation is equivalent to a principal-components approximation of the information interface between the past output and the future output. We have presented an algorithm to directly construct the reduced-order model from the covariances of the output process.

The practicality of the algorithm, however, depends very much on its numerical performance: its sensitivity to covariance-estimation errors, finite-precision errors, and to finiteness in the dimensions of the $\mathbf{H}$ and $\mathbf{R}$ matrices. Our simulations are promising, in that they seem to indicate that finiteness in the length of the estimated covariance

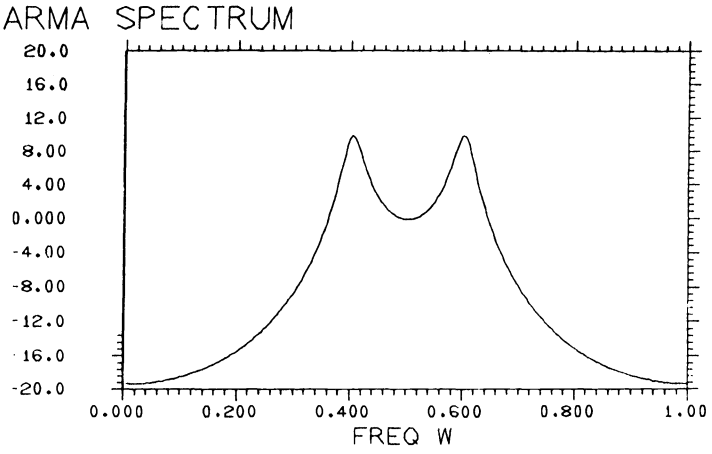


FIG. 1(a). *True spectrum*.

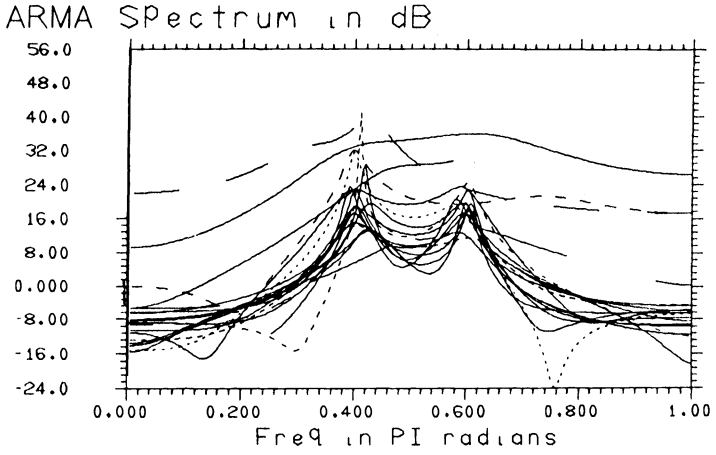


FIG. 1(b). C.C. *estimate*.

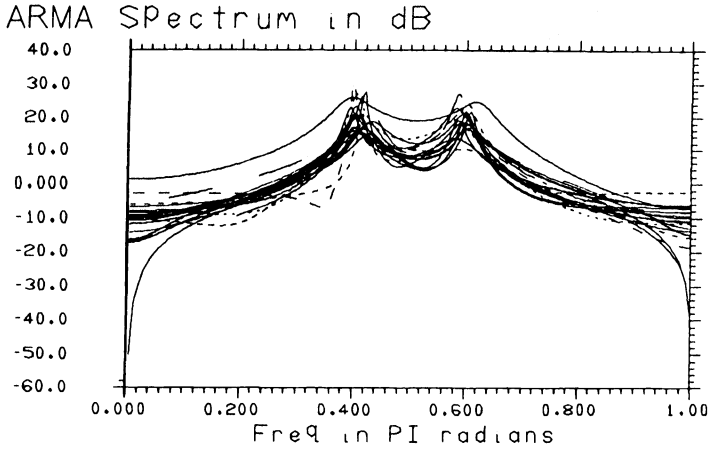


FIG. 1(c). UPC *estimate*.

TABLE 1

Method	Miss ratio	Radius		Angle (in π radians)	
		mean	st. dev.	mean	st. dev.
PC-H Method	33/200	0.93023	0.03782	0.90679	0.02405
C.C. Method	69/200	0.86709	0.09146	0.90714	0.02692
UPC Method	7/200	0.93009	0.03309	0.91254	0.02085

sequence does not badly affect the algorithm's performance. However, a more detailed analysis is necessary, before a conclusive statement can be made.

Below, we present some simulation examples to illustrate the application of these methods.

Example 1. In this example, we consider the problem of estimating a fourth-order rational spectrum from a short record of one sample sequence of the stochastic process. The stochastic process was generated using the following model:

$$y(n) = -1.456y(n-2) - 0.81y(n-4) + v(n)$$

that has four doubly symmetric poles approximately at $0.95e^{\pm j0.4\pi}$ and $0.95e^{\pm j0.6\pi}$. The first 30 covariance lags were estimated from a data record of length 120 from a single sample sequence, and the system parameters were estimated from these covariance estimates using the UPC algorithm of this thesis, and the c.c. algorithm of [22]. The size of the matrices **H** and **R** used in both algorithms was 15×15, and a rank-4 approximation was used in their component selection steps. The algorithms were repeated on 20 independent data sets generated by driving the model with 20 different pseudowhite-noise sequences, and the 20 spectral estimates from each method were plotted over one another. The resultant plots and the true model spectrum are displayed in Figs. 1(a)–1(c).

The c.c. estimates show a slightly larger variation and many of the c.c. estimates failed to resolve the two peaks in the spectrum and instead reproduced one peak at the center. The better ability of the UPC methods to resolve spectral peaks is brought out more dramatically when the number of simulations is increased further, as in the next example.

Example 2. The problem considered here is the estimation of the poles of a second-order rational model from unbiased estimates of the first 26 covariance lags. The true pole positions (on the z-plane) are $0.9e^{\pm j0.9\pi}$, so that the problem is one of resolving two spectral peaks that are 0.2π radians apart. The covariance lags were estimated from a data record of length 120, from a single sample sequence of the output random process. Two hundred statistically independent data records were generated by driving the model with different (pseudo) white-noise sequences, and 200 sets of estimates for the 26 covariance lags were obtained.

The PC-H method using the algorithm of [29], the c.c. approach using the algorithm of [22], and the UPC algorithm were tested on these 200 sets of the 26 covariance estimates. The matrices **H** and **R** used were of size 13×13 in all three algorithms, and a rank-2 approximation was used in their first steps. The second step ($F = \boldsymbol{O}_1^{\dagger} \cdot \boldsymbol{O}_2$) is common to all three algorithms, and is taken from [3]. The results, in terms of the failure rate, the mean and standard deviation (taken over only the successful trials) of the pole estimates are tabulated in Table 1.[9]

[9] A trial is considered a failure if both pole estimates are on the real axis, which means the method has failed to resolve the two peaks.

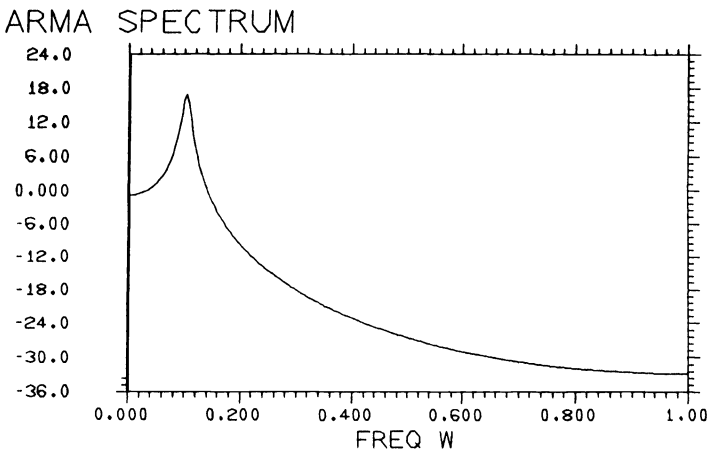


FIG. 2(a). *True spectrum*.

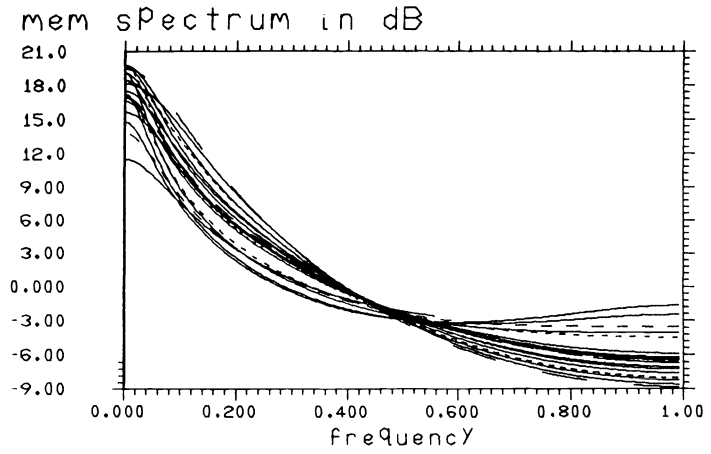


FIG. 2(b). MEM (2) *estimate*.

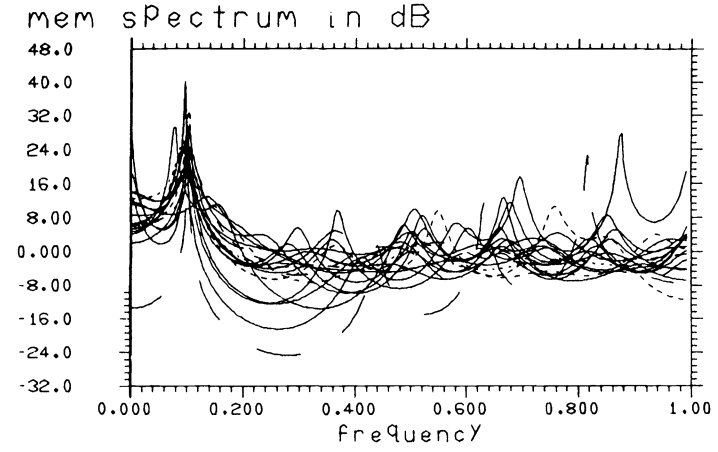


FIG. 2(c). MEM (12) *estimate*.

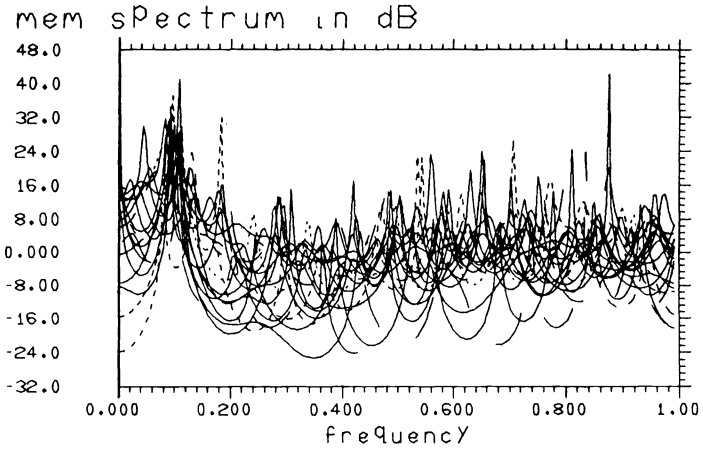


FIG. 2(d). MEM (25) *estimate*.

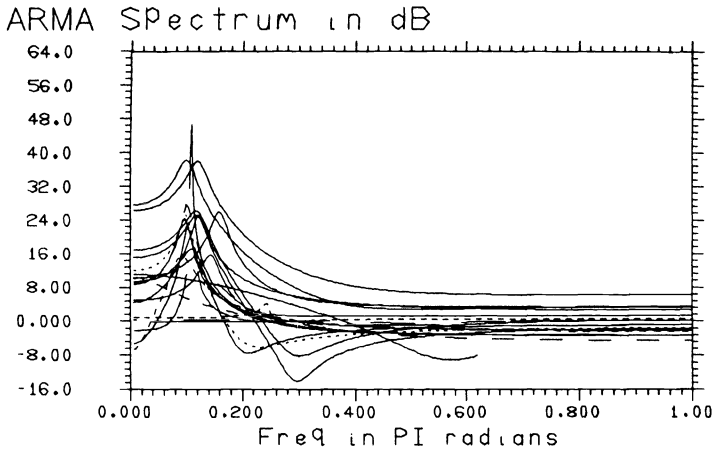


FIG. 2(e). C.C. *estimate*.

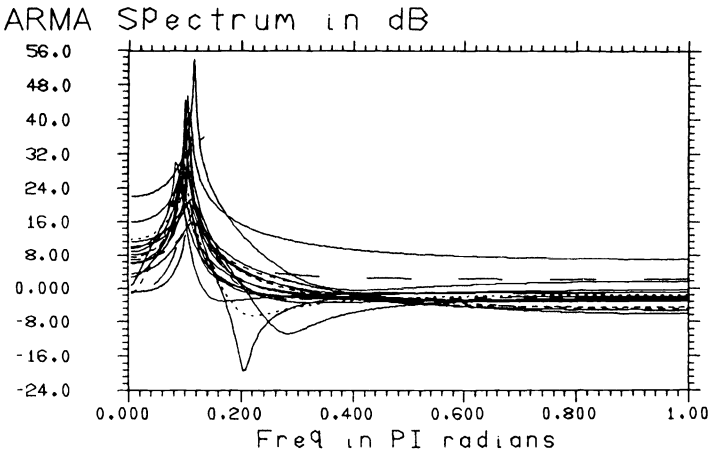


FIG. 2(f). UPC *estimate*.

The results appear to indicate that the unweighted predictive-efficiency criterion is particularly suitable for high resolution rational spectral estimation. Accordingly, for the next example we include the estimates obtained by more popular spectrum estimation methods that estimate difference-equation parameters instead of state-space parameters.

Example 3. In this example, we consider the problem of estimating the power spectrum of a second-order autoregressive process from a short record of the process in additive white noise. The stochastic process was synthesized using the following model:

$$y(n) = 1.864y(n-1) - 0.96y(n-2) + v(n)$$

that has two symmetric poles approximately at $0.98 \exp \pm j0.1\pi$, driven by a pseudo-random white-noise sequence of unit variance. Another statistically independent pseudorandom white-noise sequence of variance 10.0 was added to the output. Twenty such statistically independent data records of the time series, each of length 64, were generated.

The first 26 covariance lags $\{r(0), r(1), \cdots, r(25)\}$ were estimated using the unbiased covariance estimator from each 64-point record separately, and the model parameters were estimated from these 20 sets of covariance estimates using a variety of methods. The maximum entropy method (MEM) was first used to obtain all-pole models of orders 2, 12, and 25, that exactly match the first 2, 12, and 25 lags, respectively. The true power spectrum is plotted in Fig. 2(a), and the 20 MEM spectral estimates are plotted one over the other in Figs. 2(b), 2(c), and 2(d). MEM does not perform very well, because an autoregressive process in additive white noise needs a second-order pole-zero model or a high-order all-pole model. Hence, the second-order all-pole model that exactly matches the first two lags, fails to resolve the two peaks (that are 0.2π radians apart) in every trial. Instead, it puts both the poles on the real line, and detects only one peak at zero. On the other hand, high-order all-pole fits that use more lags give rise to spurious peaks, though the true peaks at $\pm 0.1\pi$ are resolved well. Sometimes, the strength of the spurious peak can be larger than the true peak, and the spectral shape is not reproduced with any degree of fidelity. The next two plots in Figs. 2(a)–2(f) are the spectral estimates obtained by the canonical correlations (c.c.) algorithm of [22] and by the UPC method of this paper. Here, 13×13 matrices were employed that used all the 25 covariance lags. The model order p was assumed to be predetermined to be 2, and a rank-2 approximation was used in all trials. The spectral shape is well reproduced by both methods. But, the c.c. approach fails to resolve the two peaks in two of the 20 trials. This loss in resolution capability and the large variance in pole estimates can be attributed to the numerical sensitivity problems inherent in the c.c. approach when the poles are close together [38].

Table 2 lists the failure rate, and the mean and standard deviation (taken over successful trials only) of the pole-estimates obtained by a variety of methods. For methods

TABLE 2

Method	Miss ratio	Radius		Angle (in π radians)	
		mean	st. dev.	mean	st. dev.
HOYW-2	7/20	0.9592	0.1558	0.0922	0.0306
HOYW-12	0	0.9889	0.0239	0.1064	0.0192
Cov.-of-Cov.	0	0.9838	0.0122	0.0994	0.0090
C.C.	2/20	0.9581	0.0916	0.1485	0.0964
UPC	0	0.9863	0.0181	0.1000	0.0088

that estimate second-order models, a trial is considered to be a failure, if both pole-estimates are on the real axis, which means that the method has failed to resolve the two peaks. For methods that obtain higher-order models, a trial is a failure, if the pole-estimates that are closest to the true poles are on the real axis. For the first two rows in Table 2, the second- and twelfth-order model's difference-equation parameters were obtained by exactly solving the first 2 or 12 covariance recurrence equations (called the higher-order Yule–Walker equations [44], [45]) using $2p$ covariances. Thus, only 4 lags and 24 lags were used respectively. Because of covariance estimation errors in the given lags, an exact fitting method such as this does not perform well. The second-order exact realization fails to resolve the peaks seven times in 20 trials, but it does better than second-order MEM that failed every trial. The higher-order exact rational realization does better than the second-order rational model, for it successfully resolves the two peaks every time. Yet, it is an unnecessarily complex model, and the variance of its pole-estimates is high. On the other hand, a second-order approximate realization based on all 25 lags, using the covariance-of-covariances method of [45], appears to have a much lower deviation in the pole-estimates. Here, a least-squares solution is found for the overdetermined system of 23 HOYW equations. The least-squares solution (under the unit norm constraint) is also the right singular vector of 23×3 Hankel matrix $\mathbf{H}$, corresponding to the smallest singular value [25]. The next two entries are for the SVD-based state-space approximations: the c.c. method of [22] and the UPC method of this paper. Note that the high sensitivity of the c.c. approach in the high-resolution problem also causes large deviations in the pole-estimates. This sensitivity, in fact, reduces the resolution capability of the c.c. approach as is demonstrated by the previous examples. More computer simulations of the UPC method are reported in [41].

In conclusion, we have presented three different criteria and three methods for approximate stochastic realization, as applications of the ideas of balancing introduced by Moore. Different generalizations lead to different kinds of balancing, and to three approximate stochastic realization methods. We have tried to demonstrate that an internally balanced approximation of the minimum-phase model corresponding to the given covariances leads to good approximate models. We have developed an algorithm that constructs such a reduced-order approximate model directly from covariance data. The UPC algorithm uses the same partial-state selection criterion that is used in balanced model reduction and Fujishige's model reduction. Balanced model reduction and the method of Fujishige, Nagai, and Sawaragi are identical, except that one uses a deterministic justification, that of retaining the most reachable (controllable) and most observable state-components, while the other uses a stochastic justification, that of retaining the state components with the highest predictive efficiency for the future. In that sense, the UPC algorithm is closer to Fujishige's model reduction. The difference between them lies in the crucial fact that the Fujishige method needs the full-order model, and does not require it to be minimum-phase; while the UPC algorithm works with output covariances *only*, and in effect, performs a Fujishige-like model reduction on the *minimum-phase* system corresponding to the given covariances.

Appendix.

CLAIM. Let the stable impulse response (inverse z-transform) of the phase factor be

$$\frac{H_{\min}(z)}{H_{\min}(z^{-1})} = \sum_{k=-\infty}^{\infty} c_k z^{-k}$$

then there exists a square root $\mathbf{R}^{1/2}$ that makes the composition $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ equal to the Hankel operator

$$\mathbf{C}=\begin{bmatrix} c_1 & c_2 & c_3 & c_4\cdots \\ c_2 & c_3 & c_4 & c_5\cdots \\ c_3 & c_4 & c_5 & c_6\cdots \\ c_4 & c_5 & c_6 & c_7\cdots \\ \cdot & \cdot & \cdot & \cdot\cdots \\ \cdot & \cdot & \cdot & \cdot\cdots \\ \cdot & \cdot & \cdot & \cdot\cdots \end{bmatrix},$$

and for any other choice of square root, $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ has the same singular values as $\mathbf{C}$, and it will lead to the same approximation.

Proof. The phase factor $\Phi(z)$ has poles both inside and outside the unit circle, and its stable impulse-response has both causal and anticausal parts. In general, the Hankel matrix constructed from the causal part of the impulse-response of a system is an operator that maps the past input into the future output. First, let $w(t)$ be the output of $\Phi(z)$, when the input is $v_{\min}(t)$. We will determine the Hankel operator $\mathbf{C}$ that takes the past-input $\mathbf{V}^-_{\min}$ into the future-output $\mathbf{W}^+$.

It is immediately obvious that the process $y(t)$ can be alternately generated by feeding the new process $w(t)$ to the system $H_{\min}(z^{-1})$. This implies that

$$y(t)=\sum_{k=0}^{\infty} i_{\min}(k)w(t+k),$$

that in matrix notation, translates to $\mathbf{Y}^+ = \mathbf{L}^t_{\min}\mathbf{W}^+$, where $L_{\min}$ is the lower triangular Toeplitz matrix (of (2)) built from the impulse response of the minimum-phase model. Now, $H_{\min}(z^{-1})$ has all poles outside the unit circle, and is stable if run backwards in time, as above. In addition, it is also minimum-phase, which means its inverse, $1/H_{\min}(z^{-1})$, is also stable if run backwards in time. In other words, the matrix $\mathbf{L}^t_{\min}$ is invertible, and

$$\mathbf{W}^+ = \mathbf{L}^{t^{-1}}_{\min}\mathbf{Y}^+.$$

Combining this with (2): $\mathbf{Y}^+ = \boldsymbol{H}_{\min}\mathbf{V}^-_{\min} + \mathbf{L}_{\min}\mathbf{V}^+_{\min}$, we conclude that the Hankel operator (associated with the phase-factor) that maps the past input $\mathbf{V}^-_{\min}$ into the future output $\mathbf{W}^+$ is

$$\mathbf{C}=\mathbf{L}^{t^{-1}}_{\min}\boldsymbol{H}_{\min}.$$

Let us now express the covariance Hankel $\mathbf{H}$ in terms of $\boldsymbol{H}_{\min}$ and $\mathbf{L}_{\min}$. From the definition of the impulse-response $i_{\min}(k)$

$$y(t)=\sum_{k=0}^{\infty} i_{\min}(k)v_{\min}(t-k)$$

it immediately follows that

$$\mathbf{Y}^- = \mathbf{L}^t_{\min}\mathbf{V}^-_{\min}, \quad \text{and} \quad \mathbf{R}=\mathrm{E}[\mathbf{Y}^-\mathbf{Y}^{-t}]=\rho_{\min}\mathbf{L}^t_{\min}\mathbf{L}_{\min}.$$

Therefore, $\rho^{1/2}_{\min}\mathbf{L}^t_{\min}$ is a valid square root of $\mathbf{R}$. In addition, we have

$$\mathbf{H}=\mathrm{E}[\mathbf{Y}^+\mathbf{Y}^{-t}]=\mathrm{E}[(\boldsymbol{H}_{\min}\mathbf{V}^-_{\min}+\mathbf{L}_{\min}\mathbf{V}^+_{\min})(\mathbf{L}_{\min}\mathbf{V}^-_{\min})^t]=\rho_{\min}\boldsymbol{H}_{\min}\mathbf{L}_{\min}.$$

For the same reasons as before (the model is minimum phase), $\mathbf{L}_{\min}$ is invertible, and we get $\boldsymbol{H}_{\min} = \mathbf{H}\mathbf{L}_{\min}^{-1}$. Therefore, we have

$$\mathbf{C} = \mathbf{L}_{\min}^{t^{-1}}\mathbf{H}\mathbf{L}_{\min}^{-1}.$$

Hence the claim that the Hankel operator corresponding to the phase factor is $\mathbf{C} = \mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$, for the particular choice of square root $\mathbf{R}^{1/2} = \mathbf{L}_{\min}^t$. Any other square root of $\mathbf{R}$ can be written as $\mathbf{L}_{\min}^t\mathbf{Q}$ (where $\mathbf{Q}$ is an orthogonal matrix), and then $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t} = \mathbf{Q}^t\mathbf{C}\mathbf{Q}$. Since $\mathbf{Q}$ is orthogonal, $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{R}^{-1/2^t}$ has the same singular values as $\mathbf{C}$, and the singular vectors are transformed by the matrix $\mathbf{Q}$. However, $\Psi = \mathbf{V}_1^t\mathbf{R}^{-1/2}$ is unaffected by the transformation. □

REFERENCES

[1] B. C. MOORE, *Principal component analysis in linear systems: Controllability observability and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–31.

[2] S. FUJISHIGE, H. NAGAI, AND Y. SAWARAGI, *System-theoretical approach to model reduction and system-order determination*, Internat. J. Control, 22 (1975), pp. 807–819.

[3] S. Y. KUNG, *A new identification and model reduction algorithm via singular value decomposition*, in Proc. 12th Asilomar Conference on Circuits Systems and Computers, Pacific Grove, CA, IEEE, November 1978, pp. 705–714.

[4] C. T. MULLIS AND R. A. ROBERTS, *Synthesis of minimum round-off noise fixed point digital filters*, IEEE Trans. Circuits Systems, 23 (1976), pp. 551–562.

[5] T. KAILATH, *Linear Systems*, Prentice Hall, New York, 1980.

[6] L. PERNEBO AND L. M. SILVERMAN, *Model reduction via balanced state space representations*, IEEE Trans. Automat. Control, 27 (1982), pp. 382–387.

[7] P. FAURRE, *Symposium on Optimization Nice*, Springer-Verlag, Berlin, New York, 1969.

[8] ———, *Stochastic realization algorithms*, in System Identification: Advances and Case Studies, R. K. Mehra and D. G. Lainiotis, eds., Academic Press, New York, 1976.

[9] B. D. O. ANDERSON AND T. KAILATH, *The choice of signal process models*, J. Math. Anal. Appl., 35 (1971), pp. 659–668.

[10] J. C. WILLEMS, *Least squares stationary optimal control and the algebraic Riccati equation*, IEEE Trans. Automat. Control, 16 (1971), pp. 621–634.

[11] B. D. O. ANDERSON, *Algebraic properties of minimum degree spectral factors*, Automatica, 9 (1973), pp. 491–500.

[12] H. AKAIKE, *Markovian representation of stochastic processes by canonical variables*, SIAM J. Control, 13 (1975), pp. 162–173.

[13] H. SORENSON, *Parameter Estimation: Principles and Problems*, Marcel Dekker, New York, 1980.

[14] T. KAILATH, *The innovations approach to detection and estimation theory*, Proc. IEEE, 58 (1970), pp. 680–695.

[15] M. GEVERS AND T. KAILATH, *An innovations approach to least squares estimation part VI: Discrete-time innovations-representations and recursive estimation*, IEEE Trans. Automat. Control, 18 (1973), pp. 588–600.

[16] T. KAILATH, *A view of three decades of linear filtering theory*, IEEE Trans. Inform. Theory, 20 (1974), pp. 145–181.

[17] H. HOTELLING, *Analysis of a complex of variables into principal components*, J. Educational Psych., 24 (1933), pp. 417–441 and 498–520.

[18] J. L. BROWN, JR., *Mean square truncation error in series expansions of random functions*, SIAM J. Appl. Math., 8 (1960), pp. 28–32.

[19] C. R. RAO, *The use and interpretation of principal component analysis in applied research*, Sankhyā Ser. A, 26 (1964), pp. 329–358.

[20] D. PAL, *Balanced stochastic realizations and model reduction*, Masters thesis, Washington State University, Pullman, WA, 1982.

[21] U. B. DESAI AND D. PAL, *A transformation approach to stochastic model reduction*, IEEE Trans. Automat. Control, 29 (1984), pp. 1097–1099.

[22] ———, *A realization approach to stochastic model reduction and balanced stochastic realization*, in Proc. 16th Annual Conference on Information Sciences and Systems, Princeton University, Princeton, NJ, March 1982, pp. 613–620; Internat. J. Control, 42 (1985), pp. 821–838.

[23] A. M. YAGLOM, *Outline of some topics in linear extrapolation of stationary random processes*, in Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, CA, 1965, pp. 259–278.

[24] H. AKAIKE, *Markovian representation of stochastic processes and its application to the analysis of autoregressive moving average processes*, Annals Instit. Statist. Math., 26 (1974), pp. 363–387.

[25] J. A. CADZOW, *Spectral estimation: An overdetermined rational model equation approach*, Proc. IEEE, 70 (1982), pp. 907–939.

[26] M. PREVOSTO, A. BENVENISTE, AND B. BARNOUIN, *Identification of vibrating structures subject to non-stationary excitation: A nonstationary stochastic realization problem*, in Proc. Internat. Conference on Acoustics Speech and Signal Processing, IEEE, Paris, France, May 1982, pp. 252–255.

[27] A. BENVENISTE AND J. J. FUCHS, *Single sample modal identification of a nonstationary stochastic process*, IEEE Trans. Automat. Control, 30 (1985), pp. 66–75.

[28] J. M. MACIEJOWSKI, *The use of principal components for approximate linearisation stochastic realisation and spectral factorisation*, note presented at the IEEE Colloquium on Principal Components: Model Reduction and Control at City University, London, January 7, 1983.

[29] S. Y. KUNG AND K. S. ARUN, *A novel Hankel approximation method for ARMA pole zero estimation from noisy covariance data*, in Technical Digest of the Topical Meeting on Signal Recovery and Synthesis with Incomplete Information and Partial Constraints, Optical Society of America, Incline Village, NV, January 1983, pp. WA–19.

[30] S. M. KAY AND S. L. MARPLE, JR., *Spectrum analysis—A modern perspective*, Proc. IEEE, 69 (1981), pp. 1380–1419.

[31] K. S. ARUN AND S. Y. KUNG, *Generalized principal components analysis, and its application in approximate stochastic realization*, in Theory and Practice of Modeling Stochastic Processes, U. B. Desai, ed., Kluwer, Hingham, MA, 1986, Chap. 4, pp. 75–104.

[32] H. HOTELLING, *Relations between two sets of Variates*, Biometrika, 28 (1936), pp. 321–372.

[33] I. M. GELFAND AND A. M. YAGLOM, *Calculation of the Amount of Information about a Random Function Contained in Other such Functions*, American Mathematical Society Translations Series 2, vol. 12, Providence, RI, 1959, pp. 199–246.

[34] S. KULLBACK AND R. A. LEIBLER, *On information and sufficiency*, Ann. of Math. Statist., 22 (1951), pp. 79–86.

[35] Y. BARAM, *Realization and reduction of Markovian models from nonstationary data*, IEEE Trans. Automat. Control, 26 (1981), pp. 1225–1231.

[36] J. WHITE, *Stochastic State Space Models from Empirical Data*, in International Conference on Acoustics Speech and Signal Processing, IEEE, Boston, MA, April 1983, pp. 243–246.

[37] E. A. JONCKHEERE AND J. W. HELTON, *Power spectrum reduction by optimal Hankel-norm approximation of the phase of the outer spectral factor*, IEEE Trans. Automat. Control, 30 (1985), pp. 1192–1201.

[38] A. S. KARALAMANGALA, *A principal components approach to approximate modeling and* ARMA *spectral estimation*, Ph.D. dissertation, Department of Electrical Engineering, University of Southern California, Los Angeles, CA, 1984.

[39] S. Y. KUNG, *A Toeplitz approximation method and some applications*, in Proc. Internat. Symposium on the Mathematical Theory of Networks and Systems, Santa Monica, CA, August 5–7, 1981, pp. 262–266.

[40] S. Y. KUNG, K. S. ARUN, AND D. V. BHASKARRAO, *State-space and singular value decomposition based methods for the harmonic retrieval problem*, J. Opt. Soc. Amer., 73 (1983), pp. 1799–1811.

[41] S. Y. KUNG AND K. S. ARUN, *Singular-value-decomposition algorithms for linear system approximation and spectrum estimation*, in Advances in Statistical Signal Processing, H. V. Poor, ed., JAI Press, Greenwich, CT, 1987, Chap. 6, pp. 203–250.

[42] K. S. ARUN AND S. Y. KUNG, *A new* SVD*-based algorithm for* ARMA *spectral estimation*, in Proc. IEEE ASSP Spectrum Estimation Workshop II, Tampa, FL, November 1983, pp. 266–271.

[43] C. T. MULLIS AND R. A. ROBERTS, *The use of second-order information in the approximation of discrete-time linear systems*, IEEE Trans. Acoust. Speech Signal Process., 24 (1976), pp. 226–238.

[44] G. E. P. BOX AND G. M. JENKINS, *Time Series Analysis Forecasting and Control*, Holden-Day, San Francisco, CA, 1970.

[45] A. A. BEEX AND L. L. SCHARF, *Covariance sequence approximation for parametric spectrum modeling*, IEEE Trans. Acoust. Speech Signal Process., 29 (1981), pp. 1042–1051.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 69–82, January 1990

004

MULTISPLITTING OF A SYMMETRIC POSITIVE DEFINITE MATRIX*

R. E. WHITE†

Abstract. Parallel iterative methods are studied, and the focus is on linear algebraic systems whose matrix is symmetric and positive definite. The set of unknowns may be viewed as a union of subsets of unknowns (possibly with overlap). The parallel iteration matrix is then formed by a weighted sum of iteration matrices that are associated with splittings of the matrix corresponding to the blocks. When the blocks are from a matrix in dissection form, it can be shown under suitable conditions that the parallel algorithm is convergent. When the multisplitting version of successive over-relaxation (SOR) is used, the SOR parameter is required to be less than $\omega_0 < 2.0$. Calculations done on the Alliant FX/8 multiprocessing/vector computer indicate speedups of nine to ten.

Key words. multisplitting, parallel algorithm, symmetric positive definite

AMS(MOS) subject classifications. 65F10, 65N20

1. Introduction. In this paper we continue the work of O'Leary and White in [9] on parallel algorithms generated by multisplittings of a symmetric positive definite matrix. A parallel algorithm is one whose parts can be executed concurrently by different processors of a multiprocessing computer. As indicated in [9] and by White in [14] and [15] these may be used to approximate the solutions of linear and nonlinear problems. When a multiprocessing computer is used, significant speedups can be achieved as is illustrated in [9], [14], and [15], and the last section of this paper.

We restrict our attention to the linear algebraic system

$$Au = d \tag{1}$$

where A is symmetric and positive definite. Also, we assume we can reorder the nodes via a permutation matrix P so that P^TAP has dissection form as described by George and Liu in [3]. This allows us to write the multisplitting iteration matrix as an iteration matrix of a single splitting (see Theorem 3). This single splitting is required to be P-regular, and hence, by the Householder–John theorem (see Theorem 2) in [5] and [6] the multisplitting iteration matrix will have spectral radius less than one (see Theorems 4 and 5).

In § 2 we review some of the basic concepts of multisplittings. Section 3 contains a motivating example for the results in §§ 4 and 5. In § 4 we indicate how the multisplitting may be viewed as a single P-regular splitting. Section 5 contains an application to the multisplitting version of the successive over-relaxation (SOR) algorithm applied to matrices in dissection form that are irreducibly diagonally dominant, symmetric, and have positive diagonal components. In this case there exists a $1 \leqq \omega_0 < 2$ such that if the SOR parameter is less than or equal to ω_0, then the algorithm will be convergent (see Theorem 5). The last section contains numerical experiments using different compiler directives, different overlapping blocks of unknowns, and different numbers of unknowns. These experiments were done on the Alliant FX/8 multiprocessing vector computer at Argonne National Laboratory. The Alliant FX/8 has 8 processors and each has a vector pipeline. Speedups of nine to ten over the serial codes were observed.

* Received by the editors September 9, 1986; accepted for publication (in revised form) March 20, 1989.

† Department of Mathematics, Box 8205, North Carolina State University, Raleigh, North Carolina 27695-8205 (white%crscfx40@ncsuvx.ncsu.edu). This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under contract W-31-109-Eng-38 while the author was on leave at the Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois.

2. Preliminaries. Multisplittings can be formed by considering blocks of unknowns that may overlap. This technique has also been considered by Ostrowski [12], Robert [13], and Hayes [4], and more recently by McBryan and Van DeVelde [7], Neumann, and Plemmons [8]. A multisplitting is a sequence of splittings

$$A = B_k - C_k, \qquad k = 1, \cdots, K. \tag{2}$$

If each B_k is invertible, then for each k we can form an iterative method

$$u^{n+1} = B_k^{-1} C_k u^n + B_k^{-1} d. \tag{3}$$

The multisplitting iterates in (3) can be computed concurrently. Once this has been done, we can combine the iterates by using weighting matrices, D_k. This gives the following parallel algorithm.

PARALLEL ALGORITHM. *Let $A = B_k - C_k k = 1, \cdots, K$ and assume each B_k is invertible. Let $0 \leqq D_k$ be diagonal matrices that satisfy $\sum_{k=1}^{K} D_k = I$:*

$$u^{n+1} = Hu^n + Gg \text{ where } H = \sum_{k=1}^{K} D_k B_k^{-1} C_k \text{ and } G = \sum_{k=1}^{K} D_k B_k^{-1}. \tag{4}$$

The splittings are often associated with blocks of nodes, $S_k \subset \{1, \cdots, N\}$ where A is an $N \times N$ matrix. Usually the ith components of the D_k are zero if the ith node is not in block S_k. Thus, we only need to compute those components of $B_k^{-1} C_k$ and B_k^{-1} that are in block S_k. This reduces the work per processor, and the computation can be done concurrently.

Remark. If G has an inverse, then we may view this algorithm as given by a single splitting $A = B - C$ where $B = G^{-1}$ and $C = G^{-1} H$.

Example 1. This is a parallel version of the Gauss–Seidel algorithm as described in [14]. Let $A = D - L - L^T$ be an $N \times N$ matrix where L is the negative strictly lower triangular part of A.

Suppose $\cup_{k=1}^{K} S_k = \{1, \cdots, N\}$. Define

$$B_k = D - L_k \quad \text{and} \quad C_k = L + L^T - L_k$$

where $L_k = (a_{ij}^k)$, $A = (a_{ij})$ and

$$a_{ij}^k = \begin{cases} -a_{ij}, & i, j \in S_k, \quad j < i, \\ 0 \quad \text{otherwise.} \end{cases}$$

In this case each $B_k^{-1} = (D - L_k)^{-1}$ exists and is a lower triangular matrix whose diagonal is D^{-1}. (D is invertible because A is positive definite). Since $\sum_{k=1}^{K} D_k = I$, $G = \sum_{k=1}^{K} D_k B_k^{-1}$ will also have this form, and so, G^{-1} exists. The convergence of parallel algorithms has been studied when A is an M-matrix as defined in Berman and Plemmons [1]. When the splittings are weak regular splittings, then O'Leary and White [9] proved $\rho(H) < 1$. Later White [14] showed that there exists a monotonic norm such that

$$\| H \| \leqq \| H_J \| < 1$$

where $H_J = D^{-1}(L + U)$ is the Jacobi iteration matrix and H is from the multisplitting in Example 1. Recently, Neumann and Plemmons [8] strengthened this to the following for A being irreducible:

$$\rho(H) \leqq \rho(H_J) < 1.$$

In a paper by Elsner [2] presented at the Third SIAM Conference on Applied Linear Algebra (May 1988 at Madison, WI), a more general result was presented. Let $A^{-1} \geqq 0$ and $A = B_k - C_k$ be weak regular splittings with $\underline{B} \leqq B_k \leqq \bar{B}$. If $A = \bar{M} - \bar{N} = \underline{M} - \underline{N}$ are regular splittings with $\underline{B} \leqq \bar{B}$, then $\rho(\underline{M}^{-1}\underline{N}) \leqq \rho(H) \leqq \rho(\bar{M}^{-1}\bar{N})$. The conclusion is false if $A = \bar{M} - \bar{N} = \underline{M} - \underline{N}$ are not regular splittings.

When A is symmetric and positive definite, the parallel algorithm may not converge even if the splittings are P-regular. ($A = B - C$ is P-regular if and only if B^{-1} exists and $B^* + C$ is positive definite.) The following example given in [9] illustrates this:

$$A = \begin{bmatrix} 0.75 & 0.0 \\ 0.0 & 0.75 \end{bmatrix} = \begin{bmatrix} 0.5 & -1. \\ 1. & 4. \end{bmatrix} - \begin{bmatrix} -.25 & -1. \\ 1. & 3.25 \end{bmatrix} = B_1 - C_1$$

and

$$= \begin{bmatrix} 4. & 1. \\ -1. & 0.5 \end{bmatrix} - \begin{bmatrix} 3.25 & 1. \\ -1. & -.25 \end{bmatrix} = B_2 - C_2.$$

When

$$D_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad D_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

then

$$H = \begin{bmatrix} .875 & .25 \\ .25 & .875 \end{bmatrix} \quad \text{and} \quad \rho(H) = 1.125.$$

Also,

$$G = D_1 B_1^{-1} + D_2 B_2^{-1} = \frac{1}{6} \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}.$$

When we restrict the weighting matrices, then the following theorem can be proved (see [9]).

THEOREM 1. *If A is symmetric and positive definite with P-regular splittings $A = B_k - C_k$, and $D_k = a_k I$ with $a_k \geqq 0$ and $\sum_{k=1}^{K} a_k = 1$, then for H given in* (4), $\rho(H) < 1$.

The proof of this theorem has the same flavor as the proof of the Householder–John theorem in [5] and [6]. The interested reader may wish to consult Ortega and Plemmons [10] where some interesting generalizations are considered.

THEOREM 2 (Householder–John). *Let $A = B - C$ be Hermitian and $B^* + C$ be positive definite. Then $\rho(B^{-1}C) < 1$ if and only if A is positive definite.*

3. Motivating examples. In the following example we indicate a reordering scheme that will allow us to view the multisplitting as a single P-regular splitting. This is a special case of the results in §§ 4 and 5.

Example 2. Consider $-\Delta u = f$ where the five-point finite-difference method is used with three unknowns in each direction. We may consider this as a two-block problem, as indicated by S_1 and S_2 in Fig. 1, where the center row is the overlapping subblock. This gives a 9×9 system matrix that we indicate by nine 3×3 blocks

$$A = \begin{bmatrix} A_0 & -I & \phi \\ -I & A_0 & -I \\ \phi & -I & A_0 \end{bmatrix}$$

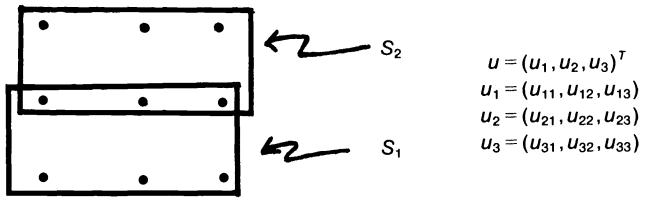


FIG. 1. *Two blocks and classical order.*

where

$$A_0 = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix},$$

$I = 3 \times 3$ identity matrix, and $\phi = 3 \times 3$ zero matrix.

Now, reorder the nodes so that the overlapping row, row 2, becomes the last row, that is, $u \rightarrow (u_1, u_3, u_2)^T$. Then the appropriate permutation matrix P gives

$$P^T AP = \begin{bmatrix} A_0 & \phi & -I \\ \phi & A_0 & -I \\ -I & -I & A_0 \end{bmatrix}.$$

The Gauss–Seidel splittings associated with the two blocks are

$$P^T AP = B_1 - C_1 = \begin{bmatrix} D_0 - L_0 & \phi & \phi \\ \phi & D_0 - L_0 & \phi \\ -I & \phi & D_0 - L_0 \end{bmatrix} - \begin{bmatrix} U_0 & \phi & I \\ \phi & U_0 & I \\ \phi & I & U_0 \end{bmatrix}$$

and

$$P^T AP = B_2 - C_2 = \begin{bmatrix} D_0 - L_0 & \phi & \phi \\ \phi & D_0 - L_0 & \phi \\ \phi & -I & D_0 - L_0 \end{bmatrix} - \begin{bmatrix} U_0 & \phi & I \\ \phi & U_0 & I \\ I & \phi & U_0 \end{bmatrix}$$

where

$$A_0 = (D_0 - L_0) - U_0, \qquad D_0 = 4I,$$

$$L_0 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad U_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

One choice of weighting matrices is

$$D_1 = \begin{bmatrix} I & \phi & \phi \\ \phi & \phi & \phi \\ \phi & \phi & \frac{1}{2}I \end{bmatrix} \quad \text{and} \quad D_2 = \begin{bmatrix} \phi & \phi & \phi \\ \phi & I & \phi \\ \phi & \phi & \frac{1}{2}I \end{bmatrix}.$$

Then

$$G = D_1 B_1^{-1} + D_2 B_2^{-1} = \begin{bmatrix} (D_0 - L_0)^{-1} & \phi & \phi \\ \phi & (D_0 - L_0)^{-1} & \phi \\ \frac{1}{2}(D_0 - L_0)^{-2} & \frac{1}{2}(D_0 - L_0)^{-2} & (D_0 - L_0)^{-1} \end{bmatrix}$$

and

$$G^{-1} = \begin{bmatrix} D_0 - L_0 & \phi & \phi \\ \phi & D_0 - L_0 & \phi \\ -\frac{1}{2}I & -\frac{1}{2}I & D_0 - L_0 \end{bmatrix}.$$

Write $H = B^{-1}C$ where $B = G^{-1}$ and $P^TAP = B - C$, and then

$$C = \begin{bmatrix} U_0 & \phi & I \\ \phi & U_0 & I \\ \frac{1}{2}I & \frac{1}{2}I & U_0 \end{bmatrix},$$

$$B^T + C = \begin{bmatrix} D_0 & \phi & \frac{1}{2}I \\ \phi & D_0 & \frac{1}{2}I \\ \frac{1}{2}I & \frac{1}{2}I & D_0 \end{bmatrix}.$$

Since $B^T + C$ is symmetric, irreducibly diagonally dominant and has positive diagonal components, $B^T + C$ is positive definite (see Theorem 2.3.10 of [11]). By the Householder–John theorem we conclude $\rho(H) < 1$ and the parallel algorithm converges.

Remarks. (1) Let the reordering have the splitting $P^TAP = B - C$ where P is the permutation matrix. Then the original matrix has the splitting $A = PBP^T - PCP^T$ and $(PBP^T)^{-1}(PCP^T) = P(B^{-1}C)P^T$. Thus, $\rho(B^{-1}C) = \rho((PBP^T)^{-1}(PCP^T))$ = spectral radius.

(2) If the serial Gauss–Seidel method is used with the new ordering, then for the Gauss–Seidel splitting $P^TAP = B_{GS} - C_{GS}$ we have

$$B_{GS}^T + C_{GS} = \begin{bmatrix} D_0 & \phi & \phi \\ \phi & D_0 & \phi \\ \phi & \phi & D_0 \end{bmatrix}.$$

(3) If the Jacobi method is used with this new order, then for the Jacobi splitting $P^TAP = B_J - C_J$ we have

$$B_J^T + C_J = \begin{bmatrix} D_0 & \phi & I \\ \phi & D_0 & I \\ I & I & D_0 \end{bmatrix}.$$

Later we present Example 3 where for a very simple case a parallel version of the SOR method is given. An explicit condition on the SOR parameter, w, is given so that the algorithm converges. This requires $0 < w \leqq w_0 < 2$; in contrast, the serial SOR algorithm only requires $0 < w < 2$.

4. A multisplitting as a single *P*-regular splitting. In this section, we restrict the forms of the system matrix and the multisplitting so that we can explicitly compute B_k^{-1} and $(\sum_k^K D_k B_k^{-1})^{-1}$. This will allow us to view the multisplitting iteration matrix as an iteration matrix from a single splitting. The form of the matrix is from a reordering and has been described in George and Liu [3].

DEFINITION. An $N \times N$ matrix A is in *dissection form* if and only if

$$A = \begin{pmatrix} A_1 & & & C \\ & \ddots & & \\ & & A_K & \\ C^T & & & A_0 \end{pmatrix} \tag{5}$$

where A_k are $n_k \times n_k$ matrices for $1 \leqq k \leqq K$, A_0 is $m \times m$ matrix with $n_1 + \cdots + n_k + m = N$, and C is $(N - m) \times m$ matrix. If A_0 is also in dissection form, then A is said to be in *nested dissection form*.

Example. Consider $-\Delta u = f$ and discretize it via the five-point finite-difference method. Let the unknowns be a disjoint union of nodes as indicated in Fig. 2. Reorder

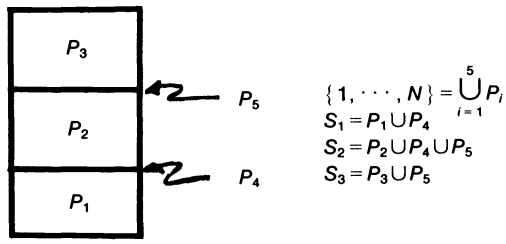


FIG. 2. *Partition of nodes.*

the nodes by listing the two smaller sets of nodes P_4, P_5 last. Then $K = 3$ and

$$A_0 = \begin{pmatrix} A_4 & 0 \\ 0 & A_5 \end{pmatrix}, \qquad C = \begin{pmatrix} C_{14} & 0 \\ C_{24} & C_{25} \\ 0 & C_{35} \end{pmatrix}.$$

Assumed structure and multisplitting of A.

1. Let A have dissection form where $A_0 = \text{diag}\,(A_l)$ where

$$l = K+1, \cdots, 2K-1, \tag{6}$$

$$C = (C_{kl}) k = 1, \cdots, K, \qquad l = K+1, \cdots, 2K-1.$$

2. $A = B_k - C_k$ where $A_k = M_k - N_k$, $A_l = M_l - N_l$ and

$$B_k = \begin{pmatrix} \ddots & & & \\ & M_k & & \\ & \vdots \;\; \ddots & & \\ & -C_{kl}^T \quad M_l & & \\ & \vdots & \ddots & \end{pmatrix}. \tag{7}$$

3. $D_k = \text{diag}\,(d_i^k)$ where d_i^k are uniform on P_l, that is,

$$d_i^k = d_{kl} \quad \text{for all } i \in P_l,$$

$$S_1 = P_1 \cup P_{K+1},$$

$$S_k = P_k \cup P_l \cup P_{l+1}, \qquad l = K+k-1, \tag{8}$$

$$S_K = P_K \cup P_{2K-1},$$

$$d_{kl} = 0 \text{ for } P_l \not\subset S_k \quad \text{and} \quad \sum_{k=1}^{K} d_{kl} = 1.$$

The following lemma will be used repeatedly in the proof of Theorem 3.

LEMMA.

$$\begin{pmatrix} \ddots & & \\ & A_k & \\ & \ddots & \\ \cdots C_k \cdots & & A_0 \end{pmatrix}^{-1} = \begin{pmatrix} \ddots & & \\ & A_k^{-1} & \\ & \ddots & \\ \cdots -\bar{C}_k \cdots & & A_0^{-1} \end{pmatrix}$$

where $\bar{C}_k \equiv A_0^{-1} C_k A_k^{-1}$, A_k^{-1} *exists for* $k = 0, \cdots, K$.

Proof. Use the definition of an inverse of a matrix.

THEOREM 3. *Let A have the dissection form* (5), (6), *the multisplitting* (7), *and the uniform weights* (8). *If M_k^{-1}, M_l^{-1} for $k = 1, \cdots, K$ and $l = K+1, \cdots, 2K-1$ exist, then the multisplitting iteration matrix H in* (4) *may be written as $H = B^{-1}C$ where $A = B - C$ and*

$$B = \begin{pmatrix} \ddots & & & \\ & M_k & & \\ & \vdots & \ddots & \\ \cdots & -d_{kl}C_{kl}^T & \cdots & M_l \\ & \vdots & & & \ddots \end{pmatrix}. \tag{9}$$

Proof. In order to compute B_k^{-1}, use the lemma with $A_k = M_k$ for $k = 1, \cdots, k$, $A_0 = \operatorname{diag}(M_l)$ for $l = K+1, \cdots, 2K-1$, and

$$C_k = -(C_{k,K+1}, C_{k,K+2}, \cdots, C_{k,2K-1})^T.$$

Then $\tilde{C}_k = -(C_{k,K+1}, \cdots, C_{k,2K-1})^T$ where

$$\tilde{C}_{kl}^T = M_l^{-1} C_{kl}^T M_k^{-1}, \quad \text{and} \quad D_k = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & I_k & & & \\ & & & \ddots & & \\ & & & & 0 & \\ & & & & & \ddots \\ & & & & & & d_{kl}I_l \\ & & & & & & & \ddots \end{pmatrix}$$

where I_k identity on P_k.

Thus

$$D_k B_k^{-1} = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & M_k^{-1} & & & \\ & & & \ddots & & \\ & & & & 0 & \\ & & & & & \ddots \\ & & d_{kl}\tilde{C}_{kl}^T & & & d_{kl}M_l^{-1} \\ & & \vdots & & & & \ddots \end{pmatrix}.$$

Since $\sum_{k=1}^{K} d_{kl} = I$,

$$\sum_{k=1}^{K} D_k B_k^{-1} = \begin{pmatrix} \ddots & & & \\ & M_k^{-1} & & \\ & & \ddots & \\ & \vdots & & \\ \cdots & d_{kl}\tilde{C}_{kl}^T & \cdots & d_{kl}M_l^{-1} \\ & \vdots & & & \ddots \end{pmatrix}.$$

Apply the lemma with $A_k = M_k^{-1}$, $A_0 = \operatorname{diag}(M_l^{-1})$ for $l = K+1, \cdots, 2K-1$, $C_k = (d_{k,K+1}\tilde{C}_{k,K+1}, \cdots, d_{k,2K-1}\tilde{C}_{k,2K-1})^T$. Thus

$(d_{k,K+1}C_{k,K+1}, \quad d_{k,2K-1}C_{k,2K-1})$.

$$G^{-1}=\left(\sum_{k=1}^{K} D_k B_k^{-1}\right)^{-1}=\begin{pmatrix} \ddots & & & \\ & M_k & & \\ & & \ddots & \\ \cdots -d_{kl}C_{kl}^T \cdots & & & M_l \\ & & & & \ddots \end{pmatrix}.$$

Since $GA = I - H$, $A = G^{-1} - G^{-1}H = B - C$.

THEOREM 4. *Let A be symmetric positive definite, and have the multisplitting as in* (5)–(8). *Let $A = B - C$, as given by Theorem* 3. *Then*

$$B^T+C=\begin{pmatrix} \ddots & & \vdots & \\ & M_k^T+N_k & \cdots(1-d_{kl})C_{kl}\cdots & \\ & & \ddots & \vdots \\ & SYM & & M_l^T+N_l \\ & & & & \ddots \end{pmatrix}.$$

If $B^T + C$ is positive definite, then the multisplitting algorithm in (4) *is convergent.*

Proof. Since $A = B - C$, $C = B - A$ and $B^T + C = B^T + B - A$. By (7) and (9) we have

$$B^T+C=\begin{pmatrix} \ddots & & \vdots & \\ & M_k^T+M_k & \cdots -d_{kl}C_{kl}\cdots & \\ & & \ddots & \vdots \\ \cdots -d_{kl}C_{kl}^T\cdots & & & M_l^T+M_l \\ & & & & \ddots \end{pmatrix}-\begin{pmatrix} \ddots & & \vdots & \\ & M_k-N_k & \cdots -C_{kl} & \cdots \\ & & \ddots & \vdots \\ \cdots & -C_{kl}^T & \cdots & M_l-N_l \\ & & & & \ddots \end{pmatrix}$$

$$=\begin{pmatrix} \ddots & & \vdots & \\ & M_k^T+N_k & \cdots(1-d_{kl})C_{kl}\cdots & \\ & & \ddots & \vdots \\ \cdots(1-d_{kl})C_{kl}^T\cdots & & & M_l^T+N_l \\ & & & & \ddots \end{pmatrix}.$$

Since A is symmetric and positive definite, the Householder–John theorem yields the desired conclusion when $B^T + C$ is positive definite.

The following example illustrates the condition of $B^T + C$ being positive definite. In this example a multisplitting version of the SOR method is given, and convergence is characterized by the SOR parameter being less than $\omega_0 < 2$. This is a special case of the more general result in the next section.

Example 3. Consider

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

which is positive definite for $2a^2 < 1$, and reorder the nodes by $(1, 2, 3) \to (1, 3, 2)$. Then define two SOR multisplittings:

$$B_1 - C_1 = \begin{bmatrix} 1/w & 0 & 0 \\ 0 & 1/w & 0 \\ -a & 0 & 1/w \end{bmatrix} - \begin{bmatrix} (1-w)/w & a & 0 \\ 0 & (1-w)/w & a \\ 0 & a & (1-w)/w \end{bmatrix},$$

$$B_2 - C_2 = \begin{bmatrix} 1/w & 0 & 0 \\ 0 & 1/w & 0 \\ 0 & -a & 1/w \end{bmatrix} - \begin{bmatrix} (1-w)/w & a & 0 \\ a & (1-w)/w & a \\ 0 & 0 & (1-w)/w \end{bmatrix}.$$

When the following weighting matrices are used:

$$D_1 = \begin{bmatrix} 1 & & \\ & 0 & \\ & & \frac{1}{2} \end{bmatrix} \quad \text{and} \quad D_2 = \begin{bmatrix} 0 & & \\ & 1 & \\ & & \frac{1}{2} \end{bmatrix},$$

then

$$B = (D_1 B_1^{-1} + D_2 B_2^{-1})^{-1}$$

$$= \begin{bmatrix} w & 0 & 0 \\ 0 & w & 0 \\ a/2w^2 & a/2w^2 & w \end{bmatrix}^{-1} = \begin{bmatrix} 1/w & 0 & 0 \\ 0 & 1/w & 0 \\ -a/2 & -a/2 & 1/w \end{bmatrix}.$$

Define C by $A = B - C$ to get

$$C = \begin{bmatrix} (1-w)/w & 0 & a \\ 0 & (1-w)/w & a \\ a/2 & a/2 & (1-w)/w \end{bmatrix}.$$

Then

$$B^T + C = \begin{bmatrix} (2-w)/w & 0 & a/2 \\ 0 & (2-w)/w & a/2 \\ a/2 & a/2 & (2-w)/w \end{bmatrix}.$$

By the definition of positive definite and by completing the square of a quadratic we have that $B^T + C$ is positive definite if and only if

$$0 < w < \frac{\sqrt{22}}{|a| + \sqrt{2}} \leqq 2.$$

It is interesting to compare the constraint on w with the serial SOR and Jacobi SOR constraints on w. For the Jacobi SOR splitting,

$$A = \begin{bmatrix} 1/w & 0 & 0 \\ 0 & 1/w & 0 \\ 0 & 0 & 1/w \end{bmatrix} - \begin{bmatrix} (1-w)/w & 0 & a \\ 0 & (1-w)/w & a \\ a & a & (1-w)/w \end{bmatrix}$$

and

$$B^T + C = \begin{bmatrix} (2-w)/w & 0 & a \\ 0 & (2-w)/w & a \\ a & a & (2-w)/w \end{bmatrix},$$

and this is positive definite if and only if

$$0 < w < \frac{\sqrt{22}}{|2a| + \sqrt{2}}.$$

Since

$$\frac{\sqrt{22}}{|2a| + \sqrt{2}} < \frac{\sqrt{22}}{|a| + \sqrt{2}} < 2,$$

for this example the Jacobi SOR method requires more restriction on the SOR parameter than the parallel SOR method. The serial SOR method requires less restriction on the SOR parameter than the parallel SOR method.

5. Application to parallel SOR. The serial SOR algorithm is given by the single splitting

$$A = \frac{1}{w}(D - wL) - \frac{1}{w}((1-w)D + wL^T).$$

Then $B^T + C = ((2 - w)/w)D$ will be positive definite when $0 < w < 2$ and D has positive diagonal components. The multisplitting version of the SOR algorithm will be more restrictive on w.

PARALLEL SOR ALGORITHM.

$$u_i^{k,n+1/2} = \left(d_i - \sum_{\substack{j<i \\ j \in S_k}} a_{ij} u_j^{k,n+1} - \sum_{\substack{\text{other} \\ j \neq i}} a_{ij} u_j^n \right) \Big/ a_{ii}, \qquad i \in S_k, \tag{10}$$

$$u_i^{k,n+1} = (1-w)u_i^n + w u_i^{k,n+1/2}, \qquad i \in S_k, \tag{11}$$

$$u_i^{n+1} = \sum_{k=1}^{K} d_i^k u_i^{k,n+1}. \tag{12}$$

For $i \in S_k$, (10) and (11) are computed concurrently. Since $d_i^l = 0$ for $i \notin S_k$, only those $u_i^{k,n+1}$ for $i \in S_k$ are needed to compute (12). This algorithm may be written as the following multisplitting:

$$A = \frac{1}{w}(D - wL_k) - \frac{1}{w}((1-w)D + w(L + L^T - L_k)),$$

$$L_k = (-a_{ij}^k),$$

$$a_{ij}^k = \begin{cases} -a_{ij}, & i,j \in S_k \quad \text{and} \quad j < i, \\ 0, & \text{otherwise.} \end{cases}$$

Consider the dissection form as specified by (5)–(8). Let $D^k = \operatorname{diag}(a_{ii})$ where $i \in P_k$ and $\{1, \cdots, N\}$ is the disjoint union of P_k for $k = 1, \cdots, K, K+1, \cdots, 2K-1$. S_k are the overlapping blocks as given in (8). Then (10)–(12) may be written in the form of (7) where

$$\begin{aligned} A_k &= (a_{ij}), \qquad i,j \in P_k \\ &= \frac{1}{w}(D^k - wL^k) - \frac{1}{w}((1-w)D^k + wL^{kT}) \\ &= M_k - N_k \end{aligned} \tag{13}$$

where

$$L^k=(l_{ij}^k), \qquad i,j\in P_k,$$

$$l_{ij}^k=\begin{cases}-a_{ij}, & i,j\in P_k \quad \text{and} \quad j<i,\\ 0 \quad \text{otherwise.}\end{cases}$$

In this case, L_k is a combination of L^k and $-C_{kl}^T$:

$$L_k=\begin{pmatrix} 0 & & & & \\ & L^k & & & \\ & & 0 & \ddots & \\ & \vdots & & & \\ & -C_{kl}^T & & L^l & \\ & \vdots & & & \ddots \end{pmatrix}.$$

Then $M_k^T + N_k = ((2 - w)/w)D^k$ and

$$B^T+C=\begin{pmatrix} \ddots & & & \vdots & \\ & ((2-w)/w)D^k & \cdots & (1-d_{kl})C_{kl} & \cdots \\ & & \ddots\ddots & \vdots & \\ \cdots & (1-d_{kl})C_{kl}^T & \cdots & ((2-w)/w)D^l & \\ & \vdots & & & \ddots \end{pmatrix}. \tag{14}$$

Theorem 4 requires w to be such that $B^T + C$ is positive definite. The next theorem gives conditions that will yield $B^T + C$ positive definite. We further restrict A to be symmetric, have positive diagonal components, be irreducibly diagonally dominant, and hence, positive definite.

THEOREM 5. *Let A be symmetric, have positive diagonal components, and be irreducibly diagonally dominant. Assume that A has dissection form and has the parallel* SOR *multisplitting given by* (5)–(8) *and* (13). *There exists a $w_0 \geqq 1$ such that if $0 < w \leqq w_0$, then $B^T + C$ in* (14) *is positive definite.*

Proof. Since A is irreducibly diagonally dominant, $a_{ii} \geqq \sum_{j\neq i} |a_{ij}|$. Also, $0 \leqq 1 - d_{kl} \leqq 1$. Thus, we may choose $w_{ik} \geqq 1$ such that for $1 \leqq k \leqq K$

$$\frac{2-w_{ik}}{w_{ik}}a_{ii}=\sum_{l>K}\sum_{j\in P_l}(1-d_{kl})|a_{ij}|, \qquad i\in P_k.$$

Also, for $K + 1 \leqq l \leqq 2K - 1$ we may choose $w_{il} \geqq 1$ such that

$$\frac{2-w_{il}}{w_{il}}a_{ii}=\sum_{k\leqq K}\sum_{j\in P_k}(1-d_{kl})|a_{ij}|, \qquad i\in P_l.$$

Define $w_0 = \min\{w_{ik} : i \in P_k, 1 \leqq k \leqq 2K - 1\}$. Since $a_{ii} \geqq \sum_{j\neq i} |a_{ij}|$ and $0 \leqq 1 - d_{kl} \leqq 1$, $B^T + C$ in (14) will be irreducibly diagonally dominant when $0 < w \leqq w_0$. Also, $B^T + C$ is symmetric and has positive diagonal components, and hence, by Theorem 2.3.10 of [10], $B^T + C$ must be positive definite.

6. Numerical experiments. In this section we illustrate Theorem 5 by considering the algebraic system that evolves from an elliptic partial

$$\begin{aligned} -\Delta u &= 10.0 \quad \text{on } \Omega=(0,1)\times(0,1),\\ u &= 0.0 \quad \text{on } \partial\Omega. \end{aligned} \tag{15}$$

This problem was discretized by the five-point finite-difference method and the resulting system was scaled $(D^{-1/2}AD^{-1/2})(D^{1/2}u) = D^{-1/2}d$. In all the calculations $N = (n-1)^2$ and $\Delta x = \Delta y = 1.0/n$. The stopping criteria was the absolute error with $\varepsilon = 0.0001$. Double precision (64 bit reals) was used, and the optimal SOR parameter w_{opt} was estimated by numerical experimentation to within ± 0.005.

All calculations were done on the Alliant FX/8 multiprocessing computer at Argonne National Laboratory. The Alliant FX/8 has eight processors and each has vector instructions. In order to use the vector instructions, in each block the red-black ordering was used. Various compiler directives can be used to control the amount of parallelism:

Alliant FX/8	Compiler directive
$-0g$	optimized serial (1 processor)
$-0gv$	optimized vector (1 processor)
$-0gc$	optimized concurrent
-0	optimized concurrent with vector

Table 1 indicates some calculations where different complier directives were used. $N = 57^2$ was the number of unknowns. In all cases the number of iterations needed for convergence was 114 and $w_{\text{opt}} = 1.900$. In this case $K = 1$ and the red-black order was used. Consequently, the $-0gv$ gave a significant decrease in computing time.

The $-0gc$ directive did not give a significant decrease in time because the problem was not partitioned properly. Table 2 contains calculations for different blocks. In all calculations $N = 57^2$ and the -0 directive was used. Note that the iterations required for convergence increased as K increased. However, if the overlap between blocks is increased, then there is some decrease in iterations required for convergence. Figure 3 explains the configuration of the blocks $K = 4a, 4b, 8b$.

TABLE 1
Variable compiler directives.

Directive	Time (sec)
$-0g$	15.36
$-0gv$	4.56
$-0gc$	10.08
-0	2.77

TABLE 2
Variable blocks.

Blocks	Iterations	w_{opt} (± 0.0005)	Time (sec)
$K = 2$	112	1.920	1.70
$K = 4a$ (overlap = 1 row)	263	1.885	2.71
$K = 4a$ (overlap = 3 row)	202	1.910	2.27
$K = 4a$ (overlap = 5 row)	177	1.920	2.18
$K = 4b$	222	1.905	2.78
$K = 8b$	292	1.885	2.90

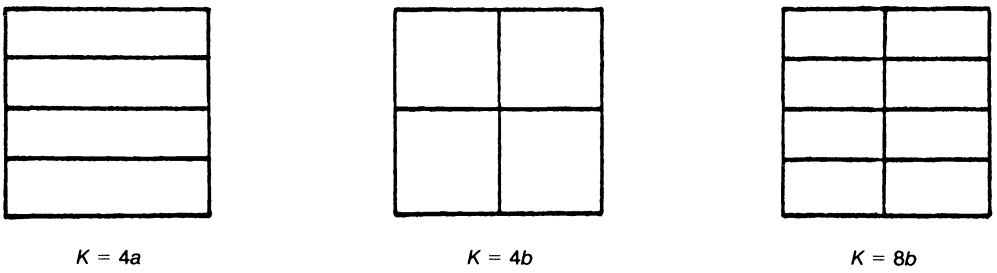


FIG. 3. *Block configurations.*

TABLE 3
Variable N, K = 2 and speedups.

N	Directive	Iterations	w_{opt} (±0.0005)	Time (sec)	Speedup
29^2	$-0g$	68	1.855	2.38	
29^2	-0	59	1.855	.31	7.67
57^2	$-0g$	114	1.900	15.36	
57^2	-0	112	1.920	1.70	9.04
113^2	$-0g$	247	1.950	131.15	
113^2	-0	243	1.960	13.75	9.54
225^2	$-0g$	482	1.975	999.81	
225^2	-0	476	1.980	105.49	9.48

Table 3 measures speedups for different N. Speedup is defined as follows:

$$\text{Speedup} = \frac{\text{time for} - 0 \text{ calculations}}{\text{time for} - 0g \text{ calculations}}.$$

In Table 3 for -0 and $K = 2$ the red-black ordering was used. The speedup for $N = 29^2$ was not as good as the others because the length of the vector pipeline in the Alliant FX/8 is 32 64 bit real numbers. Thus, for $N = 29^2$ with red-black ordering more loads of the pipe were required.

REFERENCES

[1] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[2] L. ELSNER, *Comparisons of weak regular splittings and multisplitting methods*, presented at Third SIAM Conference on Applied Linear Algebra, Madison, WI, May 1988.

[3] A. GEORGE AND J. W. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[4] L. J. HAYES, *A vectorized matrix-vector multiply and overlapping block iterative method*, in Supercomputer Applications, R. W. Numrich, ed., Plenum Press, New York, 1984, pp. 91–100.

[5] A. S. HOUSEHOLDER, *On the convergence of matrix iterations*, Technical Report 1883 Oak Ridge National Laboratory, Oak Ridge, TN, 1953.

[6] F. JOHN, *Advanced Numerical Analysis*, Lecture Notes, Department of Mathematics, New York University, New York, 1956.

[7] O. A. MCBRYAN AND E. F. VAN DE VELDE, *Parallel algorithms for elliptic equations*, Comm. Pure Appl. Math., 38 (1985), pp. 769–795.

[8] M. NEUMANN AND R. J. PLEMMONS, *Convergence of parallel multisplittings and iterative methods for M-matrices*, Linear Algebra Appl., 88/89 (1987), pp. 559–573.

[9] D. P. O'LEARY AND R. E. WHITE, *Multi-splittings of matrices and parallel solution of linear systems*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 630–640.

[10] J. M. ORTEGA AND R. J. PLEMMONS, *Extensions of the Ostrowski–Reich theorem for* SOR *iterations*, Linear Algebra Appl., 28 (1979) pp. 177–191.

[11] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[12] A. M. OSTROWSKI, *Iterative solution of Linear systems of functional equations*, J. Math. Anal. Appl., 2 (1961), pp. 351–369.

[13] F. ROBERT, *Methodes iterative serie parallele*, C.R. Acad. Sci. Paris., A-271 (1970), pp. 847–850.

[14] R. E. WHITE, *A nonlinear parallel algorithm with application to the Stefan problem*, SIAM J. Num. Anal., 23 (1986), pp. 639–652.

[15] ———, *Parallel algorithms for nonlinear problems*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 137–149.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 83–88, January 1990

ON THE PERFORMANCE OF THE MINIMUM DEGREE ORDERING FOR GAUSSIAN ELIMINATION*

PIOTR BERMAN† AND GEORG SCHNITGER†

Abstract. The minimum degree ordering for Gaussian elimination is considered. A way to resolve ties that results in a fill-in of $\theta(n^{\log_3 4})$ for $n \times n$ matrices whose zero/nonzero structure corresponds to a torus graph with an optimal fill-in of $\theta(n \log n)$ is exhibited. Experimental results suggest that random tie resolution yields a similar fill-in.

Key words. sparse matrix computation, Gaussian elimination, minimum degree ordering

AMS(MOS) subject classifications. 05C10, 65F05, 65F50

1. Introduction. We want to consider Gaussian elimination (with no pivoting) for sparse matrices. In particular, we are interested in the amount of fill-in introduced by a particular sequence of pivots, since the fill-in determines the number of arithmetic operations and also the required storage.

In [6] Parter observed that Gaussian elimination can be interpreted graph theoretically. He considered linear systems of equations that have sparse symmetric square coefficient matrices with a nonzero diagonal. Let M be such a matrix with n rows and columns. By replacing every nonzero entry of M by 1, we obtain the adjacency matrix of an undirected graph $G(M)$. Parter showed that eliminating a variable in a system of equations corresponds to eliminating the corresponding vertex in $G(M)$. To eliminate a vertex v from $G(M)$, we first add edges to interconnect any two neighbors of v, and then we remove v with all its incident edges. The number of edges introduced during a sequence of such transformations corresponds to the fill-in of M induced by the corresponding sequence of pivots. This graph-theoretic approach was developed further by Rose [7].

Both Parter and Rose considered cases where one can order vertices to be eliminated in a way that minimizes the resulting fill-in. However, for an arbitrary graph the problem of finding a sequence of vertices minimizing the fill-in is probably intractable (since in [9] a language version of this problem was shown to be NP-complete). Therefore it is appropriate to consider heuristics that hopefully provide an acceptable level of fill-in. We investigate the widely used heuristic of "minimum degree ordering" (MINDEG) introduced in [8] (for a more comprehensive exposition see [1] or [3]). Here, each time we eliminate a vertex of minimum degree. Since the heuristic itself has to be efficient, it would be nice if we could break ties arbitrarily. In [1, p. 137] experimental data is presented which suggest that some tie-breaking strategies are ineffective. Our analysis confirms this conclusion.

We analyze MINDEG on the torus T_k with $n = k^2$ vertices. We obtain T_k when we identify corresponding vertices on parallel sides of the $(k + 1) \times (k + 1)$ mesh. Our result is the following theorem.

THEOREM. *Assume $n = k^2$ where $k/4$ is a power of 3. Then there exists a tie-breaking strategy for T_k with a fill-in larger than $n^{\log_3 4}$ and the number of induced arithmetic operations larger than $n^{1.5\log_3 4}$.*

* Received by the editors June 10, 1987; accepted for publication (in revised form) May 9, 1989. The support of National Science Foundation grant DCR 84-07256, Air Force Office of Scientific Research contract 87-0400, and Office of Naval Research contract N0014-80-0517 is gratefully acknowledged.

† Department of Computer Science, Pennsylvania State University, University Park, Pennsylvania 16802 (berman@shire.cs.psu.edu and georg@shire.cs.psu.edu).

One should note that the optimal tie-breaking strategy for T_k yields the fill-in of $\theta(n \log n)$ and the induced number of operations $\theta(n^{1.5})$ (see [5]). Thus the ratio between the worst and best possible fill-in obtainable by minimum degree orderings for a graph of size n can be as bad as $n^{0.26}$ ($n^{0.39}$ for the number of arithmetic operations). Previously, only constant lower bounds for these ratios were known. Determining the worst possible performance of the minimum degree ordering for the torus graph remains an interesting open problem.

A similar result can be proven when we consider a square mesh instead of a torus; however, the proof becomes quite tedious in this case. We also experimentally studied random tie-breaking strategies and found that the resulting fill-in was approximately the same as the one implied by the strategy described in this paper. Therefore we expect that most tie-breaking strategies are far from optimal. On the other hand, it should be remarked that the rate of growth of $n^{0.26}$ is initially moderate. This could explain why MINDEG behaves satisfactorily in practice.

2. The construction. We first introduce some terminology. We consider edges of an undirected graph to be two-element sets of vertices; elements of an edge are called *neighbors*; the *degree* of a vertex is the number of its neighbors.

DEFINITION. (1) Given an undirected graph G and a vertex v, we define the elimination graph G_v as the result of the following transformation:

(a) add edges to G to interconnect any two neighbors of v in G;

(b) delete v and all its incident edges.

(2) Given a sequence $\vec{v}$ of distinct vertices of G, we define $G_{\vec{v}}$ inductively:

$$G_{\vec{v}} = \begin{cases} G & \text{if } \vec{v} \text{ is the empty sequence} \\ (G_{\vec{u}})_w & \text{if } \vec{v} = (\vec{u}, w). \end{cases}$$

(3) We say that a sequence of vertices $(v_1, \cdots, v_m)$ is a *minimum degree sequence* if and only if v_{i+1} is a vertex of minimum degree in $G_{v_1, \cdots, v_i}$ for $0 \leqq i < m$.

Assume that $k/4$ is a power of 3. To prove the theorem, we will construct a minimum degree sequence $\vec{v}$ such that $(T_k)_{\vec{v}}$ is a complete graph with more than $1.5k^{\log_3 4}$ vertices. This will suffice, because for a complete graph with N nodes the number of edges is $\binom{N}{2}$ and the induced number of operations is at least N^3.

The vertices of T_k correspond to the integer points on the plane where the points (x, y), $(x + k, y)$, and $(x, y + k)$ are identified. All edges have the form $\{(x, y), (x + 1, y)\}$ and $\{(x, y), (x, y + 1)\}$.

Let $k = 2lm$. We define the (l, k)-brick graph B that will be most important for our construction.

Consider the rectangle (or "brick") with corners $(0, l)$, $(l, 0)$, $(3l, 2l)$, and $(2l, 3l)$ on the torus T_k. The translations of this rectangle by the vectors $(0, 2l)$ and $(2l, 0)$ will cover T_k with m^2 copies of the original rectangle. (The resulting cover resembles a slanted wall of bricks.) We label bricks with pairs (i, j): the "first" brick is labeled $(0, 0)$; this brick, when translated by $(2li, 2lj)$, is labeled (ij).

The vertices of B are exactly the vertices of T_k (i.e., the points with integer coordinates) that lie on the boundaries of the rectangles. The edges of B connect vertices belonging to the same rectangle. (Thus each rectangle is completely interconnected.)

Note that this graph was defined by covering T_k with polygons. In general, for any set of polygons that cover T_k, we can define a graph with exactly those vertices of T_k that lie on the boundaries of the polygons, while the edges connect vertices belonging to the same polygon. Such polygons provide a pictorial representation of a graph, which we will call (as in [2]) the finite element graph representation.

LEMMA 1. *Assume that* 4 *divides* k. *Then there exists a minimum degree sequence that transforms the torus* T_k *into the* $(2, k)$*-brick graph.*

Proof. We say that a vertex is *odd* if the sum of its coordinates is *odd*. Let $\vec{u}$ be a sequence of all odd vertices of T_k. Because no two *odd* vertices are adjacent, $\vec{u}$ is a minimum degree sequence. We eliminate all *odd* vertices to obtain the graph $G_1 = (T_k)_{\vec{u}}$ (Fig. 1(a) shows the finite element representation of G_1). Note that all vertices of G_1 have degree 8.

Next we form the sequence $\vec{v}$ of all vertices (i, j) with i, j even and $i + j$ divisible by 4. Again, no two such vertices are adjacent and $\vec{v}$ is a minimum degree sequence. Therefore we can obtain $G_2 = (G_1)_{\vec{v}}$. The polygons of the finite element representation of G_2 (see Fig. 1(b)) are translations of the square with corners $(0, 2)$, $(2, 0)$, $(4, 2)$, and $(2, 4)$. Observe that each corner of such a square is shared by exactly four squares. Thus the corners of each square have degree 20, while all other vertices have degree 12.

Finally, we form a sequence $\vec{w}$ of all vertices of the form $(4i + 3, 4j + 3)$. All of them have degree 12, and no two are adjacent; therefore $\vec{w}$ is a minimum degree sequence. Observe that $(G_2)_{\vec{w}}$ is the desired $(2, k)$-brick graph (see Fig. 1(c)). □

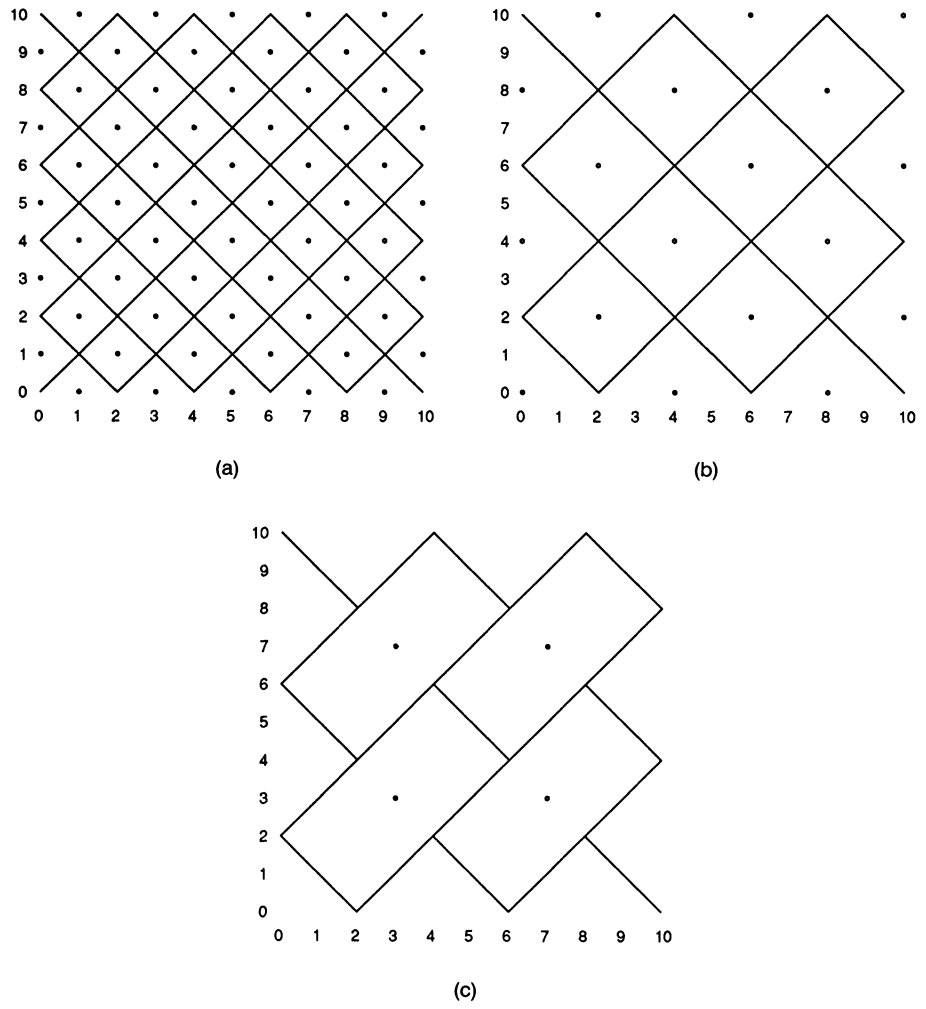


FIG. 1. *Circles denote previously eliminated vertices.*

Next we show how to increase the size of the bricks without eliminating too many vertices.

LEMMA 2. *Assume that* $6l$ *divides* k. *Then there exists a minimum degree sequence that transforms the* (l, k)*-brick graph into a graph isomorphic to the* $(4l, 4k/3)$*-brick graph*.

Proof. We will merge adjacent bricks so that the number of "bricks" decreases by a factor of 9, while the circumference (and so the number of adjacent vertices) of a "brick" increases by a factor of 4. Before describing the process of merging bricks, we will explain the evolution of graphs with finite element representation.

In the graphs considered below a vertex has minimum degree only if it belongs to exactly two polygons of the representation. Consider a vertex v that has (minimum) degree m and that belongs to polygons P_1 and P_2. Let C be the set of vertices that belong only to P_1 and P_2. The neighbors of v are the vertices that belong to P_1 or P_2. Thus all elements of C have the same minimum degree m. Moreover, if $\vec{v}$ is a sequence of i distinct elements from C, then the degree in $G_{\vec{v}}$ of the remaining elements of C is $m - i$, which must be minimum: elements of C do not gain any new neighbors in this transformation. As a result, any ordering $\vec{w}$ of elements of C is a minimum degree sequence. (Here a collection of minimum degree vertices is eliminated simultaneously; George and Liu [4] call this process *mass elimination*.)

One can see that $G_{\vec{w}}$ has the same finite element representation with the exception that polygons P_1 and P_2 are replaced by a single polygon, $P_1 \cup P_2$.

Now we are ready to prove the lemma. First we partition the set of bricks into triples, using the labeling of bricks introduced in the definition of brick graphs:

$$\{(0, 0), (0, 1), (1, 1)\}, \{(1, 2), (1, 3), (2, 3)\}, \{(2, 1), (2, 2), (3, 2)\}$$

belong to the partition;

if a triple belongs to the partition and we add $(0, 3)$ (or $(3, 0)$) to every label, then it still belongs to the partition.

Next in each triple we merge a pair of bricks to obtain a "long rectangle." For example, in the triple $\{(0, 0), (0, 1), (1, 1)\}$ we merge bricks $(0, 0)$ and $(1, 1)$. The remaining original bricks, one in every triple, are pairwise disjoint (see Fig. 2(a)). Because of the repetitiveness of the brick arrangement, every pair of adjacent bricks contains a vertex of minimum degree, thus we can perform these mergers via mass elimination.

Subsequently for each triple we merge the "long rectangle" with the remaining brick. The resulting polygons, which we call "hats," are shown on Fig. 2(b). Note that we had a choice: merge a pair of "long rectangles"; merge a "long rectangle" with a brick adjacent to its short edge; and merge a "long rectangle" with a brick adjacent to its long edge. The degrees of the first vertices eliminated in these mergers are, respectively, $19l - 2$, $15l - 2$, and $14l - 2$. Hence we can create the "hats" using a minimum degree sequence.

Note that the system of hats has similar symmetries as the system of bricks: each hat is adjacent to six other hats, and the pairs of adjacent hats share the same number, $2l + 1$, of vertices. (This property allows us to interpret the system of hats as a hexagonal mesh.) Therefore we can repeat the entire process once more with hats: we group them in triplets, etc. Two stages of this process are shown on Figs. 2(c)–2(d). Let us call the resulting polygons "neo-bricks."

Observe that we reached our goal. Before, all bricks were derived from one via translations by vectors $(2l, 0)$ and $(0, 2l)$. Now, all neo-bricks can be derived from one via translations by vectors $(6l, 0)$ and $(0, 6l)$. We label the neo-brick containing brick

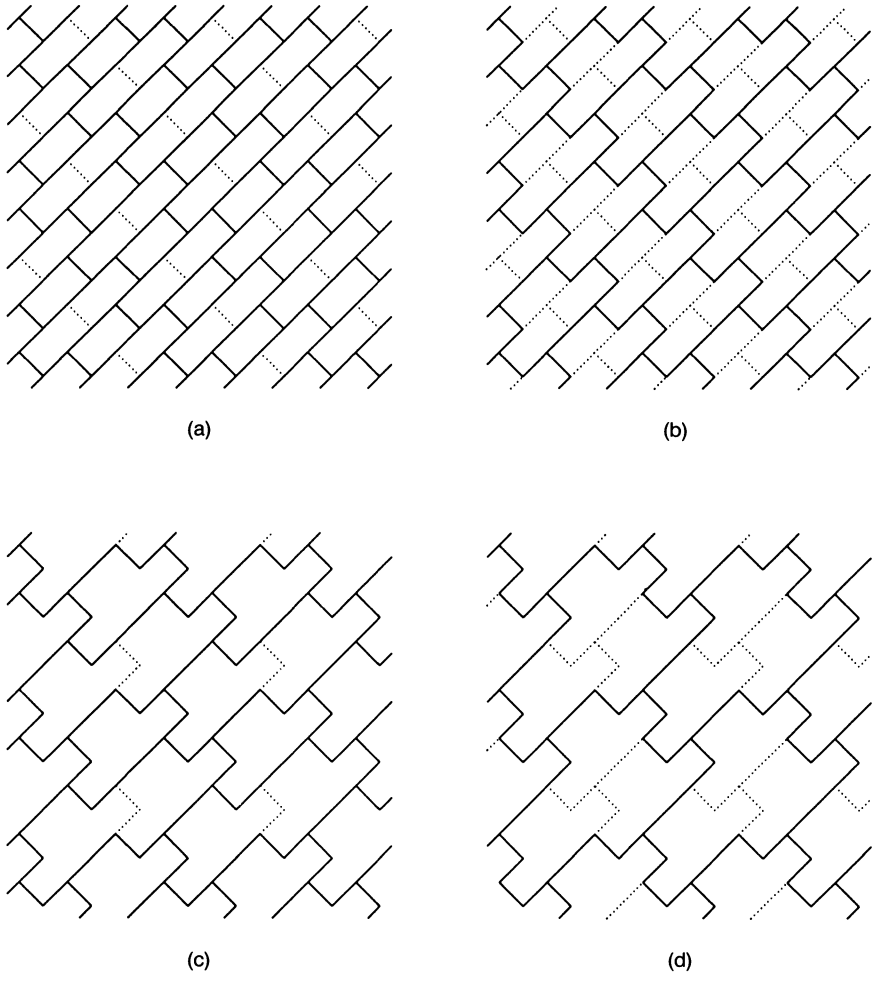


FIG. 2

(0, 0) with (0, 0), while the translation of this neo-brick by the vector $(6li, 6lj)$ is labeled with (i, j). Observe that

(a) neo-bricks (i, j), $(i + m/3, j)$, and $(i, j + m/3)$ are the same;

(b) neo-brick (i, j) shares $4l + 1$ vertices with bricks $(i, j + 1)$, $(i + 1, j + 1)$, $(i - 1, j)$, $(i + 1, j)$, $(i - 1, j - 1)$, $(i, j - 1)$, while its intersection with other neobricks is empty;

(c) three neo-bricks share one vertex if and only if the intersection of any two of them is not empty; the intersection of any four neo-bricks is empty.

Therefore the resulting graph is isomorphic to the $(4l, 4k/3)$-brick graph. □

Proof of the Theorem. Let $k = 3^{m+1}4$. With Lemma 1, we transform T_k into a $(2, k)$-brick graph. By repeatedly applying Lemma 2, we obtain a $(\lambda, 6\lambda)$-brick graph for $\lambda = 4^m 2$. This graph contains 9 bricks in its finite element representation (recall that the $(\lambda, 2\lambda\mu)$-brick graph contains μ^2 bricks). The bricks in this graph are labeled with pairs (i, j) such that $0 \leqq i, j < 3$. We initially merge bricks (0, 0) and (1, 1), (1, 2) with (2, 0), and (0, 2) with (1, 0). Then we merge the first result with (2, 2), the second

TABLE 1
($n = k^2$).

k	Fill-in	(Fill-in)/($n \log_2 n$)	(Fill-in)/$n^{\log_3 4}$	$\frac{\text{(Fill-in)}}{n^2}$
8	353	0.91927	1.85622	0.08618
16	2695	1.31592	2.46433	0.04112
32	17133	1.67314	2.72432	0.01634
64	104400	2.12402	2.88675	0.00622
128	580116	2.52911	2.78939	0.00216

with (0, 1), and the third with (1, 0). At this point the graph becomes fully interconnected and contains $18l$ vertices.

Note that

$$18l = 4^m 36 = 36(k/12)^{\log_3 4} = [36/(3^{\log_3 4} 4^{\log_3 4})]k^{\log_3 4} = (9/4^{\log_3 4})k^{\log_3 4} > 1.565 n^{0.5\log_3 4}.$$

Since the number of edges in a completely interconnected graph with N vertices is $\binom{N}{2}$, the resulting fill-in is larger than $1.22 n^{\log_3 4}$. Similarly, for the completely interconnected graph with N vertices, the implied number of multiplications is N^3 (see [1]), which means that this strategy yields the implied number of operations larger than $n^{1.5\log_3 4}$. □

3. Experimental results. We ran MINDEG for the $k \times k$ torus graph resolving ties at random, using the linear congruential random number generator. For each value of k at least 10 trials were performed. The maximal deviation from the average was close to 5 percent each time. The results in Table 1 below suggest that random tie-breaking strategies do not perform better than the strategy discussed in the second paragraph. It is worth mentioning that these results are proportional to the ones described in [1, p. 137].

Acknowledgments. We wish to thank John Gilbert for introducing us to this problem. Thanks also to Alex Pothen for helpful comments.

REFERENCES

[1] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, England, 1986.

[2] J. A. GEORGE AND D. R. MCINTYRE, *On the application of the minimum degree algorithm to finite element systems*, SIAM J. Numer. Anal., 15 (1978), pp. 90–112.

[3] J. A. GEORGE AND J. W. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[4] ———, *The evolution of the minimum degree ordering algorithm*, Tech. Report ORNL/TM-10452, Oak Ridge National Laboratory, Oak Ridge, TN, 1987.

[5] R. J. LIPTON, D. J. ROSE, AND R. E. TARJAN, *Generalized nested dissection*, SIAM J. Numer. Anal., 16 (1979), pp. 346–358.

[6] S. PARTER, *The use of linear graphs in Gauss elimination*, SIAM Rev., 3 (1961), pp. 119–130.

[7] D. ROSE, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. Read, ed., Academic Press, New York, 1972, pp. 183–217.

[8] W. F. TINNEY AND J. W. WALKER, *Direct solutions of sparse network equations by optimally ordered triangular factorization*, Proc. IEEE, 55 (1967), pp. 1801–1809.

[9] M. YANNAKAKIS, *Computing the minimum fill-in is* NP-*complete*, SIAM J. Algebraic Discrete Methods, 2 (1981), pp. 77–79.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 89–106, January 1990

006

BLOCK METHODS FOR THE SOLUTION OF LINEAR INTERVAL EQUATIONS*

JÜRGEN GARLOFF†

Dedicated to Professor Dr. Karl Nickel on the occasion of his 65th birthday and his retirement from the University of Freiburg i.Br.

Abstract. Feasibility results are generalized for the interval arithmetic versions of Gaussian elimination and of total-step, single-step, and symmetric single-step methods to block methods. It is shown that block Gaussian elimination is always feasible for H-matrices and for a new class of interval matrices. Convergence results for the block iterative methods are given and the quality of the enclosure and the speed of convergence are compared with respect to the fineness of the partition into blocks of the given matrix.

Key words. linear interval equations, block Gaussian elimination, block total-step method, block single-step method, block symmetric single-step method, M-matrix, H-matrix

AMS(MOS) subject classifications. 65G10, 65F05, 65F10

1. Introduction. In this paper we consider the problem of finding good interval enclosures for the set

$$\Sigma(A,B) := \{ \tilde{A}^{-1}\tilde{B} \mid \tilde{A} \in A, \tilde{B} \in B \}$$

of solutions $\tilde{X}$, $\tilde{A}\tilde{X} = \tilde{B}$, where $\tilde{A}$ is varying in the interval matrix $A = [\underline{A}, \bar{A}]$ and B is varying in the interval vector $B = [\underline{B}, \bar{B}]$. The best possible interval enclosure is the hull of $\Sigma(A, B)$, i.e., the intersection of all interval vectors containing $\Sigma(A, B)$. A survey of methods available to determine the hull of $\Sigma(A, B)$ or an outer approximation to it recently has been given by Neumaier [14].

Our paper differs from most papers considering direct and iterative methods for enclosing $\Sigma(A, B)$ in that we are investigating block methods, i.e., we are working with submatrices of the given matrix rather than with single entries. As we shall show, block methods may give an improvement on nonblock methods, i.e., they may result in an interval vector of smaller radius.

Linear systems whose coefficient matrices have a natural block structure often appear in the numerical solution of matrix equations arising from finite-difference approximations to partial differential equations [25]. Typically, the coefficient matrices are block tridiagonal matrices. Here the number of arithmetical operations required can be considerably reduced by using block methods (see, e.g., [8, § 2.3.3]).

To the best of our knowledge, interval methods for block systems have been considered in the literature only in three cases: Valenca [24] solves a tridiagonal block system arising from using multiple shooting to find bounds for the solution of a two-point boundary value problem; however, she does not give any conditions for solvability of such a system by interval methods. Tost [22], [23] presents a method for solving systems of interval equations arising from discretizations of the Laplace equation. Schwandt [20], [21] applies an interval variant of the Buneman algorithm to solve systems of interval equations related to nonlinear Dirichlet problems. In the last two methods (as in [6] for methods for solving interval Toeplitz systems of equations) a substantial re-

* Received by the editors June 16, 1986; accepted for publication (in revised form) March 23, 1989.

† Institut für Angewandte Mathematik, Universität Freiburg i.Br., Hermann-Herder-Strasse 10, D-7800 Freiburg, Federal Republic of Germany. Present address, Fachhochschule für Technik Esslingen, Aussenstelle Göppingen, Robert-Bosch-Strasse 6, D-7320 Göppingen, Federal Republic of Germany.

duction both in the total computing effort and in coefficient storage is obtained by exploiting the special structure of the matrices involved.

In the next section of this paper we recall some results from interval mathematics and from the theory of M-matrices. In § 3 we consider an interval variant of block Gaussian elimination and show that this direct method is always feasible for H-matrices and for a new class of interval matrices. In § 4 we use the concept of sublinear maps due to Neumaier [12] to prove some convergence results for the block total-step, single-step, and symmetric single-step methods. It turns out that for M-matrices a partition with smaller block size results in at least as good enclosures. On the other hand, a partition with smaller block size gives possibly larger spectral radii of some matrices connected with the iteration processes that indicates slower convergence. So roughly speaking, a finer partition yields for M-matrices a better enclosure at the expense of slower convergence.

2. Preliminaries. We denote by $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ the set of real numbers, real n-dimensional (column) vectors, and real $n \times m$ matrices, respectively. To avoid in the sequel parallel definitions for real numbers, vectors, and matrices we identify 1×1 matrices with real numbers and $n \times 1$ matrices with vectors. We consider $\mathbb{R}^{n\times m}$ endowed with the natural (componentwise) partial ordering $\leqq$. Compact, nonempty intervals of $\mathbb{R}^{n\times m}$ with respect to this partial ordering

$$A := [\underline{A}, \bar{A}] = \{\tilde{A} \in \mathbb{R}^{n\times m} \mid \underline{A} \leqq \tilde{A} \leqq \bar{A}\}$$

are usually referred to as *interval matrices* or *matrix intervals*. By $\mathbb{IR}^{n\times m}$ we denote the set of all interval matrices. If $\underline{A} = \bar{A}$ we call $A \in \mathbb{IR}^{n\times m}$ *thin* and identify the set of thin interval matrices with $\mathbb{R}^{n\times m}$. In this paper, we simply refer to elements of $\mathbb{IR}^{n\times m}$ as *matrices*. The entries of a matrix $A \in \mathbb{IR}^{n\times m}$ are written as a_{ij}, $i = 1, \cdots, n, j = 1, \cdots, m$. Transposition is denoted by a superscript T.

We denote the (real) identity matrix by I and by $E^{(i)}$ its ith column vector. The order of I will always be clear from the context. The *absolute value* $|A|$ and *radius* $r(A)$ of a matrix $A \in \mathbb{IR}^{n\times m}$ are given by the real matrices

$$|A| := \max\{|\tilde{A}| \mid \tilde{A} \in A\}, \qquad r(A) := \tfrac{1}{2}(\bar{A} - \underline{A}).$$

The *distance* $q(A, B) \in \mathbb{R}^{n\times m}$ of $A, B \in \mathbb{IR}^{n\times m}$ is defined as

$$q(A,B) := \sup\{|\underline{A} - \underline{B}|, |\bar{A} - \bar{B}|\}.$$

The *Ostrowski-operator* $\langle\cdot\rangle : \mathbb{IR}^{n\times n} \to \mathbb{R}^{n\times n}$ is defined as follows. In the one-dimensional case, we have

$$\langle a\rangle := \min\{|\tilde{a}| \mid \tilde{a} \in a\},$$

or, explicitly, $\langle a\rangle = \min\{|\underline{a}|, |\bar{a}|\}$ if $0 \notin a$, and $\langle a\rangle = 0$ otherwise. For $n > 1$ this operator is defined by

$$\langle A\rangle_{ii} := \langle a_{ii}\rangle, \quad \langle A\rangle_{ij} := -|a_{ij}| \quad \text{for } i \neq j.$$

If P is a bounded, nonempty subset of $\mathbb{R}^{n\times m}$, we denote by

$$\square P := [\inf P, \sup P]$$

the *interval hull* of P.

The interval arithmetical operations are defined as usual (see [2, Chaps. 1, 10] for a more detailed introduction and summary of properties). If $A, B \in \mathbb{IR}^{n\times n}$, then the *sum*

and *difference* of A and B are defined as

$$A \pm B := \{ \tilde{A} \pm \tilde{B} \mid \tilde{A} \in A, \tilde{B} \in B \} = \begin{cases} [\underline{A} + \underline{B}, \bar{A} + \bar{B}] & \text{for "+"}, \\ [\underline{A} - \bar{B}, \bar{A} - \underline{B}] & \text{for "−"}. \end{cases}$$

If $A \in \mathbb{IR}^{n \times m}$, $B \in \mathbb{IR}^{m \times p}$, then the *product* of A and B is defined as

$$A \cdot B := AB := \square \{ \tilde{A}\tilde{B} \mid \tilde{A} \in A, \tilde{B} \in B \}.$$

If $n = m = p = 1$, we have

$$ab = [\min \{ \underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b} \}, \max \{ \underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b} \}],$$

and otherwise AB can be calculated componentwise by

$$(AB)_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}, \qquad i = 1, \cdots, n, \quad j = 1, \cdots, p.$$

We call a matrix $A \in \mathbb{IR}^{n \times m}$ *O-symmetric*, if $\underline{A} = -\bar{A}$. If $a \in \mathbb{IR}$ and $A \in \mathbb{IR}^{n \times m}$ we define *scalar multiplication* by

$$(aA)_{ij} := a a_{ij}, \qquad i = 1, \cdots, n, \quad j = 1, \cdots, m.$$

Writing $\varepsilon := [-1, 1]$, we may represent O-symmetric matrices A as $\varepsilon |A|$ and have for O-symmetric matrices A, B : $A \pm B = \varepsilon(|A| + |B|)$. Furthermore, if $A \in \mathbb{IR}^{n \times m}$ is O-symmetric and $B \in \mathbb{IR}^{m \times p}$, then AB, BA are O-symmetric and $AB = A|B|$, $BA = |B|A$ hold. We call $A \in \mathbb{IR}^{n \times n}$ *regular* if all $\tilde{A} \in A$ are regular. Then the *inverse* of A is defined as

$$A^{-1} := \square \{ \tilde{A}^{-1} \mid \tilde{A} \in A \}.$$

For $n = 1$, if $0 \notin a$, we obtain

$$\frac{1}{a} := a^{-1} = \left[\frac{1}{\bar{a}}, \frac{1}{\underline{a}} \right]$$

and for $b \in \mathbb{IR}$, we define

$$\frac{b}{a} := b \cdot \left(\frac{1}{a} \right).$$

For $n \geqq 2$, A^{-1} can be given explicitly in the following cases:

(i) $n = 2$: if $0 \notin a_{ij}$, $i, j = 1, 2$, then

$$A^{-1} = \begin{pmatrix} (a_{11} - a_{12}\, a_{21}/a_{22})^{-1} & (a_{21} - a_{11}\, a_{22}/a_{12})^{-1} \\ (a_{12} - a_{11}\, a_{22}/a_{21})^{-1} & (a_{22} - a_{12}\, a_{21}/a_{11})^{-1} \end{pmatrix}.$$

Related formulas hold for the other cases.

(ii) A is *diagonal*, i.e., $A = \operatorname{diag}(a_{11}, \cdots, a_{nn})$; then

$$A^{-1} = \operatorname{diag}(1/a_{11}, \cdots, 1/a_{nn}).$$

(iii) A is *inverse-nonnegative*, i.e., $\inf A^{-1} \geqq 0$; then [19] $A^{-1} = [\bar{A}^{-1}, \underline{A}^{-1}]$.

A fundamental property of the interval arithmetical operations is given in the following lemma (see, e.g., [2, p. 6]).

LEMMA 2.1 (inclusion isotonicity). *Let* $* \in \{+, -, \cdot, /\}$, $A * B$ *be defined, and* $A' \subseteq A$, $B' \subseteq B$. *Then* $A' * B'$ *is defined and* $A' * B' \subseteq A * B$ *holds.*

Now we extend the definition of the Schur complement to interval matrices. Let $C \in \mathbb{IR}^{n\times n}$; then the *leading principal submatrix* of order ν ($\nu > 0$) of C is the matrix $(c_{ij})_{i,j=1,\cdots,\nu}$. Now let C be partitioned as

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}, \quad \text{where } C_{11}, C_{22} \text{ are square,} \tag{2.1}$$

and suppose that C_{11} is regular. Then the *Schur complement* of C_{11} in C is defined as

$$(C/C_{11}) := C_{22} - C_{21}C_{11}^{-1}C_{12}.$$

Following [12] and [13], we call a map $S : \mathbb{IR}^n \to \mathbb{IR}^n$ *sublinear* if the following axioms are valid for all $X, Y \in \mathbb{IR}^n$:

$$\text{(S1)} \qquad X \subseteq Y \Rightarrow SX \subseteq SY.$$

$$\text{(S2)} \qquad \alpha \in \mathbb{R} \Rightarrow S(\alpha X) = \alpha(SX).$$

$$\text{(S3)} \qquad S(X+Y) \subseteq SX + SY.$$

The *absolute value* of a sublinear map is the unique nonnegative matrix $|S| \in \mathbb{R}^{n\times n}$ satisfying $SX = |S|X$ for $X = [-E^{(i)}, E^{(i)}]$, $i = 1, \cdots, n$. A sublinear map is called *normal*, if

$$\text{(S4)} \qquad r(SX) \geqq |S| r(X) \quad \text{for all } X \in \mathbb{IR}^n.$$

The relation $S \subseteq T$ between sublinear maps S, T means that $SX \subseteq TX$ for all $X \in \mathbb{IR}^n$.

Examples of sublinear maps [12]. Let $A \in \mathbb{IR}^{n\times n}$.

(1) Multiplication of A by a vector, $A^M : X \to AX$, is sublinear and normal with $|A^M| = |A|$.

(2) If A is regular then the map $A^H : \mathbb{IR}^n \to \mathbb{IR}^n$,

$$A^H X := \square\Sigma(A, X), \tag{2.2}$$

is sublinear with $|A^H| = |A^{-1}|$ and $A^H \subseteq (A^{-1})^M$.

The determination of $A^H X$ is, in general, a difficult problem (see § 1.1 of [14] for a survey of available methods). However, it is easy for $n \leqq 2$ and for general n for an arbitrary regular interval matrix that is thin, diagonal, or inverse-nonnegative (if $\underline{X} \geqq 0$, $\bar{X} \leqq 0$, or $0 \in X$) [4].

(3) The *Gauss inverse* A^G of A is obtained by performing ordinary Gaussian elimination in interval arithmetic (see, e.g., [2, Chap. 15]). If A^G exists then it is a sublinear and normal map with $A^H \subseteq A^G$. If $A \in \mathbb{IR}^{2\times 2}$ and A is regular, then A^G exists [17].

We call $A \in \mathbb{IR}^{n\times n}$ an *M-matrix* if A is inverse-nonnegative and $\bar{a}_{ij} \leqq 0$ for all $i \neq j$, and we call A an *H-matrix* if $\langle A \rangle$ is an M-matrix.

The following lemma gathers some properties of M-matrices (see, e.g., [5, Chap. 6]), we shall frequently use in the sequel (ρ denotes the spectral radius).

LEMMA 2.2. *Let* $A \in \mathbb{R}^{n\times n}$.

(i) *If* $A = \langle A \rangle$ *then the following five conditions* (a)–(e) *are mutually equivalent*:

(a) *A is an M-matrix*;

(b) *The real part of each eigenvalue of A is positive*;

(c) *There exists a vector* $U \in \mathbb{R}^n$, $U > 0$, *such that* $AU > 0$;

(d) *A admits a regular splitting, i.e., it has a representation*

$$A = M - N, \text{ where } M \text{ is regular with } M^{-1} \geqq 0,\ N \geqq 0$$

which is convergent, i.e., $\rho(M^{-1}N) < 1$;

(e) *Each regular splitting of A is convergent.*

(ii) *If A is an M-matrix and $B \in \mathbb{R}^{n\times n}$ with $\langle B\rangle = B$ and $A \leqq B$, then B is also an M-matrix with $B^{-1} \leqq A^{-1}$.*

In the sequel we shall make frequent use of the following lemma.

LEMMA 2.3 [1], [12, Lem. 11, Thm. 4]. *Let $A \in \mathbb{IR}^{n\times n}$ be an H-matrix. Then $|A^{-1}| \leqq \langle A\rangle^{-1}$, and A^G exists with $|A^G| \leqq \langle A\rangle^{-1}$.*

Remark. In Lemma 2.3 equality holds if A is an M-matrix. Equality $|A^G| = \langle A\rangle^{-1}$ follows from $\langle A\rangle = \underline{A}$ and $A^G W = \underline{A}^{-1} W$ for any $W \in \mathbb{IR}^n$ with $0 \in W$ by [3, 4]. For $|A^{-1}| = \langle A\rangle^{-1}$ see [12].

In the sequel we denote by $\|\cdot\|$ a norm on $\mathbb{IR}^{n\times m}$ that is *monotone*, i.e., that satisfies $\|A\| = \||A|\|$ for all A or, equivalently [11], $|A| \leqq |B|$ implies $\|A\| \leqq \|B\|$ for all A, B. We note that [11]

$$A \subseteq B \Rightarrow \|A\| \leqq \|B\| \quad \text{for all } A, B \in \mathbb{IR}^{n\times m}. \tag{2.3}$$

If $n = m$ we assume that the norm is *multiplicative*, i.e., $\|AB\| \leqq \|A\|\,\|B\|$ holds for all $A, B \in \mathbb{IR}^{n\times n}$.

The following lemma extends a well-known result for real matrices to the interval case.

LEMMA 2.4. *If $A \in \mathbb{IR}^{n\times n}$ with $\|A\| < 1$, then $I - A$ is regular and*

$$\|(I-A)^{-1}\| \leqq (1 - \|A\|)^{-1}.$$

Proof. Let $\tilde{A} \in A$; then by (2.3) $\|\tilde{A}\| \leqq \|A\| < 1$ and it follows that $I - \tilde{A}$ is regular, and hence $I - A$ is regular. Since $\rho(|A|) \leqq \||A|\| = \|A\| < 1$ it follows by Lemma 2.2(i), (a) $\Rightarrow$ (b), that the matrix $I - |A|$ is an M-matrix and since $\langle I - A\rangle \geqq I - |A|$ holds it follows by Lemma 2.2(ii) that $\langle I - A\rangle$ is also an M-matrix with $\langle I - A\rangle^{-1} \leqq (I - |A|)^{-1}$. By applying Lemma 2.3 we obtain

$$\|(I-A)^{-1}\| = \||(I-A)^{-1}|\| \leqq \|\langle I-A\rangle^{-1}\| \leqq \|(I-|A|)^{-1}\|.$$

Since the statement is true for the thin matrix $|A|$ the statement for A follows. □

In § 4 we shall use *scaled maximum norms* defined by

$$\|A\|_U := \max_i (\Sigma_k |a_{ik}| u_k)/u_i$$

for some fixed positive vector U.

We note that for $A \in \mathbb{R}^{n\times n}$ with $A \geqq 0$

$$\rho(A) = \inf_{U>0} \|A\|_U. \tag{2.4}$$

In this paper we consider partitions of a matrix $A \in \mathbb{IR}^{n\times n}$

$$A = \begin{pmatrix} A_{11} & A_{12} \cdots A_{1k} \\ A_{21} & A_{22} \cdots A_{2k} \\ \vdots & \vdots \quad \vdots \\ A_{k1} & A_{k2} \cdots A_{kk} \end{pmatrix} \tag{2.5}$$

where the blocks A_{ij} are $n_i \times n_j$-matrices, $n_1 + n_2 + \cdots + n_k = n$; in particular, all diagonal blocks A_{ii} are square. Then

$$\langle A\rangle = \begin{pmatrix} \langle A_{11}\rangle & -|A_{12}| & \cdots & -|A_{1k}| \\ -|A_{21}| & \langle A_{22}\rangle & -|A_{23}| & -|A_{2k}| \\ \vdots & \vdots & \ddots & \vdots \\ -|A_{k1}| & -|A_{k2}| & \cdots & -|A_{k,k-1}| \ \langle A_{kk}\rangle \end{pmatrix}.$$

Let π, π^* be two partitions (2.5) with

$$n = n_1(\pi) + \cdots + n_k(\pi) = n_1(\pi^*) + \cdots + n_{k^*}(\pi^*).$$

Then π^* is called *finer* than π if $k^* \geqq k$ and there are numbers $1 = j_1, j_2, \cdots, j_k$, $j_{k+1} = k^* + 1$ such that

$$n_i(\pi) = n_{j_i}(\pi^*) + n_{j_i+1}(\pi^*) + \cdots + n_{j_{i+1}-1}(\pi^*), \qquad i = 1, \cdots, k,$$

i.e., the diagonal blocks with respect to π^* are diagonal blocks of diagonal blocks with respect to π.

Vectors $X \in \mathbb{IR}^n$ are considered to be partitioned in conformity with (2.5) : $X = (X_1, \cdots, X_k)^T$, where $X_i \in \mathbb{IR}^{n_i}$, $i = 1, \cdots, k$.

3. Block Gaussian elimination. Let $A \in \mathbb{IR}^{n \times n}$ be regular and let $B \in \mathbb{IR}^n$. We want to find an interval vector containing $\Sigma(A, B)$. An obvious way to find such an enclosure is to perform Gaussian elimination in interval arithmetic [2, Chap. 15], resulting in $A^G B$. However, it may fail due to division by an interval containing zero even when A is regular and columns and rows of A may be interchanged [17]. As we have seen in Lemma 2.3, A^G exists if A is an H-matrix.

In this section we first extend interval Gaussian elimination to the block case and show that for an H-matrix block Gaussian elimination is feasible for any partition (2.5) of the matrix. Then we present a further class of interval matrices for which block Gaussian elimination does not fail.

Let the partition (2.5) of A be fixed. We assume that A_{11} is regular. Starting with the formulas

$$\begin{aligned} &A_{1j}^{(1)} := A_{1j}, \quad j = 1, \cdots, k, && B_1^{(1)} := B_1, \\ &A_{ij}^{(1)} := A_{ij} - A_{i1} A_{11}^{-1} A_{1j}, && i, j = 2, \cdots, k, \\ &B_i^{(1)} := B_i - A_{i1} A_{11}^{-1} B_1, && \\ &A_{i1} := 0, && i = 2, \cdots, k, \end{aligned} \tag{3.1}$$

we obtain a new coefficient tableau $A_{ij}^{(1)}$ (partitioned in conformity with (2.5)). By Lemma 2.1 it is easy to see that $\Sigma(A, B) \subseteq \Sigma(A^{(1)}, B^{(1)})$. If $A_{22}^{(1)}$ is regular we proceed in a similar way in transforming $(A_{ij}^{(1)}, B_i^{(1)})_{i,j=2,\cdots,k}$. For convenience, we also renumber the columns and rows that are left unchanged.

After $k-1$ steps, the original coefficient tableau (A, B) is changed to $(A^{(k-1)}, B^{(k-1)})$ with an upper block triangular matrix $A^{(k-1)}$. Obviously, $\Sigma(A, B) \subseteq \Sigma(A^{(k-1)}, B^{(k-1)})$ holds. Now we use the formulas

$$\begin{aligned} &X_k := (A_{kk}^{(k-1)})^{-1} B_k, \\ &X_i := (A_{ii}^{(k-1)})^{-1} \left(B_i^{(k-1)} - \sum_{j=i+1}^{k} A_{ij}^{(k-1)} X_j \right), \qquad i = k-1, \cdots, 1, \end{aligned} \tag{3.2}$$

to obtain $X \in \mathbb{IR}^n$ satisfying $\Sigma(A, B) \subseteq X$. If all diagonal blocks A_{ii} are of order one, then block Gaussian elimination is ordinary (interval) Gaussian elimination. If $k = 1$ then block Gaussian elimination results in $A^{-1}B$.

A variant that avoids the calculation of inverses is to replace in (3.1) $A_{11}^{-1} A_{1j}$ by $A_{11}^G A_{1j}$, i.e., Gaussian elimination applied to the columns of A_{1j}, and to use $(A_{ii}^{(k-1)})^G$ instead of $(A_{ii}^{(k-1)})^{-1}$ in (3.2) (if the Gauss inverses exist).

We say that block Gaussian elimination applied to A is *feasible* (with respect to the partitioning (2.5)) if all $A_{ii}^{(i-1)}$ are regular for $i = 1, \cdots, k$ $(A^{(0)} := A)$.

THEOREM 3.1. *If $A \in \mathbb{IR}^{n\times n}$ is an H-matrix, then block Gaussian elimination is feasible and forming the inverse in* (3.1) *and* (3.2) *may be replaced by forming the Gauss inverses.*

Proof. Since $\langle A_{11}\rangle$ is a principal submatrix of $\langle A\rangle$ it is an M-matrix, and it follows by [1] that A_{11} is nonsingular. Hence, formulas (3.1) can be applied. We now show that $(A_{ij}^{(1)})_{i,j=2,\cdots,k}$ is again an H-matrix. The first statement of the theorem follows then by induction.

By using the rules for O-symmetric intervals we obtain from (3.1) for $i \neq j$ (noting that $\varepsilon^2 = \varepsilon$)

$$\begin{aligned} A_{ij}^{(1)} &\subseteq \varepsilon|A_{ij}| - (\varepsilon|A_{i1}|)A_{11}^{-1}(\varepsilon|A_{1j}|) \\ &= \varepsilon|A_{ij}| - \varepsilon|A_{i1}||A_{11}^{-1}||A_{1j}| \\ &= \varepsilon(|A_{ij}| + |A_{i1}||A_{11}^{-1}||A_{1j}|), \end{aligned}$$

whence by Lemma 2.3

$$A_{ij}^{(1)} \subseteq \varepsilon(|A_{ij}| + |A_{i1}|\langle A_{11}\rangle^{-1}|A_{1j}|). \tag{3.3}$$

For $i = j$ we obtain similarly

$$A_{ii}^{(1)} \subseteq A_{ii} - \varepsilon|A_{i1}|\langle A_{11}\rangle^{-1}|A_{1j}|. \tag{3.4}$$

From (3.3) and (3.4) it follows that (note that $0 \notin a_{ii}$, $i = 1, \cdots, n$)

$$\langle A\rangle^{(1)} \leqq \langle A^{(1)}\rangle \tag{3.5}$$

where $\langle A\rangle^{(1)}$ is the result of applying the first step of block Gaussian elimination to $\langle A\rangle$. It is sufficient to prove that $\langle A\rangle^{(1)}$ is an M-matrix since then by (3.5) and Lemma 2.2(ii) $\langle A^{(1)}\rangle$ is an M-matrix, also. That $\langle A\rangle^{(1)}$ is an M-matrix follows by two facts, namely:

(i) The matrix $(\langle A\rangle^{(1)})_{i,j=2,\cdots,k}$ may be obtained by n_1 steps of ordinary Gaussian elimination. This follows from the close connection of the (real) Schur complement with the process of ordinary Gaussian elimination (see, e.g., [15]).

(ii) If Gaussian elimination is applied to an M-matrix, then all intermediate matrices remain M-matrices (see, e.g., [26]).

The second statement follows similarly by using the enclosure

$$A_{11}^{G}X \subseteq \varepsilon\langle A_{11}\rangle^{-1}|X| \quad \text{for all } X \in \mathbb{IR}^n,$$

a consequence of Lemma 2.3. □

In [3] it has been shown that for M-matrices $A \in \mathbb{IR}^{n\times n}$ Gaussian elimination yields $A^H B = \square\Sigma(A, B)$ under the conditions $\underline{B} \geqq 0$, $\bar{B} \leqq 0$, or $0 \in B$. This result carries over to block Gaussian elimination. The proof uses the fact that for an M-matrix $C \in \mathbb{IR}^{n\times n}$ partitioned as in (2.1) the Schur complement (C/C_{11}) is also an M-matrix and $(C/C_{11}) = [(\underline{C}/\underline{C}_{11}), (\bar{C}/\bar{C}_{11})]$ (for $A_{11} = a_{11}$, see [12, Prop. 6]).

A question that naturally arises is the following. Does block Gaussian elimination give any improvement, i.e., a resulting enclosure of $\Sigma(A, B)$ of smaller radius, on nonblock Gaussian elimination? The answer is that the block method may give an improvement on, but may also be worse, than the nonblock method.

Example 1. Choose

$$A := \begin{pmatrix} 2 & [-1,0] \\ [-1,0] & 2 \end{pmatrix}, \qquad B := \begin{pmatrix} 2 \\ -4 \end{pmatrix}$$

(cf. [13]). Then Gaussian elimination results in

$$A^G B = \begin{pmatrix} [-\frac{1}{3}, 1] \\ [-\frac{8}{3}, -\frac{3}{2}] \end{pmatrix};$$

block Gaussian elimination yields

$$A^{-1}B = \begin{pmatrix} [-\frac{1}{3}, \frac{4}{3}] \\ [-\frac{8}{3}, -\frac{4}{3}] \end{pmatrix} \supsetneq A^G B.$$

Example 2. Choose

$$A := \begin{pmatrix} 4 & [1,2] & [0,1] \\ [-2,-1] & 4 & [-1,0] \\ 0 & [0,1] & 2 \end{pmatrix}, \qquad B := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Then

$$A^G B = \begin{pmatrix} [-\frac{25}{128}, \frac{1}{32}] \\ [-\frac{1}{16}, \frac{1}{8}] \\ [\frac{17}{38}, \frac{17}{32}] \end{pmatrix},$$

which contains properly the vector

$$\begin{pmatrix} [-\frac{53}{289}, 0] \\ [-\frac{1}{17}, \frac{36}{289}] \\ [\frac{17}{38}, \frac{9}{17}] \end{pmatrix},$$

the result of the block method with partition

$$A_{11} = \begin{pmatrix} 4 & [1,2] \\ [-2,-1] & 4 \end{pmatrix}, \qquad A_{22} = 2.$$

Now we present a class of matrices for which ordinary (interval) Gaussian elimination (without row and column interchanges) may fail.

We associate with each pair (i, j), $i, j = 1, \cdots, k$, a monotone norm $\|\cdot\|_{ij}$ on $\mathbb{IR}^{n_i \times n_j}$. These norms may be different for different pairs but must satisfy the following compatibility condition:

$$\|CD\|_{ij} \leqq \|C\|_{il} \|D\|_{lj}, \qquad i,j,l = 1, \cdots, k,$$

for $C \in \mathbb{IR}^{n_i \times n_l}$, $D \in \mathbb{IR}^{n_l \times n_j}$.

LEMMA 3.2 (cf. [18, p. 226]). *Let $A \in \mathbb{R}^{n \times n}$, $A \geqq 0$, be partitioned as in* (2.5) *and let $C \in \mathbb{R}^{k \times k}$ be defined by $c_{ij} := \|A_{ij}\|_{ij}$, $i, j = 1, \cdots, k$. Then $\rho(A) \leqq \rho(C)$.*

Proof. Proceeding similarly as in [7, p. 42], we may find vector norms $\|\cdot\|_i$ on $\mathbb{R}^{n_i}$, $i = 1, \cdots, k$, such that

$$\|DZ\|_i \leqq \|D\|_{ij} \|Z\|_j \quad \text{for all } D \in \mathbb{R}^{n_i \times n_j}, \quad Z \in \mathbb{R}^{n_j}.$$

Denote a Perron vector of A by X,

$$AX = \rho(A)X. \tag{3.6}$$

Taking norms in (3.6), we obtain

$$\rho(A)(\|X_1\|_1, \cdots, \|X_k\|_k)^T \leqq C(\|X_1\|_1, \cdots, \|X_k\|_k)^T.$$

The assertion follows now by using Theorem 1.1 of [5, p. 28]. □

THEOREM 3.3. *Let $A \in \mathbb{IR}^{n\times n}$ be partitioned as in* (2.5) *and let all diagonal blocks A_{ii} be regular. If one of the following two matrices $C_1(A)$, $C_2(A)$ defined by*

$$(C_1(A))_{ij} := \begin{cases} 1 & \text{if } i=j, \\ -\|A_{ii}^{-1}A_{ij}\|_{ij} & \text{if } i\neq j, \end{cases}$$

$$(C_2(A))_{ij} := \begin{cases} 1 & \text{if } i=j, \\ -\|A_{ij}A_{jj}^{-1}\|_{ij} & \text{if } i\neq j \end{cases}$$

is an M-matrix, then block Gaussian elimination is feasible. In particular, A is regular.

Proof. We give the proof only for $C_1(A)$ since the proof for $C_2(A)$ is similar. Let $C_1(A)$ be an M-matrix. Then by Lemma 2.2(i) there exists a vector $U \in \mathbb{R}^k$ with positive entries u_i such that $C_1(A)U > 0$, i.e.,

$$\sum_{\substack{j=1 \\ j\neq i}}^{k} \|A_{ii}^{-1}A_{ij}\|_{ij}u_j < u_i, \qquad i = 1, \cdots, k. \tag{3.7}$$

We now show that $C_1((A_{ij}^{(1)})_{i,j=2,\cdots,k})$ satisfies condition (c) of Lemma 2.2(i) with the positive vector $(u_2, \cdots, u_k)^T$. The statement of the theorem then follows by induction.

First we show that $A_{ii}^{(1)}$ is regular for $i = 2, \cdots, k$. We set $D_{ij} := A_{i1}A_{11}^{-1}A_{1j}$, $i, j = 2, \cdots, k$. Let $A'_{ii} \in A_{ii}^{(1)}$; then $A'_{ii} = \tilde{A}_{ii} - \tilde{D}_{ii}$, where $\tilde{A}_{ii} \in A_{ii}$, $\tilde{D}_{ii} \in D_{ii}$, and

$$A'_{ii} = \tilde{A}_{ii}(I - \tilde{A}_{ii}^{-1}\tilde{D}_{ii}) \in A_{ii}(I - A_{ii}^{-1}A_{i1}A_{11}^{-1}A_{1i}). \tag{3.8}$$

Since by (3.7)

$$\gamma_i := \|A_{ii}^{-1}A_{i1}A_{11}^{-1}A_{1i}\|_{ii} \leqq \|A_{ii}^{-1}A_{i1}\|_{i1}\|A_{11}^{-1}A_{1i}\|_{1i}$$

$$< \frac{u_i}{u_1}\frac{u_1}{u_i} = 1$$

we may apply Lemma 2.4 to obtain that $I - A_{ii}^{-1}D_{ii}$ is regular and

$$\|(I - A_{ii}^{-1}D_{ii})^{-1}\|_{ii} \leqq (1 - \gamma_i)^{-1}. \tag{3.9}$$

It follows from (3.8) that $A_{ii}^{(1)}$ is regular. Since $A'_{ii} \in A_{ii}^{(1)}$ was chosen arbitrarily, we obtain that $(A_{ii}^{(1)})^{-1}$ exists and by (3.8)

$$A'^{-1}_{ii} = (I - \tilde{A}_{ii}^{-1}\tilde{D}_{ii})^{-1}\tilde{A}_{ii}^{-1} \in (I - A_{ii}^{-1}D_{ii})^{-1}A_{ii}^{-1},$$

and hence

$$(A_{ii}^{(1)})^{-1} \subseteq (I - A_{ii}^{-1}D_{ii})^{-1}A_{ii}^{-1}. \tag{3.10}$$

For $i = 2, \cdots, k$ we have by (3.10) and (3.9) (here $\sum'$ means that the running index is not equal to (i)):

$$\sum_{j=2}^{k}{}' \|(A_{ii}^{(1)})^{-1}A_{ij}^{(1)}\|_{ij}u_j$$

$$\leqq \sum_{j=2}^{k}{}' \|(I - A_{ii}^{-1}D_{ii})^{-1}A_{ii}^{-1}(A_{ij} - D_{ij})\|_{ij}u_j$$

$$\leqq \sum_{j=2}^{k}{}' \|(I - A_{ii}^{-1}D_{ii})^{-1}\|_{ii}\|A_{ii}^{-1}(A_{ij} - A_{i1}A_{11}^{-1}A_{1j})\|_{ij}u_j$$

$$\leqq (1-\gamma_i)^{-1} \sum_{j=2}^{k}{}' \| A_{ii}^{-1} A_{ij} - A_{ii}^{-1} A_{i1} A_{11}^{-1} A_{1j} \|_{ij} u_j$$

$$\leqq (1-\gamma_i)^{-1} \left[\sum_{j=2}^{k}{}' \| A_{ii}^{-1} A_{ij} \|_{ij} u_j + \| A_{ii}^{-1} A_{i1} \|_{i1} \sum_{j=2}^{k}{}' \| A_{11}^{-1} A_{1j} \|_{1j} u_j \right].$$

By (3.7) we have

$$\sum_{j=2}^{k}{}' \| A_{11}^{-1} A_{1j} \|_{1j} u_j < u_1 - \| A_{11}^{-1} A_{1i} \|_{1i} u_i$$

so that we may further estimate

$$\sum_{j=2}^{k}{}' \| (A_{ii}^{(1)})^{-1} A_{ij}^{(1)} \|_{ij} u_j$$

$$< (1-\gamma_i)^{-1} \left[\sum_{j=1}^{k}{}' \| A_{ii}^{-1} A_{ij} \|_{ij} u_j - \| A_{ii}^{-1} A_{i1} \|_{i1} \| A_{11}^{-1} A_{1i} \|_{1i} u_i \right]$$

$$< (1-\gamma_i)^{-1} (1 - \| A_{ii}^{-1} A_{i1} \|_{i1} \| A_{11}^{-1} A_{1i} \|_{1i}) u_i \leqq u_i$$

by using (3.7) again. This concludes the proof. □

Remarks. (i) If all diagonal blocks of A are of order one, the norms are the absolute values and Theorem 3.3 reduces to the result obtained in [1] for the case of real intervals (note that for $a \in \mathbb{IR}$ with $0 \notin a$ the property $|a^{-1}|^{-1} = \langle a \rangle$ holds).

(ii) A class closely related to matrices $A \in \mathbb{R}^{n \times n}$ satisfying the assumptions of Theorem 3.3 is the class of quasiblock diagonal dominant matrices [16].

The following corollary gives a more easily verifiable condition for the feasibility of block Gaussian elimination.

COROLLARY 3.4. *Let $A \in \mathbb{IR}^{n \times n}$ be partitioned as in* (2.5) *and let all diagonal blocks A_{ii} be regular. If the matrix $C_3(A)$ defined by*

$$(C_3(A))_{ij} := \begin{cases} \| A_{ii}^{-1} \|_{ii}^{-1} & \text{if } i = j, \\ -\| A_{ij} \|_{ij} & \text{if } i \neq j \end{cases} \tag{3.11}$$

is an M-matrix, then block Gaussian elimination is feasible.

The following theorem relates a subclass of the matrices satisfying the assumptions of Corollary 3.4 to H-matrices.

THEOREM 3.5. *Let $A \in \mathbb{IR}^{n \times n}$, and suppose that $C_3(A)$ (cf.* (3.11)) *is an M-matrix, and let all A_{ii} be M-matrices. Then A is an H-matrix.*

Proof. Let A satisfy the above assumptions. Then $\langle A_{ii} \rangle = \underline{A}_{ii}$, and $\| A_{ii}^{-1} \|_{ii} = \| \underline{A}_{ii}^{-1} \|_{ii}$, $i = 1, \cdots, k$. We will show that the regular splitting of $\langle A \rangle$, $\langle A \rangle = M_1 - N_1$ with $M_1 := \operatorname{diag}(\underline{A}_{11}, \cdots, \underline{A}_{kk})$, $N_1 := M_1 - \langle A \rangle$, is convergent.

Since $C_3(A)$ is an M-matrix, it follows from Lemma 2.2(i) that the regular splitting of $C_3(A)$, $C_3(A) = M_2 - N_2$ with $M_2 := \operatorname{diag}(\| A_{11}^{-1} \|_{11}^{-1}, \cdots, \| A_{kk}^{-1} \|_{kk}^{-1})$, $N_2 := M_2 - C_3(A)$, is convergent. Applying Lemma 3.2, we obtain $\rho(M_1^{-1} N_1) \leqq \rho(M_2^{-1} N_2) < 1$, which completes the proof. □

The following example shows that in Theorem 3.5 the assumption that all A_{ii} are M-matrices cannot be replaced by the assumption that all A_{ii} are H-matrices.

Example 3. Choose

$$A := \left(\begin{array}{cc|cc} 2 & 2 & 1.5 & 0 \\ -1 & 2 & 0 & 0 \\ \hline 0.7 & 0 & 1 & 1 \\ 0 & 0 & -0.5 & 1 \end{array}\right)$$

with the partition marked by dashed lines and choose as norms $\|\cdot\|_{ij}$ the maximum row sum norm. Then

$$C_3(A) = \begin{pmatrix} 1.5 & -1.5 \\ -0.7 & 0.75 \end{pmatrix}$$

is an M-matrix. However, the determinant of the leading principal submatrix of order 3 of $\langle A \rangle$ has a negative value that shows (cf. [5, p. 134]) $\langle A \rangle$ is not an M-matrix.

Suppose that A is not an M-matrix and that some diagonal blocks of A are of size n_i, $n_i \geqq 3$, and inverse-nonnegative. For these diagonal blocks the inverse is given by $A_{ii}^{-1} = [\bar{A}_{ii}^{-1}, \underline{A}_{ii}^{-1}]$. However, in the κth step ($\kappa > 1$) $A_{ii}^{(\kappa)}$ may not be inverse-nonnegative and often will be not sparse and not thin so that it is hard to find $(A_{ii}^{(\kappa)})^{-1}$. This difficulty can be overcome by using the variant of block Gaussian elimination, avoiding the calculation of inverses or by using iterative methods (see the next section). But for some special block band matrices often appearing in practical problems the determination of $(A_{ii}^{(\kappa)})^{-1}$ may be avoided. For example, for tridiagonal block H-matrices we may apply an interval variant of the block method described in [8, § 2.3.3] that computes the solution of $AX = B$ through solutions of linear systems of smaller size and avoids the computation of inverses of submatrices of order greater than one (the proof of feasibility uses the same idea as the proof of Theorem 3.1).

4. Iteration. In this section we study methods for enclosing $\Sigma(A, B)$ iteratively. We begin with block forward and backward substitution. Let $L \in \mathbb{IR}^{n\times n}$ be partitioned as in (2.5) and let L be a regular *lower block triangular matrix*, i.e., L_{ii} is regular for $i = 1, \cdots, k$ and $L_{ij} = 0$ for $j > i$. Then *block forward substitution* is the map $L^F : \mathbb{IR}^n \to \mathbb{IR}^n$ defined as follows: Let $X \in \mathbb{IR}^n$. Then $L^F X = Y$, where

$$Y_i := L_{ii}^G \left(X_i - \sum_{j=1}^{i-1} L_{ij} Y_j \right), \qquad i = 1, \cdots, k. \tag{4.1}$$

If $R \in \mathbb{IR}^{n\times n}$ is a regular *upper block triangular matrix*, i.e., R_{ii} is regular for $i = 1, \cdots, k$ and $R_{ij} = 0$ for $j < i$, then *block backward substitution* is the map $R^F : \mathbb{IR}^n \to \mathbb{IR}^n$ defined by ($X \in \mathbb{IR}^n$, $Y = R^F X$)

$$Y_i := R_{ii}^G \left(X_i - \sum_{j=n}^{i+1} R_{ij} Y_j \right), \qquad i = k, k-1, \cdots, 1. \tag{4.2}$$

Then, as can readily be confirmed, L^F and R^F are sublinear maps. Let $A \in \mathbb{IR}^{n\times n}$ be partitioned as in (2.5). Then we express A as the matrix sum

$$A = L + D + U \tag{4.3}$$

where $D = \operatorname{diag}(A_{11}, A_{22}, \cdots, A_{kk})$, L is a strictly lower block triangular matrix, i.e., $L_{ij} = 0$ for $j \geqq i$, and U is a strictly upper block triangular matrix, i.e., $U_{ij} = 0$ for $j \leqq i$.

THEOREM 4.1. *Let $A \in \mathbb{IR}^{n\times n}$ be splitted as in* (4.3) *and let D^G exist with*

$$\rho(|D^G|(|L|+|U|))<1. \tag{4.4}$$

Then for arbitrary $B \in \mathbb{IR}^n$, the following statements are true:

(i) *The equations*

$$X = D^G(B-(L+U)X), \tag{4.5}$$

$$Y = (D+L)^F(B-UY), \tag{4.6}$$

$$Z = (D+U)^F(B-L((D+L)^F(B-UZ))) \tag{4.7}$$

have unique solutions $X, Y, Z \in \mathbb{IR}^n$ with $X = Y = Z \supseteq A^H B$.

(ii) *The sequences generated by*

$$X^{(l+1)} := D^G(B-(L+U)X^{(l)}) \quad (\textit{block total step method}) \tag{4.8}$$

$$Y^{(l+1)} := (D+L)^F(B-UY^{(l)}) \quad (\textit{block single step method}) \tag{4.9}$$

$$\left.\begin{aligned} Z^{(l+1/2)} &:= (D+L)^F(B-UZ^{(l)}) \\ Z^{(l+1)} &:= (D+U)^F(B-LZ^{(l+1/2)}) \end{aligned}\right\} \quad (\textit{block symmetric single step method}) \tag{4.10}$$

converge to X for all starting vectors $X^{(0)}, Y^{(0)}, Z^{(0)} \in \mathbb{IR}^n$. Setting

$$R := |D^G||L|, \qquad S := |D^G||U|, \tag{4.11}$$

we have

$$\|q(X^{(l+1)},X)\| \leqq \alpha\|q(X^{(l)},X)\| \quad \textit{with } \alpha := \|R+S\|, \tag{4.12}$$

$$\|q(Y^{(l+1)},X)\| \leqq \beta\|q(Y^{(l)},X)\| \quad \textit{with } \beta := \|(I-R)^{-1}S\|, \tag{4.13}$$

$$\|q(Z^{(l+1)},X)\| \leqq \gamma\|q(Z^{(l)},X)\| \tag{4.14}$$

with $\gamma := \|(I-S)^{-1}R(I-R)^{-1}S\|$. If $\|\cdot\|$ is a scaled maximum norm and $\alpha < 1$, then

$$\gamma \leqq \beta \leqq \alpha. \tag{4.15}$$

(iii) *The following relation holds*

$$\rho((I-S)^{-1}R(I-R)^{-1}S) \leqq \rho((I-R)^{-1}S) \leqq \rho(R+S)<1. \tag{4.16}$$

Proof. We consider the block total step method first. Since D^G exists by assumption, D^G is a normal sublinear map. Blockwise multiplication of the matrix $L + U$ by a vector $W = (W_1, \cdots, W_k)^T$ is also sublinear and normal. From (4.4) and [13, Thm. 3.1] it follows that for each $B \in \mathbb{IR}^n$, equation (4.5) has a unique solution $X \in \mathbb{IR}^n$, and that the block total-step iteration (4.8) is convergent with limit X. Moreover, we obtain estimation (4.12) for the block total-step method.

To show $A^H B \subseteq X$ we note that each $\tilde{A} \in A$ may be expressed as a sum of block matrices $\tilde{A} = \tilde{L} + \tilde{D} + \tilde{U}$ with $\tilde{L} \in L$, $\tilde{D} \in D$, $\tilde{U} \in U$. Let $\tilde{X} = \tilde{A}^{-1}\tilde{B}$ with $\tilde{B} \in B$; then $\tilde{B} = \tilde{A}\tilde{X} = (\tilde{L} + \tilde{D} + \tilde{U})\tilde{X}$ and

$$\tilde{X} = \tilde{D}^{-1}(\tilde{B}-(\tilde{L}+\tilde{U})\tilde{X}) \in D^G(B-(L+U)\tilde{X}) \tag{4.17}$$

by the inclusion isotonicity of interval arithmetical operations. If we start (4.8) with $X^{(0)} = \tilde{X}$, we obtain $X^{(0)} \subseteq X^{(1)}$ by (4.17) and by the inclusion isotonicity of the map that maps a vector $W \in \mathbb{IR}^n$ on $D^G(B - (L + U)W)$ we obtain that $\tilde{X} \in X^{(l)}$ for all l, whence $\tilde{X} \in X$. Since $\tilde{X} \in A^H B$ was chosen arbitrarily, $A^H B \subseteq X$ follows.

Now we show (4.15). Let $\|\cdot\|_{U'}$ be a scaled maximum norm and let $\|R+S\|_{U'} < 1$. Then $(I-R-S)U' > 0$ holds, and hence

$$SU' \leqq SU' + R(I-R-S)U' = (I-R)(R+S)U'. \tag{4.18}$$

Since $R \geqq 0$ and $\rho(R) < 1$ it follows that $(I-R)^{-1} \geqq 0$ and (4.18) gives $(I-R)^{-1}SU' \leqq (R+S)U'$, and hence $\beta \leqq \alpha$. Similarly, we obtain $\|(I-S)^{-1}R\|_{U'} \leqq \alpha < 1$. Finally, the inequality $\gamma \leqq \beta$ follows from

$$\gamma \leqq \|(I-S)^{-1}R\| \, \|(I-R)^{-1}S\| \leqq \beta.$$

Part (iii) is a consequence of (2.4) and (4.4).

To prove the statements on the block single-step method and the block symmetric single-step method we proceed first as in [12] to show that the maps $(D+L)^F$ and $(D+R)^F$ are normal. If $Q := (D+L)^F P$, $P \in \mathbb{IR}^n$, then Q satisfies the equation $Q = D^G(P - LQ)$, and hence

$$(D+L)^F P = D^G(P - L(D+L)^F P).$$

Now apply Proposition 3 of [12] (setting therein $R := (D+L)^F$, $S := D^G$, letting T be the map that maps a vector $W \in \mathbb{IR}^n$ on $-LW$, and noting that $\rho(|D^G|\,|L|) < 1$ by (4.4)) to obtain that $(D+L)^F$ is normal and

$$|(D+L)^F| = (I - |D^G|\,|L|)^{-1}|D^G|. \tag{4.19}$$

Similarly, we show that $(D+U)^F$ is normal and

$$|(D+U)^F| = (I - |D^G|\,|U|)^{-1}|D^G| \tag{4.20}$$

holds. Using (4.16), equality (4.19) gives

$$\rho(|(D+L)^F|\,|U|) = \rho((I-R)^{-1}S) < 1,$$

and the statements on the block single-step method follow from Theorem 3.1 of [13].

To prove the statement on the block symmetric single-step method we show a theorem similar to Theorem 3.1 of [13] for the equation $Z = H(W + J(B + KZ))$ and the corresponding iteration, where W, $B \in \mathbb{IR}^n$ and H, J, K are sublinear maps satisfying $\rho(|H|\,|J|\,|K|) < 1$. Applying this theorem and using (4.19), (4.20), and (4.16), the statements on the block symmetric single-step method follow.

It remains to show that the solutions X, Y, Z of (4.5)–(4.7) are equal. The solution Y of (4.6) satisfies $Y = D^G(B - (L+U)Y)$ and the uniqueness of the solution of (4.5) yields $X = Y$. Similarly, we have that X is a solution of $W = (D+U)^G(B - LW)$. Hence X satisfies (4.7) and by the uniqueness of the solution of (4.7) it follows that $X = Z$. □

Remarks. (i) If A is an H-matrix then D is also an H-matrix and by Lemma 2.3 D^G exists. Since $\langle A \rangle = \langle D \rangle - |L| - |U|$ is a regular splitting of the M-matrix $\langle A \rangle$ we obtain by Lemma 2.2 and Lemma 2.3

$$\rho(|D^G|(|L| + |U|)) \leqq \rho(\langle D \rangle^{-1}(|L| + |U|)) < 1.$$

Hence the assumptions of Theorem 4.1 are satisfied if A is an H-matrix and we may estimate by Lemma 2.3

$$\alpha \leqq \|\langle D \rangle^{-1}(|L| + |U|)\|, \tag{4.21}$$

$$\beta \leqq \|(I - \langle D \rangle^{-1}|L|)^{-1}\langle D \rangle^{-1}|U|\|, \tag{4.22}$$

and similarly for γ.

If in (4.1), (4.2), (4.4)–(4.16) we replace D^G by D^{-1}, then Theorem 4.1 remains true. Condition (4.4) then reads

$$\rho(|D^{-1}|(|L|+|U|))<1. \tag{4.4'}$$

Again, (4.4') is satisfied if A is an H-matrix and estimates for α, β, γ similar to (4.21), (4.22) hold.

If A satisfies the assumptions of Corollary 3.4, then by Lemma 3.2 it follows that condition (4.4') is fulfilled.

The point here is that D^{-1} may be replaced by *any* enclosure $\hat{D}$ of D^{-1} as long as

$$\rho(|\hat{D}|(|L|+|U|))<1 \tag{4.4''}$$

is guaranteed.

(ii) For a matrix A condition (4.4) (respectively, (4.4') or (4.4'')) is satisfied if $r(X^{(1)})<r(X^{(0)})$ (see [13, Prop. 3.3]). This condition is satisfied if, e.g., $X^{(1)}$ is contained in the interior of $X^{(0)}$.

(iii) It should be noted that the symmetric single-step method may be carried out in such a manner that the number of interval arithmetical operations is about the same as for the single-step method [2, p. 168]. Here we have neglected the number of operations required for the solution of the linear systems with coefficient matrices D_{ii}. These are in each step $2k$ linear systems for the symmetric single-step method and k systems for the single-step method. However, after having computed $Z^{(1/2)}$ (or $Y^{(1)}$) we know the triangular decompositions of the matrices D_{ii} so that the solution of each of the linear systems reduces to one forward and one backward substitution.

If condition (4.4) is dropped (but for any of the block methods the first iterate is contained in the starting vector) the convergence is monotonic.

LEMMA 4.2. *Let $A\in\mathbb{IR}^{n\times n}$ be split as in* (4.3) *and let D^G exist. Let $\{V^{(l)}\}$ be any of the sequences $\{X^{(l)}\}$, $\{Y^{(l)}\}$, $\{Z^{(l)}\}$ generated by the iterations* (4.8)–(4.10). *Then the following enclosures are true*:

$$\begin{aligned} &\text{If } V^{(1)}\subseteq V^{(0)}, \text{ then it holds that for } l=1,2,\cdots\\ &V\subseteq V^{(l)}\subseteq V^{(l-1)}\subseteq\cdots\subseteq V^{(0)} \end{aligned} \tag{4.23}$$

where $\lim_{l\to\infty} V^{(l)}=V$, *and V is a solution of the corresponding fixed point equation* (4.5)–(4.7).

Proof. The monotonic convergence (4.23) follows by induction using the inclusion isotonicity of the involved maps. That V is a solution of the corresponding fixed point equation follows by the continuity of these maps. □

THEOREM 4.3. *Let $A\in\mathbb{IR}^{n\times n}$ be split as in* (4.3) *and let D^G exist. If*

$$X^{(0)}=Y^{(0)}=Z^{(0)} \quad \text{and} \quad X^{(1)}\subseteq X^{(0)}, \tag{4.24}$$

then the sequences $\{X^{(l)}\}$, $\{Y^{(l)}\}$, $\{Z^{(l)}\}$ converge monotonically to the same limit vector. Specifically, for all l

$$Z^{(l)}\subseteq Y^{(l)}\subseteq X^{(l)}, \tag{4.25}$$

$$X^{(kl)}\subseteq Y^{(l)}, \tag{4.26}$$

$$Y^{(kl)}\subseteq Z^{(l)}. \tag{4.27}$$

Proof. We assume (4.24). By Lemma 4.2, the sequence $\{X^{(l)}\}$ converges monotonically. By an extension of the proof for the (nonblock) total and single-step method in [28] to the block case we show that $Y^{(1)}\subseteq Y^{(0)}$, which implies by Lemma 4.2 that $\{Y^{(l)}\}$ converges monotonically also.

Furthermore, we obtain similarly as in [28] that $Y^{(l)} \subseteq X^{(l)}$ and (4.26) hold for all l and that $Z^{(1/2)} \subseteq Z^{(0)}$.

To show (4.27) and $Z^{(l)} \subseteq Y^{(l)}$ for all l we proceed by induction and assume without loss of generality that $k \geqq 2$. The statements trivially hold for $l = 0$. We assume that $Y^{(kl)} \subseteq Z^{(l)} \subseteq Y^{(l)}$ for some l. Then by the inclusion isotonicity of the map that maps a vector $W \in \mathbb{IR}^n$ on $(D + L)^F(B - UW)$ we obtain

$$Y^{(kl+1)} \subseteq Z^{(l+1/2)} \subseteq Y^{(l+1)}.$$

Hence

$$Z_k^{(l+1)} = D_{kk}^G\left(B_k - \sum_{j=1}^{k-1} A_{kj} Z_j^{(l+1/2)}\right)$$

$$\begin{cases} \supseteq D_{kk}^G\left(B_k - \sum_{j=1}^{k-1} A_{kj} Y_j^{(kl+1)}\right) = Y_k^{(kl+1)} \\ \subseteq D_{kk}^G\left(B_k - \sum_{j=1}^{k-1} A_{kj} Y_j^{(l+1)}\right) = Y_k^{(l+1)}. \end{cases}$$

We assume that for some $k - 1 \leqq i \leqq 1$

$$Y_{i'}^{(kl+k-1)} \subseteq Z_{i'}^{(l+1)} \subseteq Y_{i'}^{(l+1)} \quad \text{for all } i', k \leqq i' \leqq i+1. \tag{4.28}$$

Then by (4.28) and the monotonic decreasing of the sequence $\{Y^{(l)}\}$, we obtain

$$Z_i^{(l+1)} = D_{ii}^G\left(B_i - \sum_{j=1}^{i-1} A_{ij} Z_j^{(l+1/2)} - \sum_{j=i+1}^{k} A_{ij} Z_j^{(l+1)}\right)$$

$$\begin{cases} \supseteq D_{ii}^G\left(B_i - \sum_{j=1}^{i-1} A_{ij} Y_j^{(kl+k)} - \sum_{j=i+1}^{k} A_{ij} Y_j^{(kl+k-1)}\right) \\ \subseteq D_{ii}^G\left(B_i - \sum_{j=1}^{i-1} A_{ij} Y_j^{(l+1)} - \sum_{j=i+1}^{k} A_{ij} Y_j^{(l)}\right) \end{cases}$$

$$\begin{cases} = Y_i^{(kl+k)} \\ = Y_i^{(l+1)} \end{cases}$$

from which $Y^{(k(l+1))} \subseteq Z^{(l+1)} \subseteq Y^{(l+1)}$ follows. In particular,

$$Z^{(1)} \subseteq Y^{(1)} = Z^{(1/2)} \subseteq Z^{(0)},$$

implying monotonic convergence of the sequence $\{Z^{(l)}\}$. The proof is complete. □

Remark. The enclosure (4.26) is *optimal* in the sense that there are matrices A and starting vectors $X^{(0)}$ satisfying (4.24) such that the sequences $\{X^{(l)}\}$ and $\{Y^{(l)}\}$ are different but $X^{(kl)} = Y^{(l)}$ for all l.

An example is provided by choosing $k = n$,

$$A := \begin{pmatrix} 0 & \cdots & 0 & \alpha \\ 1 & \ddots & & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

with $0 < \alpha < 1$ and

$$X^{(0)} := \varepsilon(1, \cdots, 1)^T.$$

Then

$$X^{(nl)} = Y^{(l)} = \alpha^l X^{(0)}.$$

The enclosure (4.27) is nearly optimal: Choosing A^T and the same starting vector $X^{(0)}$ we obtain

$$Z^{(l)} = \alpha^l X^{(0)},$$
$$Y^{((n-1)l)} = \varepsilon\alpha^{l-1}(1, \alpha, \cdots, \alpha)^T,$$
$$Y^{((n-1)l+1)} = \varepsilon\alpha^l(1, \cdots, 1, \alpha)^T.$$

The partition (2.5) has been considered to be fixed so far. Now we compare the limits of the sequences (4.8) for different partitionings. We affix a star to mark quantities that are related to a partition π^*.

THEOREM 4.4. *Let π, π^* be two partitions* (2.5) *and let π^* be finer than π. Let $A \in \mathbb{IR}^{n\times n}$ be an M-matrix and X and X^* be the solutions of equation* (4.5) *with respect to π and π^*, respectively. Then $X^* \subseteq X$ holds and there are M-matrices for which $X^* \neq X$. If $\underline{B} \geqq 0$, $\bar{B} \leqq 0$, or $0 \in B$, then $X^* = X = A^H B$ is valid.*

Proof. As we have already noted, condition (4.4) is satisfied for M-matrices. For each partition (2.5), the iteration (4.8) is a special case of the incomplete LU-factorization of Mayer [9][1] and the inclusion $X^* \subseteq X$ follows from Theorem 3 therein. Now choose π^* as the finest partition, i.e., $D := \operatorname{diag}(a_{11}, \cdots, a_{nn})$, and π as the coarsest partition, i.e., $D := A$. Then π^* gives rise to the total-step method with 1×1 blocks and π to Gaussian elimination. Choose A and B as in Example 1 of § 3; then

$$X^* = \begin{pmatrix} [0, 1] \\ [-2, -\frac{3}{2}] \end{pmatrix}, \qquad X = \begin{pmatrix} [-\frac{1}{3}, 1] \\ [-\frac{8}{3}, -\frac{3}{2}] \end{pmatrix},$$

and thus $X^* \subsetneq X$.

If $\underline{B} \geqq 0$, $\bar{B} \leqq 0$, $0 \in B$ we have by [3, 4] $X = A^G B = A^H B$, and by Theorem 4.1(i), $A^H B \subseteq X^*$ from which $X = X^* = A^H B$ follows. □

Remark. If A is an H-matrix but not an M-matrix, the inclusion $X^* \supsetneq X$ is possible. An example is provided by choosing

$$A := \begin{pmatrix} 1 & -0.5 \\ 0.5 & 1 \end{pmatrix}, \qquad B := \begin{pmatrix} [-1, 1] \\ [-1, 1] \end{pmatrix},$$
$$D^* := A, \quad D := \operatorname{diag}(1, 1).$$

Then

$$X^* = \begin{pmatrix} [-2, 2] \\ [-2, 2] \end{pmatrix} \supsetneq \begin{pmatrix} [-1.6, 1.6] \\ [-1.2, 1.2] \end{pmatrix} = A^G B.$$

This example also shows that in Theorem 3 of [9] the assumption that A is an M-matrix cannot be relaxed to A is an H-matrix. This counterexample is simpler than the one given in [10].

If A is an H-matrix the spectral radii appearing in (4.16) may be estimated by bounding $|D^G|$ by $\langle D \rangle^{-1}$. We now compare the resulting bounds for different partitions.

[1] However, it should be noted that the single-step methods (4.9) and (4.10) are not special cases of the incomplete LU-factorization of [9].

THEOREM 4.5. *Let π and π^* be two partitions* (2.5) *and let π^* be finer than π. Let $A \in \mathbb{R}^{n\times n}$ be an H-matrix. Setting*

$$\hat{R} := \langle D\rangle^{-1}|L|, \qquad \hat{S} := \langle D\rangle^{-1}|U|,$$

the following relations hold:

$$\rho(\hat{R}+\hat{S}) \leqq \rho(\hat{R}^*+\hat{S}^*), \tag{4.29}$$

$$\rho((I-\hat{R})^{-1}\hat{S}) \leqq \rho((I-\hat{R}^*)^{-1}\hat{S}^*). \tag{4.30}$$

If D is an M-matrix, then (4.29), (4.30) *become*

$$\rho(R+S) \leqq \rho(R^*+S^*),$$

$$\rho((I-R)^{-1}S) \leqq \rho((I-R^*)^{-1}S^*).$$

Proof. Since $\langle A\rangle$ has the regular splittings

$$\langle A\rangle = \langle D\rangle - |L| - |U| = \langle D^*\rangle - |L^*| - |U^*|$$

with $|L| + |U| \leqq |L^*| + |U^*|$ relation (4.29) follows by using Corollary (5.7) of [5, p. 183]. Moreover, we have $\rho(\hat{R}+\hat{S}) < 1$ and $\rho(\hat{R}^*+\hat{S}^*) < 1$, and hence $\langle D\rangle^{-1}\langle A\rangle = I - \hat{R} - \hat{S}$ and $I - \hat{R}$ are inverse-nonnegative. Since $\hat{S} \geqq 0$ we may apply Theorem 5.2 of [5, p. 181] to obtain

$$\rho((I-\hat{R})^{-1}\hat{S}) = \frac{\rho(\langle A\rangle^{-1}|U|)}{1+\rho(\langle A\rangle^{-1}|U|)}.$$

A similar formula holds for the splitting $\langle A\rangle = \langle D^*\rangle - |L^*| - |U^*|$. Since $|U| \leqq |U^*|$ and $x/(1+x)$ is a monotone increasing function of x for $x \geqq 0$, (4.30) follows.

The statement for the case that D is an M-matrix follows from the remark to Lemma 2.3. □

Remark. We did not yet succeed in showing that

$$\rho((I-\hat{S})^{-1}\hat{R}(I-\hat{R})^{-1}\hat{S}) \leqq \rho((I-\hat{S}^*)^{-1}\hat{R}^*(I-\hat{R}^*)^{-1}\hat{S}^*) \tag{4.31}$$

holds if A is an H-matrix. When proceeding as in the proof of Theorem 4.5 and using the regular splitting

$$\langle D\rangle^{-1}\langle A\rangle = I - \hat{R} - \hat{S} = (I-\hat{R})(I-\hat{S}) - \hat{R}\hat{S},$$

then we obtain

$$\rho((I-\hat{S})^{-1}\hat{R}(I-\hat{R})^{-1}\hat{S}) = \frac{\rho(\langle A\rangle^{-1}|L|\langle D\rangle^{-1}|U|)}{1+\rho(\langle A\rangle^{-1}|L|\langle D\rangle^{-1}|U|)}.$$

A similar formula holds for the right-hand side of (4.31). However, both formulas do not seem to be comparable because of $\langle D^*\rangle^{-1} \leqq \langle D\rangle^{-1}$.

If π^* is given by

$$A = \begin{pmatrix} A_{11} & A_{12} & 0 & \cdots & 0 \\ 0 & A_{22} & A_{23} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & \ddots & \\ 0 & & & \ddots & \ddots & A_{k-1,k} \\ A_{k1} & 0 & \cdots & & 0 & A_{kk} \end{pmatrix}.$$

Then by the functional equation derived in [27]

$$\rho((I-\hat{S}^*)^{-1}\hat{R}^*(I-\hat{R}^*)^{-1}\hat{S}^*) = \rho^k(\hat{R}^*+\hat{S}^*)$$

and an analogous equation for the spectral radius with respect to the partition π hold. Then (4.31) follows by (4.29).

If we have an enclosure W for $A^H B$ it is advantageous to form the intersection $V_i^{(l)} \cap W_i^{(l)}$ after having calculated $V^{(l)}$ (taking intersection after each iteration step) or $V_i^{(l)}$ (taking intersection after having calculated each component). For a discussion of different methods see [2, Chap. 14] and [12, § 8].

REFERENCES

[1] G. ALEFELD, *Über die Durchführbarkeit des Gausschen Algorithmus bei Gleichungen mit Intervallen als Koeffizienten*, Comput. Suppl., 1 (1977), pp. 15–19.

[2] G. ALEFELD AND J. HERZBERGER, *Introduction to Interval Computations*, Academic Press, New York, 1983.

[3] W. BARTH AND E. NUDING, *Optimale Lösung von Intervallgleichungssystemen*, Computing, 12 (1974), pp. 117–125.

[4] H. BEECK, *Zur scharfen Aussenabschätzung der Lösungsmenge bei linearen Intervallgleichungssystemen*, Z. Angew. Math. Mech., 54 (1974), pp. T208–T209.

[5] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[6] J. GARLOFF, *Solution of linear equations having a Toeplitz interval matrix as coefficient matrix*, Opuscula Math., 2 (1986), pp. 33–45.

[7] A. S. HOUSEHOLDER, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.

[8] E. ISAACSON AND H. B. KELLER, *Analysis of Numerical Methods*, John Wiley, New York, 1966.

[9] G. MAYER, *Enclosing the solution set of linear systems with inaccurate data by iterative methods based on incomplete* LU*-decompositions*, Computing, 35 (1985), pp. 189–206.

[10] ———, *Über Iterationsverfahren zur Lösungseinschliessung linearer Gleichungssysteme mit ungenauen Eingangsdaten*, Z. Angew. Math. Mech., 66 (1986), pp. T417–T419.

[11] O. MAYER, *Algebraische und metrische Strukturen in der Intervallrechnung und einige Anwendungen*, Computing, 5 (1970), pp. 144–162.

[12] A. NEUMAIER, *New techniques for the analysis of linear interval equations*, Linear Algebra Appl., 58 (1984), pp. 273–325.

[13] ———, *Further results on linear interval equations*, Linear Algebra Appl., 87 (1987), pp. 155–179.

[14] ———, *Linear interval equations*, in Interval Mathematics 1985, K. Nickel, ed., Lecture Notes in Computer Science 212, Springer-Verlag, Berlin, 1986, pp. 109–120.

[15] M. NEUMANN, *On the Schur complement and the* LU*-factorization of a matrix*, Linear and Multilinear Algebra, 9 (1981), pp. 241–254.

[16] Y. OHTA AND D. D. ŠILJAK, *Overlapping block diagonal dominance and existence of Liapunov functions*, J. Math. Anal. Appl., 112 (1985), pp. 396–410.

[17] K. REICHMANN, *Abbruch beim Intervall-Gauss-Algorithmus*, Computing, 22 (1979), pp. 355–361.

[18] F. ROBERT, *Blocs-H-matrices et convergence des methodes iteratives classiques par blocs*, Linear Algebra Appl., 2 (1969), pp. 223–265.

[19] J. SCHRÖDER, *Operator Inequalities*, Academic Press, New York, 1980, p. 41.

[20] H. SCHWANDT, *An interval arithmetic approach for the construction of an almost globally convergent method for the solution of the nonlinear Poisson equation on the unit square*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 427–452.

[21] ———, *The solution of nonlinear elliptic Dirichlet problems on rectangles by almost globally convergent interval methods*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 617–638.

[22] R. TOST, *Lösung der* 1. *Randwertaufgabe der Laplace-Gleichung im Rechteck mit intervallanalytischen Methoden*, Ber. Gesellsch. Math. Datenverarb. No. 28, Bonn, 1970.

[23] ———, *Zur numerischen Lösung von Randwertaufgaben mit gesicherter Fehlereinschliessung bei partiellen Differentialgleichungen*, Z. Angew. Math. Mech., 51 (1971), pp. T74–T75.

[24] M. R. VALENCA, *Multiple shooting using interval analysis*, BIT, 25 (1985), pp. 425–427; extended version, Freiburger Intervall-Ber., 85/2 (1985), pp. 35–44.

[25] R. S. VARGA, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1962.

[26] R. S. VARGA AND DO-YONG CAI, *On the* LU*-factorization of M-matrices*, Numer. Math., 38 (1981), pp. 179–192.

[27] R. S. VARGA, W. NIETHAMMER, AND DO-YONG CAI, *p-cyclic matrices and the symmetric successive overrelaxation method*, Linear Algebra Appl., 58 (1984), pp. 425–439.

[28] P. WISSKIRCHEN, *Vergleich intervallarithmetischer Iterationsverfahren*, Computing, 14 (1975), pp. 45–49.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 107–112, January 1990

007

LINEAR MAPS THAT PRESERVE AN INERTIA CLASS*

RAPHAEL LOEWY†

Abstract. The purpose of this paper is to prove the following result: Let $n \geq 3$ and let r, s be given positive integers such that $r \neq s$ and $r + s \leq n$. Let $\mathcal{H}_n$ denote the space of all $n \times n$ hermitian matrices. Suppose that $T: \mathcal{H}_n \to \mathcal{H}_n$ is a linear transformation that maps the set of all matrices with r positive eigenvalues and s negative eigenvalues into itself. Then there exists an $n \times n$ nonsingular matrix S such that either $T(A) = S^*AS$ for all $A \in \mathcal{H}_n$ or $T(A) = S^*A^tS$ for all $A \in \mathcal{H}_n$. This gives an affirmative answer to a problem raised by Johnson and Pierce.

Key words. Hermitian matrix, inertia, rank-k nonincreasing linear map

AMS(MOS) subject classifications. 15A57, 15A04

1. Introduction.

Let $\mathcal{H}_n$ denote the set of all $n \times n$ complex hermitian matrices, and let S_n denote the set of all $n \times n$ real symmetric matrices. If A is in $\mathcal{H}_n$ or S_n, and has r positive eigenvalues, s negative eigenvalues, and t zero eigenvalues, $r + s + t = n$, then the *inertia* of A is defined to be the triple $\text{In}(A) = (r, s, t)$. We let $\pi(A) = r$, $\nu(A) = s$, $\delta(A) = t$. Following [5], let $G(r, s, t)$ denote the set of all $A \in \mathcal{H}_n$ such that $\text{In}(A) = (r, s, t)$. $G_S(r, s, t)$ will denote the corresponding set in S_n. We assume throughout that r, s, t are nonnegative integers such that $r + s + t = n$.

Given a linear transformation T on $\mathcal{H}_n$, we say T preserves $G(r, s, t)$ if and only if $T(G(r, s, t)) \subseteq G(r, s, t)$. Our purpose here is to consider the following conjecture, due to Johnson and Pierce [5].

Conjecture. Suppose that $n \geq 3$ and $T: \mathcal{H}_n \to \mathcal{H}_n$ is a linear map that preserves the (fixed) inertia class $G(r, s, t)$. Suppose that $r > 0$ and $s > 0$. Then there exists a nonsingular $n \times n$ complex matrix S such that

$$(1) \qquad \begin{aligned} &\text{either} \quad T(A) = \varepsilon S^*AS \quad \text{for all } A \in \mathcal{H}_n \\ &\text{or} \quad T(A) = \varepsilon S^*A^tS \quad \text{for all } A \in \mathcal{H}_n. \end{aligned}$$

where $\varepsilon = 1$ if $r \neq s$ and $\varepsilon = \pm 1$ if $r = s$.

We prove this conjecture except for the case $r = s$. Note that T is not assumed to be invertible in the conjecture.

Johnson and Pierce [5] proved the conjecture in the special cases where (r, s, t) takes the form $(n - 1, 1, 0)$ or $(k + 1, k, 0)$ (and, obviously, triples obtained from these by interchanging r and s). They also obtained in [5] the following result.

THEOREM 1. *Suppose that $T: \mathcal{H}_n \to \mathcal{H}_n$ is an invertible linear map. Suppose that (r, s, t) is an inertia triple which is not one of*

$$\{(n, 0, 0), (0, n, 0), (0, 0, n), (n/2, n/2, 0)\}$$

and suppose that T preserves the inertia class $G(r, s, t)$. Then T has the form (1).

There are additional results in the literature that deal with inertia preservers. Helton and Rodman [3] showed that if $n \geq 3$ and k is a fixed positive integer such that $2k \neq n$ and $k < n$, and if $T: \mathcal{H}_n \to \mathcal{H}_n$ is an invertible linear map that preserves $G(k, n - k, 0)$ and satisfies $T(I_n) = I_n$, then T is a unitary similarity or a unitary similarity

* Received by the editors December 1, 1988; accepted for publication (in revised form) April 7, 1989. This research was supported by the Fund for the Promotion of Research at the Technion.

† Department of Mathematics, Technion-Israel Institute of Technology, Haifa 32000, Israel (MAR28AA@TECHNION.BITNET).

composed with transposition. Schneider [7] showed that if a linear map on $\mathcal{H}_n$ maps the set of positive definite matrices onto itself, then it must be a congruence or a congruence composed with transposition. Johnson and Pierce [4] extended Schneider's result and showed that if k is a fixed integer such that $0 < k < n$ and T: $\mathcal{H}_n \to \mathcal{H}_n$ is a linear transformation that maps $G(k, n-k, 0)$ onto itself, then either $T(A) = \varepsilon S^* A S$ for all $A \in \mathcal{H}_n$ or $T(A) = \varepsilon S^* A^t S$ for all $A \in \mathcal{H}_n$, where $\varepsilon = 1$ if $2k \neq n$, and $\varepsilon = \pm 1$ if $2k = n$. *Finally, it should be noted that the obvious analogues of the results quoted above hold if we consider S_n instead of $\mathcal{H}_n$.*

We end the introduction with a few preliminary remarks. The rank of a matrix A is denoted by $\rho(A)$. If α is any subset of $\{1, 2, \cdots, n\}$ and A is an $n \times n$ matrix. $A[\alpha]$ denotes the principal submatrix of A based on the row and column indices of α.

The set of $n \times n$ positive semidefinite matrices is known to be a closed, pointed, convex cone. The space $\mathcal{H}_n$ is a partially ordered vector space over the reals, where we define for $A, B \in \mathcal{H}_n$ that $A \geqq B$ if and only if $A-B$ is positive semidefinite. Given any $n \times n$ positive semidefinite matrix Q, $\phi(Q)$, the *face* generated by Q is defined by

$$\phi(Q) = \{A \in \mathcal{H}_n : A \geqq 0 \quad \text{and } \exists \alpha > 0 \quad \text{such that } Q \geqq \alpha A\}.$$

(For a definition of a face in an arbitrary convex cone and some basic properties of faces, cf. [1].)

Now suppose that $\rho(Q) = r$. Then there exists a nonsingular $n \times n$ complex matrix S such that

$$Q = S^*(I_r \oplus O_{n-r})S = S^* \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} S.$$

It is well known that for $A \in \mathcal{H}_n$, $A \geqq 0$, $A \in \phi(Q)$ if and only if there exists $B \in \mathcal{H}_r$, $B \geqq 0$, such that

$$A = S^* \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} S.$$

If $A \in \mathcal{H}_n$, then A belongs to the subspace spanned by $\phi(Q)$ if and only if there exists $B \in \mathcal{H}_r$ such that

$$A = S^* \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} S.$$

Given a linear map T on $\mathcal{H}_n$ and a positive integer k such that $0 < k < n$, we say T is *rank k nonincreasing*, provided $A \in \mathcal{H}_n$ and $\rho(A) = k$ imply $\rho(T(A)) \leqq k$.

2. Main result. Our purpose is to prove the following result, thus confirming the conjecture of Johnson and Pierce, except for the case $r = s$.

THEOREM 2. *Let $n \geqq 3$ and let T: $\mathcal{H}_n \to \mathcal{H}_n$ be a linear transformation. Let (r, s, t) be an inertia triple such that $r > 0$, $s > 0$ and $r \neq s$. If T preserves $G(r, s, t)$, then T has the form* (1) *with $\varepsilon = 1$.*

Remark 1. The obvious analogue of Theorem 2 holds if one considers a linear map on S_n preserving $G_S(r, s, t)$. The proof follows along the same line given here for the hermitian case.

Remark 2. Let T be a linear transformation on $\mathcal{H}_n$. Let S_1 and S_2 be any $n \times n$ nonsingular complex matrices. Define W: $\mathcal{H}_n \to \mathcal{H}_n$ by $W(A) = S_2 T(S_1 A S_1^*) S_2^*$. Then it is easy to check that T satisfies the assumptions of Theorem 2 if and only if W does, and the same holds true for the conclusion of Theorem 2. Thus, whenever convenient, we may replace T by W.

The proof of Theorem 2 will be based on a sequence of results, some of which might be of independent interest. The next theorem follows from Theorem 3 of [3] and is also stated explicitly in [5].

THEOREM 3. *Let $n \geqq 2$ and let $T: \mathcal{H}_n \to \mathcal{H}_n$ be an invertible linear transformation such that $\rho(T(A)) = 1$ whenever $\rho(A) = 1$. Then there exists a nonsingular $n \times n$ complex matrix S and $\varepsilon = \pm 1$ such that either $T(A) = \varepsilon S^*AS$ for all $A \in \mathcal{H}_n$, or $T(A) = \varepsilon S^*A^tS$ for all $A \in \mathcal{H}_n$.*

THEOREM 4 [6]. *Let $T: \mathcal{H}_n \to \mathcal{H}_n$ be a linear transformation and suppose there exists an integer k, $0 < k < n$, such that $\rho(A) = k$ implies $\rho(T(A)) \leqq k$. Then, for any integer l such that $k \leqq l \leqq n$, $\rho(A) = l$ implies $\rho(T(A)) \leqq l$.*

Thus, Theorem 4 states that if T is rank-k nonincreasing, it must be rank-l nonincreasing for all $k \leqq l \leqq n$.

THEOREM 5 [6]. *Let $T: \mathcal{H}_n \to \mathcal{H}_n$ be a linear transformation and suppose that $\rho(A) < n$ implies $\rho(T(A)) < n$. Then, there exists a real α such that* det $T(A) = \alpha$ det A *for all $A \in \mathcal{H}_n$.*

Analogues on Theorem 5 and the next corollary were obtained by Botta [2] for the space of all $n \times n$ matrices over an algebraically closed field.

COROLLARY 1. *Let $T: \mathcal{H}_n \to \mathcal{H}_n$ be a linear transformation and suppose that $\rho(A) < n$ implies $\rho(T(A)) < n$. Then, either T is invertible or the image of T is contained in the set of singular $n \times n$ hermitian matrices.*

Proof. By Theorem 5, there exists a real α such that det $T(A) = \alpha$ det A for all $A \in \mathcal{H}_n$. If $\alpha = 0$ the result holds, so we may assume $\alpha \neq 0$. Hence, T maps the set of nonsingular hermitian matrices into itself. Suppose T is not invertible. Then there exists $B \in \text{Ker } T$ such that $0 < \rho(B) < n$. Obviously, there exists $C \in \mathcal{H}_n$ such that $\rho(C) = n - \rho(B)$ and $B + C$ is nonsingular. Hence $T(B + C)$ is nonsingular. But $T(B + C) = T(B) + T(C) = T(C)$, and $T(C)$ is singular, a contradiction. □

LEMMA 1. *Let (r, s, t) be an inertia triple such that $r \geqq s > 0$. Suppose $T: \mathcal{H}_n \to \mathcal{H}_n$ is a linear transformation that preserves $G(r, s, t)$. Then, for any $B \in \mathcal{H}_n$ such that* In $(B) = (s, s, n - 2s)$ *we have $\pi(T(B)) \leqq s$, $\nu(T(B)) \leqq s$.*

Proof. We may assume $r > s$, for the case $r = s$ is trivial. Let $B \in \mathcal{H}_n$ be such that In $(B) = (s, s, n - 2s)$. Then, there exists an $n \times n$ nonsingular matrix S such that $B = S^*DS$, where $D = I_s \oplus -I_s \oplus O_{n-2s}$. Let $A = S^*ES$, where

$$E = I_s \oplus -I_s \oplus I_{r-s} \oplus O_{n-r-s}.$$

Then In $(A + \mu B) = (r, s, t)$ for all $\mu > 0$. Therefore, by assumption,

$$\text{In } (T(B + \mu^{-1}A)) = (r, s, t)$$

for all $\mu > 0$. It follows by continuity that $\pi(T(B)) \leqq r$ and $\nu(T(B)) \leqq s$. Next, consider $A - \mu B$. For any $\mu > 1$, In $(A - \mu B) = (r, s, t)$, so In $(T(\mu^{-1}A - B)) = (r, s, t)$. It follows that $\pi(-T(B)) \leqq r$ and $\nu(-T(B)) \leqq s$. Hence $\pi(T(B)) \leqq \min\{r, s\} = s$, and similarly $\nu(T(B)) \leqq s$. □

LEMMA 2. *Let (r, s, t) be an inertia triple such that $r > 0$, $s > 0$, and $r + s < n$. Let $T: \mathcal{H}_n \to \mathcal{H}_n$ be a linear transformation such that $\rho(T(A)) \leqq r + s$ for any $A \in G(r, s, t)$. Then, $\rho(A) \leqq r + s$ implies $\rho(T(A)) \leqq r + s$.*

Proof. Let $m = r + s$, so $t = n - m$. It clearly suffices to show $\rho(T(A)) \leqq m$ whenever $A \in \mathcal{H}_n$ and $\rho(A) = m$. Suppose first that $A \in \mathcal{H}_n$ and

$$\text{In } (A) = (r + 1, s - 1, n - m).$$

Then there exists a nonsingular S such that $A = S^*DS$, where

$$D = \text{diag } (\alpha_1, \alpha_2, \cdots, \alpha_r, \alpha_{r+1}, -\beta_1, \cdots, -\beta_{s-1}, 0, \cdots, 0)$$

and $\alpha_1, \alpha_2, \cdots, \alpha_{r+1}, \beta_1, \cdots, \beta_{s-1}$ are positive. Let

$$D_\varepsilon = \operatorname{diag}(\alpha_1, \alpha_2, \cdots, \alpha_r, \varepsilon, -\beta_1, \cdots, -\beta_{s-1}, 0, \cdots, 0),$$

$B_\varepsilon = S^* D_\varepsilon S$, and $C_\varepsilon = T(B_\varepsilon)$. Then, for any $\varepsilon < 0$, $B_\varepsilon \in G(r, s, t)$ so $\rho(C_\varepsilon) \leqq m$. Hence, every $m+1 \times m+1$ minor of B_ε, which is a polynomial in ε, vanishes for all $\varepsilon < 0$. Therefore it vanishes for all real ε, in particular for $\varepsilon = \alpha_{r+1}$. Hence $\rho(T(A)) \leqq m$.

Repeating the process, one obtains that $\rho(T(A)) \leqq m$ whenever $\rho(A) = m$ and $\pi(A) > r$. Similarly, we may conclude $\rho(T(A)) \leqq m$ whenever $\rho(A) = m$ and $\nu(A) > s$. □

Remark 3. In the proof of Theorem 2 we may clearly assume that $r > s$.

LEMMA 3. *Let $n \geqq 3$ and let $T: \mathscr{H}_n \to \mathscr{H}_n$ be a linear transformation. Let $(r, s, 0)$ be an inertia triple such that $r > 0$, $s > 0$, and $r > s$. If T preserves $G(r, s, 0)$, then T has the form (1) with $\varepsilon = 1$.*

Proof. It follows from Lemma 1 that $\rho(T(A)) \leqq 2s$ for any $A \in G(s, s, n-2s)$. Hence, by Lemma 2, $\rho(T(A)) \leqq 2s$ for any $A \in \mathscr{H}_n$ such that $\rho(A) \leqq 2s$. Since $2s < r + s = n$, it follows now from Theorem 4 that $T(A)$ is singular whenever A is singular. But by the assumptions of the lemma, it is impossible that the image of T consists of singular matrices only, and therefore T must be invertible, by Corollary 1. The result follows now from Theorem 1. □

Proof of Theorem 2. We prove the theorem by induction on n. We may assume $r > s$, by Remark 3. Consider first the case $n = 3$. Here we must have $r = 2$ and $s = 1$, so $r + s = n$. The theorem holds by Lemma 3.

We now describe the general step. Let $m = r + s$, so $t = n - m$. We may assume $m < n$ by Lemma 3. Let Q be any positive semidefinite matrix such that $\rho(Q) = m$. We show that $T(Q)$ is also a positive semidefinite matrix of rank m, and that $\phi(Q)$, the face generated by Q, is mapped by T onto $\phi(T(Q))$.

By Remark 2, we may assume $Q = I_m \oplus O_{n-m}$. Let $D = I_r \oplus -I_s \oplus O_{n-m}$, so $\operatorname{In}(D) = (r, s, n-m)$. By Remark 1, we may also assume $T(D) = D$. Let

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

where $H_{1,1} \in \mathscr{H}_m$ and let

$$T(A) = \begin{bmatrix} F_{1,1} & F_{1,2} \\ F_{1,2}^* & F_{2,2} \end{bmatrix}$$

be partitioned conformably. We claim that $F_{2,2} = 0$. Indeed, for any real ε,

$$T(D + \varepsilon A) = \begin{bmatrix} \varepsilon F_{1,1} + (I_r \oplus -I_s) & \varepsilon F_{1,2} \\ \varepsilon F_{1,2}^* & \varepsilon F_{2,2} \end{bmatrix}.$$

If $\varepsilon > 0$ is small enough, then $D + \varepsilon A \in G(r, s, n-m)$, so $\operatorname{In}(T(D + \varepsilon A)) = (r, s, n-m)$. But if $F_{2,2} \neq 0$ we get immediately that $\rho(T(D + \varepsilon A)) \geqq m + 1$ for $\varepsilon > 0$ sufficiently small, a contradiction. Hence $F_{2,2} = 0$.

Now define a linear transformation $\hat{T}: \mathscr{H}_m \to \mathscr{H}_m$ as follows:

$$\hat{T}(H) = T\left(\begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix}\right)[\{1, 2, \cdots, m\}],$$

for any $H \in \mathscr{H}_m$. By the properties of T and Lemma 1 it is clear that if $H \in \mathscr{H}_m$ and $H \in G(s, s, m-2s)$, then $\rho(\hat{T}(H)) \leqq 2s$. Hence, by Lemma 2, $\rho(\hat{T}(H)) \leqq 2s$ whenever $H \in \mathscr{H}_m$ and $\rho(H) \leqq 2s$. Thus, $\hat{T}$ is rank-$2s$ nonincreasing. Since $2s < r + s = m$,

we may conclude from Theorem 4 that $\hat{T}$ maps any singular matrix in $\mathscr{H}_m$ to a singular matrix. Since $T(D) = D$ we have $\hat{T}(I_r \oplus -I_s) = I_r \oplus -I_s$, so the image of $\hat{T}$ contains an invertible matrix. Hence, by Corollary 1, $\hat{T}$ is invertible. It follows from Theorem 3 of [3] that there exists an invertible $m \times m$ matrix χ such that $\hat{T}(H) = \chi^* H \chi$ for all $H \in \mathscr{H}_m$, or $\hat{T}(H) = \chi^* H^t \chi$ for all $H \in \mathscr{H}_m$ (note that the ε that appears in Theorem 3 of [3] must be 1, as $\hat{T}(I_r \oplus -I_s) = I_r \oplus -I_s$). Thus, $\hat{T}$ preserves inertia of every matrix in $\mathscr{H}_m$. In particular, if $H \in \mathscr{H}_m$ and $\rho(H) = 1$, then $\rho(\hat{T}(H)) = 1$, while if H is positive definite, so is $\hat{T}(H)$.

Now we consider $T(Q)$. By the definition of Q, we have

$$T(Q) = \begin{bmatrix} \hat{T}(I_m) & F \\ F^* & 0 \end{bmatrix},$$

where $\hat{T}(I_m)$ is positive definite, and $F \in \mathbb{C}^{m,n-m}$. By Lemma 2, $\rho(T(Q)) \leqq m$. This implies that $F = 0$. Similarly if $H \in \mathscr{H}_m$ is positive definite, then

$$T\left(\begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} \hat{T}(H) & 0 \\ 0 & 0 \end{bmatrix},$$

where $\hat{T}(H)$ is positive definite. Since every hermitian matrix is the difference of two positive definite matrices, we get that

$$T\left(\begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} \hat{T}(H) & 0 \\ 0 & 0 \end{bmatrix}$$

for any $H \in \mathscr{H}_m$. Since $\hat{T}$ is invertible, we conclude that T maps $\phi(Q)$ onto $\phi(T(Q))$ and the subspace spanned by $\phi(Q)$ onto the subspace spanned by $\phi(T(Q))$. Also, for any matrix B in the subspace spanned by $\phi(Q)$, we have $\text{In}\,(T(B)) = \text{In}\,(B)$.

We claim now that if $H \in \mathscr{H}_n$ is a rank-1 positive semidefinite matrix, so is $T(H)$. Indeed, such a matrix belongs to some face generated by a positive semidefinite matrix of rank m. The preceding discussion shows that $\text{In}\,(T(H)) = \text{In}\,(H)$, so $T(H)$ is a rank-1 positive semidefinite matrix. Hence T is order preserving, that is, if $A \geqq B$ then $T(A) \geqq T(B)$. Using a similar reasoning, since $m = r + s \geqq 3$, if $H \in \mathscr{H}_n$ and $\text{In}\,(H) = (1, 1, n-2)$, then $\text{In}\,(T(H)) = (1, 1, n-2)$.

Our next goal is to show that T is invertible. We first show that if H is any positive semidefinite matrix of rank $n-1$, then so is $T(H)$. We know that for such a matrix H, $T(H)$ is positive semidefinite. Therefore, by Remark 1, we may assume $H = I_{n-1} \oplus O_1$, $T(H) = I_l \oplus O_{n-l}$, for some $l \leqq n-1$. Since T is order preserving, it is clear that given any matrix $H_1 \in \mathscr{H}_{n-1}$, there exists $K_1 \in \mathscr{H}_l$ such that

$$T\left(\begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} K_1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Define now a linear transformation $T_1 \colon \mathscr{H}_{n-1} \to \mathscr{H}_{n-1}$ by

$$T_1(H_1) = T\left(\begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix}\right)[\{1, 2, \cdots, n-1\}].$$

Since T preserves the inertia class $G(r, s, n-m)$ in $\mathscr{H}_n$, it is clear that T_1 preserves the inertia class $G(r, s, n-m-1)$ in $\mathscr{H}_{n-1}$. By the induction hypothesis, T_1 preserves inertia. Hence $l = n-1$, and T maps $\phi(H)$ onto $\phi(T(H))$, and the subspace spanned by $\phi(H)$ onto the subspace spanned by $\phi(T(H))$.

We now show that the image of T must contain a nonsingular matrix. In fact, we claim $T(D_1 + D_2)$ is positive definite, where $D_1 = I_{n-1} \oplus O_1$ and $D_2 = O_1 \oplus I_{n-1}$.

Suppose this is false. Then, as any two positive semidefinite matrices may be simultaneously diagonalized by congruence, we may assume $T(D_1) = I_{n-1} \oplus O_1$. $T(D_2) = \text{diag}(\alpha_1, \alpha_2, \cdots, \alpha_{n-1}, 0)$, where $\alpha_i > 0$, $i = 1, 2, \cdots, n-1$. By our earlier conclusions, there exists a nonzero vector $x = (x_i) \in \mathbb{C}^n$ such that $x_n = 0$ and $T(e_1 e_1^*) = xx^*$, where e_1 is the first standard unit vector. But xx^* belongs also to the face $\phi(T(D_2))$. Hence, there exists a nonzero vector $y = (y_i) \in \mathbb{C}^n$ such that $y_1 = 0$ and $T(yy^*) = xx^*$. We conclude that $0 = xx^* - xx^* = T(e_1 e_1^* - yy^*)$. But

$$\text{In}(e_1 e_1^* - yy^*) = (1, 1, n-2),$$

a contradiction.

Hence, the image of T contains a nonsingular matrix. We also know from Lemmas 1 and 2 that T is rank-$2s$ nonincreasing, that is, $\rho(A) = 2s$ implies $\rho(T(A)) \leqq 2s$. We now conclude from Theorem 4, Theorem 5, and Corollary 1 that T must be invertible. The proof is complete by Theorem 1. □

REFERENCES

[1] G. P. BARKER, *The lattice of faces of a finite dimensional cone*, Linear Algebra Appl., 7 (1973), pp. 71–82.

[2] P. BOTTA, *Linear maps that preserve singular and nonsingular matrices*, Linear Algebra Appl., 20 (1978), pp. 45–49.

[3] J. W. HELTON AND L. RODMAN, *Signature preserving linear maps of hermitian matrices*, Linear and Multilinear Algebra, 17 (1985), pp. 29–37.

[4] C. R. JOHNSON AND S. PIERCE, *Linear maps on hermitian matrices: The stabilizer of an inertia class*, Canad. Math. Bull., 28 (1985), pp. 401–404.

[5] ———, *Linear maps on hermitian matrices: The stabilizer of an inertia class*, II, Linear and Multilinear Algebra, 19 (1986), pp. 21–31.

[6] T. J. LAFFEY AND R. LOEWY, *Linear transformations which do not increase rank*, Linear and Multilinear Algebra, to appear.

[7] H. SCHNEIDER, *Positive operators and an inertia theorem*, Numer. Math., 7 (1965), pp. 11–17.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 113–127, January 1990

008

THE REALIZATION PROBLEM FOR A CLASS OF NONLINEAR SYSTEMS*

LI TIEJUN† AND S. F. McCORMICK†

Abstract. This paper studies the realization problem for a class of nonlinear systems. Necessary and sufficient conditions are contained for realizability of a system and an algorithm is presented for the construction of a realization.

Key words. realization, control systems, commutative diagrams

AMS(MOS) subject classifications. 93B15, 93C10

1. Introduction. Linear theory for control systems is fairly well developed. However, most practical dynamic processes can be described precisely only by nonlinear models. In recent years, the theory of general nonlinear systems of the form $\dot{x}(t) = f(x(t), u(t))$, $y = g(x(t))$, has been developed by using abstract theoretical methods [1]–[3]. In this paper, we study the realization problem for a class of nonlinear systems arising from practical dynamic processes, for example, the model of the fractionating towers in chemical engineering [4]. The general form of this class of nonlinear systems is

$$x(t+1) = Ax(t) + \sum_{i=1}^{m}\sum_{k=1}^{r} u_i^k(t) D_{ik} x(t) + \sum_{i=1}^{m}\sum_{k=1}^{r} u_i^k(t) B_{ik},$$

$$y(t) = Cx(t),$$

where the state $x(t) \in \Re^n$, $u_i(t) \in \Re^1$, $i = 1, 2, \cdots, m$, are inputs of the system, A and D_{ik} are $n \times n$ matrices, B_{ik} and C are $n \times 1$ and $p \times n$ matrices, respectively, and time $t = 0, 1, 2, \cdots$. A necessary and sufficient condition of realizability of this system and an algorithm to construct the realization are given in § 3. We first develop some preliminaries in this and the next section.

For convenience of discussion but without loss of generality, we study only the single-input system case ($m = 1$):

$$\begin{aligned} x(t+1) &= Ax(t) + \sum_{k=1}^{r} u^k(t) D_k x(t) + \sum_{k=1}^{r} u^k(t) B_k, \\ y(t) &= Cx(t). \end{aligned} \tag{1.1}$$

We will use the shortened notation $\Sigma = (A, D_1, \cdots, D_r, B_1, \cdots, B_r, C)$ to refer to system (1.1).

We first introduce additional notation for system (1.1). Let

$$u^{(1)}(t) = \begin{bmatrix} u(t) \\ u^2(t) \\ \vdots \\ u^r(t) \end{bmatrix}$$

* Received by the editors May 23, 1988; accepted for publication (in revised form) March 15, 1989. This work was supported by Air Force Office of Scientific Research grant AFOSR-86-0126 and National Science Foundation grant NSF DMS-8704169.

† Computational Mathematics Group, Department of Mathematics, University of Colorado, Denver, Colorado 80204.

and

$$u^{(j)}(t)=\begin{bmatrix} u^{(j-1)}(t) \\ u^{(j-1)}(t)u(t+j-1) \\ u^{(j-1)}(t)u^2(t+j-1) \\ \vdots \\ u^{(j-1)}(t)u^r(t+j-1) \end{bmatrix},$$

for all $j = 2, 3, \cdots$ and $t = 0, 1, 2, \cdots$. With $u^{(j)}(t)$, we can express the state and output solutions of (1.1) with zero-initial-state (i.e., $x(0) = 0$) as

$$x(t)=\sum_{j=1}^{t} P_j u^{(j)}(t-j), \qquad t\geqq 1$$

and

$$y(t)=\sum_{j=1}^{t} CP_j u^{(j)}(t-j), \qquad t\geqq 1,$$

where P_j is defined by

$$P_1=[B_1, B_2, \cdots, B_r]$$

and

$$P_j=[AP_{j-1}, D_1P_{j-1}, \cdots, D_rP_{j-1}], \qquad j\geqq 2. \tag{1.2}$$

Letting $W_j = CP_j, j \geqq 1$, we then have

$$y(t)=\sum_{j=1}^{t} W_j u^{(j)}(t-j), \qquad t\geqq 1.$$

This expression indicates that the relation between the inputs and outputs of (1.1) is completely determined by the infinite matrix sequence $\{W_j\}$. We call $\{W_j\}$ the input-output (I/O) matrix sequence of system (1.1). Now here W_j is a $p \times r(r+1)^{j-1}$ matrix, where $j \geqq 1$. In this paper, the notation W_j will always signify a matrix with such a size.

The purpose of this paper is to consider the realization problem for (1.1), which consists of the following two subproblems:

(1) For a given infinite sequence of matrices $\{W_j\}$, is there a system of the form (1.1) whose I/O matrix sequence is just $\{W_j\}$? This is the so-called realizability problem. If such a system exists, the sequence $\{W_j\}$ is called realizable and (1.1) is called a realization.

(2) If $\{W_j\}$ is realizable, how can a realization be constructed; that is, how can we construct a system of the form (1.1) whose I/O matrix sequence is just $\{W_j\}$?

If $\{W_j\}$ is realizable, then, in general, it may have a lot of realizations. A realization of $\{W_j\}$ is called minimal if its dimension is the smallest among all its realizations.

The results in § 3 of this paper will provide a complete answer to these two problems.

For system (1.1), we introduce the following matrices:

$$R_1 = C$$

and

$$R_j=\begin{bmatrix} R_{j-1}A \\ R_{j-1}D_1 \\ \vdots \\ R_{j-1}D_r \end{bmatrix}, \qquad j\geqq 2. \tag{1.3}$$

We define matrices Q_q and T_k from P_j and R_j as follows:

$$Q_q = [P_1, P_2, \cdots, P_q], \qquad q \geqq 1 \tag{1.4}$$

and

$$T_k = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_k \end{bmatrix}, \qquad k \geqq 1. \tag{1.5}$$

2. Generalized Hankel matrix. It was well known that Hankel matrices play an important role in the realization problem for linear systems. Here we extend this concept to nonlinear systems by introducing what we call the generalized Hankel matrix of $\{W_j\}$.

Let $S_{1j} = W_j$, $j = 1, 2, \cdots$. Since W_j is a $p \times r(r+1)^{j-1}$ matrix, the number of columns of each W_j, for $j \geqq 2$, is a multiple of $(r+1)$. So we can divide S_{1j} into equal $(r+1)$ blocks:

$$S_{1j} = [S_{1j}^0, S_{1j}^1, \cdots, S_{1j}^r], \qquad j \geqq 2 \tag{2.1}$$

where each S_{1j}^k is a $p \times r(r+1)^{j-2}$ matrix, $k = 0, 1, \cdots, r$. We use these blocks to define

$$S_{2j} = \begin{bmatrix} S_{1j+1}^0 \\ S_{1j+1}^1 \\ \vdots \\ S_{1j+1}^r \end{bmatrix}, \qquad j \geqq 1. \tag{2.2}$$

Note that S_{2j} is defined in terms of S_{1j+1}^k, not S_{1j}^k. Note also that S_{2j} is a $p(r+1) \times r(r+1)^{j-1}$ matrix; we can thus divide it into equal $(r+1)$ blocks:

$$S_{2j} = [S_{2j}^0, S_{2j}^1, \cdots, S_{2j}^r], \qquad j \geqq 2. \tag{2.3}$$

Define

$$S_{3j} = \begin{bmatrix} S_{2j+1}^0 \\ S_{2j+1}^1 \\ \vdots \\ S_{2j+1}^r \end{bmatrix}, \qquad j \geqq 1. \tag{2.4}$$

We can continue in this way and, in general, define

$$S_{i-1j} = [S_{i-1j}^0, S_{i-1j}^1, \cdots, S_{i-1j}^r], \qquad i \geqq 2, \quad j \geqq 2 \tag{2.5}$$

and

$$S_{ij} = \begin{bmatrix} S_{i-1j+1}^0 \\ S_{i-1j+1}^1 \\ \vdots \\ S_{i-1j+1}^r \end{bmatrix}, \qquad i \geqq 2, \quad j \geqq 1 \tag{2.6}$$

where S_{ij} is a $p(r+1)^{i-1} \times r(r+1)^{j-1}$ matrix.

We first give the following lemma, which establishes a relationship among S_{ij}, R_i, and P_j. It will be used in the proof of Theorem 1 in the next section.

LEMMA 1. *Assuming that* $\{W_j\}$ *is the* I/O *matrix sequence of the system* (1.1), *then*

$$S_{ij} = R_i P_j \quad \textit{for all } i, j \geqq 0. \tag{2.7}$$

Proof. Since $\{W_j\}$ is the I/O matrix sequence, we have $W_j = CP_j$ for all $j \geqq 1$. This means that (2.7) is true for $i = 1$ and all $j \geqq 1$. Assume that (2.7) is true for $i - 1$ and all $j \geqq 1$. Then we can write

$$S_{i-1\,j+1} = R_{i-1}P_{j+1} = R_{i-1}[AP_j, D_1P_j, \cdots, D_rP_j], \qquad j \geqq 1.$$

From the definition of S_{ij}, we then have

$$S_{ij} = \begin{bmatrix} R_{i-1}AP_j \\ R_{i-1}D_1P_j \\ \vdots \\ R_{i-1}D_rP_j \end{bmatrix} = \begin{bmatrix} R_{i-1}A \\ R_{i-1}D_1 \\ \vdots \\ R_{i-1}D_r \end{bmatrix} P_j = R_iP_j. \qquad \square$$

We now define the infinite matrix

$$H = \begin{bmatrix} S_{11} & S_{12} & S_{13} & \cdots \\ S_{21} & S_{22} & S_{23} & \cdots \\ S_{31} & S_{32} & S_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{2.8}$$

which we call the generalized Hankel matrix of $\{W_j\}$.

The finite part H_{kq} of H given by

$$H_{kq} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1q} \\ S_{21} & S_{22} & \cdots & S_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ S_{k1} & S_{k2} & \cdots & S_{kq} \end{bmatrix} \tag{2.9}$$

is called the (k, q)-finite Hankel matrix of $\{W_j\}$. H_{kq} is an $l_k \times m_q$ matrix, where $l_k = p(((1 + r)^k - 1)/r)$ and $m_q = (1 + r)^q - 1$. Because of the definition in (2.5) and (2.6) of S_{ij}, H_{kq} is determined by the first $k + q - 1$ terms of the sequence $\{W_j\}$. Evidently, we may write H_{kq} in either of the following two ways:

$$H_{kq} = \begin{bmatrix} S_{11} & S_{12}^0 & S_{12}^1 & \cdots & S_{12}^r & \cdots & S_{1q}^0 & S_{1q}^1 & \cdots & S_{1q}^r \\ S_{21} & S_{22}^0 & S_{22}^1 & \cdots & S_{22}^r & \cdots & S_{2q}^0 & S_{2q}^1 & \cdots & S_{2q}^r \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{k1} & S_{k2}^0 & S_{k2}^1 & \cdots & S_{k2}^r & \cdots & S_{kq}^0 & S_{kq}^1 & \cdots & S_{kq}^r \end{bmatrix}$$

or

$$H_{kq} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1q} \\ S_{12}^0 & S_{13}^0 & \cdots & S_{1q+1}^0 \\ \vdots & \vdots & \ddots & \vdots \\ S_{12}^r & S_{13}^r & \cdots & S_{1q+1}^r \\ \vdots & \vdots & \ddots & \vdots \\ S_{k-12}^0 & S_{k-13}^0 & \cdots & S_{k-1q+1}^0 \\ \vdots & \vdots & \ddots & \vdots \\ S_{k-12}^r & S_{k-13}^r & \cdots & S_{k-1q+1}^r \end{bmatrix}.$$

From these two forms, we see that H_{kq+1} and H_{k+1q} contain the same $(r + 1)$ submatrices H_{kq}^i:

$$H_{kq}^i = \begin{bmatrix} S_{12}^i & S_{13}^i & \cdots & S_{1q+1}^i \\ S_{22}^i & S_{23}^i & \cdots & S_{2q+1}^i \\ \vdots & \vdots & \ddots & \vdots \\ S_{k2}^i & S_{k3}^i & \cdots & S_{kq+1}^i \end{bmatrix}, \qquad i = 0, 1, \cdots, r. \tag{2.10}$$

To be precise, define

$$U_i = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ E_1^i & & & \\ & E_2^i & & \\ & & \ddots & \\ & & & E_q^i \end{bmatrix}$$

and

$$V_i = \begin{bmatrix} 0 & \tilde{E}_1^i & & \\ 0 & & \tilde{E}_2^i & \\ \vdots & & & \ddots \\ 0 & & & \tilde{E}_k^i \end{bmatrix}$$

where

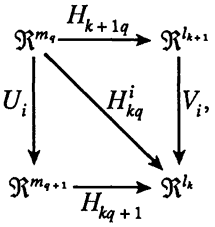
$$E_j^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_j^i \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{and} \quad \tilde{E}_j^i = [0 \cdots 0\, \tilde{I}_j^i\, 0 \cdots 0],$$

with I_j^i the $r(r+1)^{j-1}$ and $\tilde{I}_j^i$ the $p(r+1)^{j-1}$-dimensional identity matrices in the ith positions of E_j^i and $\tilde{E}_j^i$, respectively. Then we have

$$H_{kq}^i = H_{kq+1} U_i = V_i H_{k+1q}, \qquad i = 0, 1, \cdots, r. \tag{2.11}$$

From (2.11) we obtain the following $(r+1)$ commutative diagrams:

(2.12)



3. The realization problem. To study the realization problem for (1.1), we first consider the finite subsequence $\{W_1, W_2, \cdots, W_N\}$ formed from the first N terms of $\{W_j\}$. This is called realizable if there exists a system of the form (1.1) so that the first N terms of the system's I/O matrix sequence are just $\{W_1, W_2, \cdots, W_N\}$. Such a system is called a realization of $\{W_1, W_2, \cdots, W_N\}$. Now the finite Hankel matrices determined by $\{W_1, W_2, \cdots, W_n\}$ are H_{k+1q} and H_{kq+1} given by definition (2.8), where $k, q \geqq 0$ satisfy $N = k + q$.

We first have Theorem 1.

THEOREM 1. *System* (1.1) *is a realization of a finite matrix sequence*

$$\{W_1, W_2, \cdots, W_N\}$$

if and only if

$$H_{kq+1} = T_k Q_{q+1} \tag{3.1}$$

for all $k, q \geqq 0$ *such that* $k + q = N$. *The same is true if* (3.1) *is replaced by*

$$H_{k+1q} = T_{k+1} Q_q. \tag{3.2}$$

Proof. Necessity. If $\Sigma = (A, D_1, \cdots, D_r, B_1, \cdots, B_r, C)$ is a realization of $\{W_1, W_2, \cdots, W_N\}$, then

$$W_j = CP_j, \qquad j = 1, 2, \cdots, N.$$

We therefore can conclude that $S_{ij} = R_i P_j$, $i = 1, 2, \cdots, k, j = 1, 2, \cdots, q+1$ from Lemma 1. This is just (3.1).

Sufficiency. Assume that (3.1) holds for all $k, q \geqq 0$ such that $k + q = N$. Specifically, taking $k = 1$ and $q = N - 1$, we get

$$H_{1N} = T_1 Q_N,$$

that is,

$$[S_{11}, S_{12}, \cdots, S_{1N}] = R_1 [P_1, P_2, \cdots, P_N].$$

From this equality we get $W_j = S_{1j} = CP_j, j = 1, 2, \cdots, N$, which means that the system (1.1) is just a realization of $\{W_1, W_2, \cdots, W_N\}$. □

From the definition of H^i_{kq}, it is straightforward to see that conditions (3.1) and (3.2) are equivalent to the following conditions:

$$H_{kq} = T_k Q_q, \tag{3.3}$$

$$H^0_{kq} = T_k A Q_q, \tag{3.4}$$

$$H^i_{kq} = T_k D_i Q_q, \qquad i = 1, 2, \cdots, r \tag{3.5}$$

for all $k, q \geqq 1$ such that $k + q = N$. Therefore, we also have Theorem 1′.

THEOREM 1′. *The system* (1.1) *is a realization of* $\{W_1, W_2, \cdots, W_N\}$ *if and only if* (3.3)–(3.5) *hold.*

Now we can prove the following result from Theorem 1 (or Theorem 1′), which gives a sufficient condition on the realizability of $\{W_1, W_2, \cdots, W_N\}$.

THEOREM 2. *Suppose that the finite matrix sequence* $\{W_1, W_2, \cdots, W_N\}$ *satisfies*

$$\text{rank}\,(H_{kq}) = \text{rank}\,(H_{kq+1}) = \text{rank}\,(H_{k+1q}), \tag{3.6}$$

for some $k, q \geqq 1$ *such that* $k + q = N$. *Then* $\{W_1, W_2, \cdots, W_N\}$ *is realizable.*

Proof. From the definition of H_{kq}, we have

$$H_{kq+1} = \left[\begin{array}{c|c} H_{kq} & \begin{matrix} S_{1q+1} \\ \vdots \\ S_{kq+1} \end{matrix} \end{array}\right]$$

and

$$H_{k+1q} = \left[\begin{array}{c} H_{kq} \\ \hline S_{k+11} \cdots S_{k+1q} \end{array}\right].$$

Therefore, from condition (3.6), we can conclude that there are two matrices K and L such that

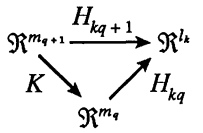
$$H_{kq+1} = H_{kq} K \quad \text{and} \quad H_{k+1q} = L H_{kq}$$

where

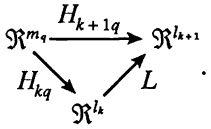
$$K = [I_{m_q} \,|\, *] \quad \text{and} \quad L = \left[\begin{array}{c} I_{l_k} \\ \hline * \end{array}\right],$$

and I_{m_q} and I_{l_k} are identity matrices of dimensions m_q and l_k, respectively.

These two equalities give two commutative diagrams:

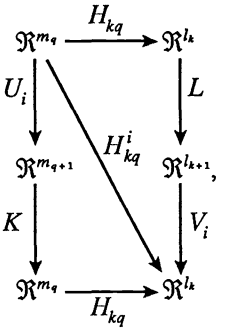


and



We can now add these to diagram (2.12) to obtain:

(3.7)

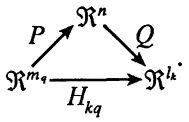


$i = 0, 1, \cdots, r.$

Denoting n = rank (H_{kq}), then H_{kq} can be decomposed into a product of two rank n matrices Q and P:

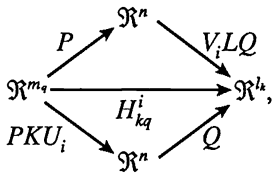
$$H_{kq} = QP,$$

where Q and P are $l_k \times n$ and $n \times m_q$ matrices, respectively. So we obtain a commutative diagram:



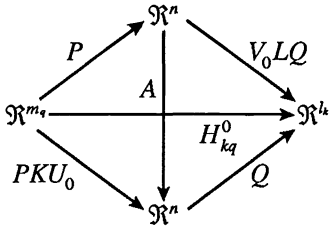
Therefore, (3.7) can be rewritten as follows:

(3.8)

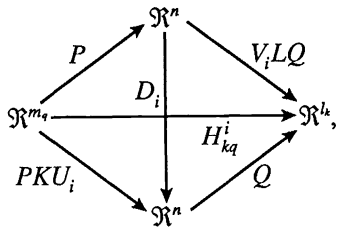


$i = 0, 1, \cdots, r.$

P and Q are rank n, P is an onto mapping and Q is an injection. So, by the Zeiger fill-in Lemma (see [5, Chap. 10]), there are $n \times n$ matrices A, $D_1, \cdots, D_r$ so that the following diagrams are still commutative:



and



(3.9) $$i = 1, 2, \cdots, r;$$

that is,

$$H^0_{kq} = QAP \quad \text{and} \quad H^i_{kq} = QD_iP, \qquad i = 1, 2, \cdots, r.$$

Now let B_i, $i = 1, 2, \cdots, r$, be the ith column of P (P is an $n \times m_q$ matrix), and let C be the first p rows of Q. Then we obtain a system

$$\Sigma = (A, D_1, \cdots, D_r, B_1, \cdots, B_r, C).$$

We want to prove that Σ is just a realization of $\{W_1, W_2, \cdots, W_N\}$. To do this, we need only to prove that $Q = T_k$ and $P = Q_q$. To this end, note from (3.9) that

(3.10) $$AP = PKU_0 \quad \text{and} \quad D_iP = PKU_i, \quad i = 1, 2, \cdots, r.$$

From the characters of K and U_i, we have

(3.11) $$KU_i = \begin{bmatrix} 0 & 0 & \cdots & 0 & \\ E^i_1 & & & & \\ & E^i_2 & & & * \\ & & \ddots & & \\ & & & E^i_{q-1} & \end{bmatrix}, \qquad i = 0, 1, 2, \cdots, r.$$

Let $P_1 = [B_1, B_2, \cdots, B_r]$ and P_i be the matrix formed from the $[(r+1)^{i-1}]$th column to the $[(r+1)^i + 1]$th column of P, $i = 1, 2, \cdots, q$. Then $P = [P_1, P_2, \cdots, P_q]$. From (3.10) and (3.11) we obtain

$$AP_1 = P \begin{bmatrix} 0 \\ E^0_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = (r+1)\text{th column to the } (2r)\text{th column of } P,$$

$$D_1P_1 = P \begin{bmatrix} 0 \\ E^1_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = (2r+1)\text{th column to the } (3r)\text{th column of } P,$$

$$\vdots$$

$$D_rP_1 = P \begin{bmatrix} 0 \\ E^r_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = [(r+1)r+1]\text{th column to the } [(r+2)r]\text{th column of } P.$$

Therefore, $P_2 = [AP_1, D_1P_1, \cdots, D_rP_1]$. In a similar way, we obtain

$$P_i = [AP_{i-1}, D_1P_{i-1}, \cdots, D_rP_{i-1}], \qquad i = 1, 2, \cdots, q.$$

These indicate that Q_q of Σ is just P, and T_k of Σ is just Q, so the proof is complete. □

We now prove the main result in this paper.

THEOREM 3. *An infinite matrix sequence* $\{W_j\}$ *is realizable if and only if there are two positive integers* k *and* q *so that*

$$\text{rank}\,(H_{k+iq+j}) = \text{rank}\,(H_{kq}) \tag{3.12}$$

for all $i, j = 0, 1, \cdots$.

Proof. Necessity. Assume that $\{W_j\}$ is realizable and let

$$\Sigma = (A, D_1, \cdots, D_r, B_1, \cdots, B_r, C)$$

be its realization with minimal dimension n (that is, the matrices A and D_i are of minimal dimension $n \times n$). Then it is not difficult to see that rank (T_n) = rank $(Q_n) = n$. Thus, there exist two positive integers k and q such that

$$\text{rank}\,(T_k) = \text{rank}\,(Q_q) = n. \tag{3.13}$$

Now letting $N = k + q$, then it is clear that Σ is also a realization of the finite matrix sequence $\{W_1, W_2, \cdots, W_N\}$. Therefore, from Theorem 1 we have

$$H_{kq} = T_kQ_q.$$

Thus, from (3.13) we have rank $(H_{kq}) = n$. Furthermore, for any positive integers i and j, Σ is also a realization of $\{W_1, W_2, \cdots, W_{N+i+j}\}$. Thus, we also have rank $(H_{k+iq+j}) = n$. Therefore, rank (H_{k+iq+j}) = rank (H_{kq}).

Sufficiency. Assume that (3.12) is true for all positive integers i and j. Let $N = k + q$. Then from Theorem 2, we conclude that $\{W_1, W_2, \cdots, W_N\}$ is realizable. Letting $\Sigma = (A, D_1, \cdots, D_r, B_1, \cdots, B_r, C)$ be the minimal realization of $\{W_1, W_2, \cdots, W_N\}$, we then have

$$W_j = CP_j, \qquad j = 1, 2, \cdots, N.$$

Now letting $CP_j = \tilde{W}_j$ for all $j \geqq N + 1$, we obtain an infinite matrix sequence

$$\{W_1, \cdots, W_N, \tilde{W}_{N+1}, \tilde{W}_{N+2}, \cdots\} \tag{3.14}$$

from the system Σ. Clearly, the proof of the sufficiency will be completed if we can show that $\tilde{W}_j = W_j$ for all $j \geqq N + 1$.

To this end, denote the Hankel matrix of (3.14) by $\tilde{H}$. Then

$$\tilde{H}_{k+1q+1} = \begin{bmatrix} H_{kq} & H_1 \\ H_2 & \tilde{S}_{k+1q+1} \end{bmatrix}, \tag{3.15}$$

where

$$H_1 = \begin{bmatrix} S_{1q+1} \\ S_{2q+1} \\ \vdots \\ S_{kq+1} \end{bmatrix}$$

and

$$H_2 = [S_{k+11}, S_{k+12}, \cdots, S_{k+1q}].$$

On the other hand, we have

$$H_{k+1q+1} = \begin{bmatrix} H_{kq} & H_1 \\ H_2 & S_{k+1q+1} \end{bmatrix}. \tag{3.16}$$

Note that

$$[H_{kq} \quad H_1] = H_{kq+1} \quad \text{and} \quad \begin{bmatrix} H_{kq} \\ H_2 \end{bmatrix} = H_{k+1q}.$$

Therefore, from (3.12), we have

$$\operatorname{rank}[H_{kq} \quad H_1] = \operatorname{rank}\begin{bmatrix} H_{kq} \\ H_2 \end{bmatrix} = \operatorname{rank}(H_{kq}) = n$$

and

$$\operatorname{rank}(H_{k+1q+1}) = n,$$

where n is the dimension of Σ. Moreover, we can also assert that rank $(\tilde{H}_{k+1q+1}) = n$. In fact, since $H_{kq} = T_k Q_q$ and rank $(H_{kq}) = n$, we have rank (T_k) = rank $(Q_q) = n$. Note that for all positive integers k and q, Q_q is an $n \times [(r+1)^q - 1]$ matrix and T_k is a $p[((r+1)^k - 1)/r] \times n$ matrix. We therefore conclude that rank (T_{k+1}) = rank $(Q_{q+1}) = n$. It is clear that

$$\tilde{H}_{k+1q+1} = T_{k+1} \cdot Q_{q+1},$$

so rank $(\tilde{H}_{k+1q+1}) = n$.

Summarizing these facts about $\tilde{H}_{k+1q+1}$ and H_{k+1q+1} (see (3.15) and (3.16)), we have

$$\begin{aligned} \operatorname{rank}(\tilde{H}_{k+1q+1}) &= \operatorname{rank}(H_{k+1q+1}) \\ &= \operatorname{rank}(H_{kq}) \\ &= \operatorname{rank}[H_{kq} \quad H_1] \\ &= \operatorname{rank}\begin{bmatrix} H_{kq} \\ H_2 \end{bmatrix}. \end{aligned} \tag{3.17}$$

To complete this proof, we first need to develop a simple result in linear algebra.

LEMMA 2. *Let*

$$H = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad \textit{and} \quad \tilde{H} = \begin{bmatrix} A & B \\ C & \tilde{D} \end{bmatrix}.$$

If

$$\operatorname{rank}(H) = \operatorname{rank}(\tilde{H}) = \operatorname{rank}(A) = \operatorname{rank}[A \quad B] = \operatorname{rank}\begin{bmatrix} A \\ C \end{bmatrix},$$

then we must have $D = \tilde{D}$.

Proof. Assume that rank $(H) = n$. Because rank $[A \quad B]$ = rank $(H) = n$, $[A\ B]$ can be transformed by elementary operations into the form

$$\begin{bmatrix} A_1 & B_1 \\ 0 & 0 \end{bmatrix},$$

where $[A_1\ B_1]$ is a matrix with n rows. This implies that there is a matrix P such that

$$P[A\quad B]=\begin{bmatrix}A_1 & B_1\\ 0 & 0\end{bmatrix}.$$

Since elementary transformations do not change the rank of a matrix, rank $(A_1) = n$. Now we can write

$$\begin{bmatrix}P & 0\\ 0 & I\end{bmatrix}H=\begin{bmatrix}A_1 & B_1\\ 0 & 0\\ C & D\end{bmatrix}$$

and

$$\begin{bmatrix}P & 0\\ 0 & I\end{bmatrix}\tilde{H}=\begin{bmatrix}A_1 & B_1\\ 0 & 0\\ C & \tilde{D}\end{bmatrix}.$$

Because

$$\operatorname{rank}\begin{bmatrix}A\\ C\end{bmatrix}=n,$$

there is a unique matrix L such that

$$C=LA_1.$$

On the other hand, because rank (H) = rank $(\tilde{H})$ = n, we also have

$$\operatorname{rank}\left(\begin{bmatrix}P & 0\\ 0 & I\end{bmatrix}H\right)=\operatorname{rank}\left(\begin{bmatrix}P & 0\\ 0 & I\end{bmatrix}\tilde{H}\right)=n.$$

Therefore, there are two matrices M and $\tilde{M}$ such that

$$[C\quad D]=M[A_1\quad B_1]\quad\text{and}\quad[C\quad\tilde{D}]=\tilde{M}[A_1\quad B_1].$$

So

$$C=MA_1\quad\text{and}\quad C=\tilde{M}A_1$$

and

$$D=MB_1\quad\text{and}\quad\tilde{D}=\tilde{M}B_1.$$

Note that $C = LA_1$ and the matrix L is unique. Therefore, we must have $M = \tilde{M} = L$, and hence $D = \tilde{D}$. This completes the proof of the lemma.

Returning now to (3.17), from this lemma we can conclude that $\tilde{S}_{k+1q+1} = S_{k+1q+1}$ in $\tilde{H}_{k+1q+1}$ and H_{k+1q+1}. We therefore have $\tilde{W}_{N+1} = W_{N+1}$. Induction now establishes $\tilde{W}_j = W_j$ for all $j \geqq N + 1$, so the proof of Theorem 3 is complete. □

Finally, from the discussions above, we can develop the following algorithm to construct a realization of $\{W_j\}$ whenever it is realizable.

Let $\{W_j\}$ be an infinite matrix sequence, where W_j is a $p \times r(r+1)^{j-1}$ matrix, $j = 1, 2, \cdots$.

(1) Calculate S_{ij} as in (2.1)–(2.6).

(2) Find two positive integers k and q that satisfy condition (3.12).

(3) Let n be the rank of H_{kq} and decompose H_{kq} as

$$H_{kq}=QP,$$

where Q and P are rank n.

(4) Calculate A, D_i, B_i, and C as follows:

$$A = (Q^TQ)^{-1}Q^TH_{kq}^0P^T(PP^T)^{-1},$$

$$D_i = (Q^TQ)^{-1}Q^TH_{kq}^iP^T(PP^T)^{-1}, \qquad i = 1, 2, \cdots, r,$$

$$B_i = \text{the } i\text{th column of } P, \qquad i = 1, 2, \cdots, r,$$

$$C = \text{the first } p \text{ rows of } Q.$$

This algorithm results in $\Sigma = (A, D_1, \cdots, D_r, B_1, \cdots, B_r, C)$, which is a realization of $\{W_j\}$.

4. Reachability, observability, and minimal realization. From the previous discussion we have seen that the matrices Q_n and T_n play an important role in the realization problem. In this section we will further show that they are closely connected with the reachability and the observability of (1.1).

Any $n \times l$ matrix G can be regarded as a set of column vectors in $\Re^n$. We denote the subspace generated by this set of vectors by span $\{G\}$. For (1.1) we use $x(t, x_0, u)$ and $y(t, x_0, u)$ to denote the state solution and output solution, respectively, with the initial state x_0 and the input $u(t)$.

DEFINITION 1. For (1.1), a state $\hat{x} \in \Re^n$ is called reachable if there is a time $T > 0$ and an input $u(t)$ such that $\hat{x} = x(T, 0, u)$.

It is well known that the set of all reachable states of a linear system forms a subspace of the state space. However, this is not generally true in the case we consider here because of the nonlinearity of (1.1). In order to consider the reachability, we define the reachable subspace for (1.1) to be the subspace generated by the set of all reachable states. We denote this subspace by S_R. The system (1.1) is called reachable if $S_R = \Re^n$.

DEFINITION 2. For system (1.1), a state $\hat{x} \in \Re^n$ is called unobservable if, for any two inputs u_1 and u_2, $y(t, \hat{x}, u_1) \equiv y(t, \hat{x}, u_2)$ for all $t \geqq 0$.

As in the case of linear systems, the set of all unobservable states of (1.1) forms a subspace of $\Re^n$, which we denote by S_N. The system (1.1) is called observable if $S_N = 0$ (zero space).

DEFINITION 3. The system (1.1) is called canonical if it is reachable and observable, i.e., $S_R = \Re^n$ and $S_N = 0$.

The main results we will prove in this section are that $S_R = \text{span}\{Q_n\}$ and $S_N = \ker(T_n)$. We will also show that the realization generated by using the algorithm in § 3 is minimal.

To do this, we first need to introduce the following concept.

DEFINITION 4. A subspace S of $\Re^n$ is called (A, D)-invariant if $AS \subset S$ and $D_iS \subset S$, $i = 1, 2, \cdots, r$.

For (1.1), let $S_1 = \text{span}\{P_1\}$. We denote the smallest (A, D)-invariant subspace containing S_1 by S_1^0. We first have Theorem 4.

THEOREM 4. $S_1^0 = \text{span}\{Q_n\}$.

Proof. We construct a series of subspaces of $\Re^n$ as follows:

$$S_1 = \text{span}\{P_1\},$$

$$S_j = S_{j-1} + AS_{j-1} + \sum_{i=1}^{r} D_iS_{j-1}, \qquad j \geqq 2.$$

It is clear that $S_{j-1} \subset S_j$ for all $j \geqq 2$. Since $\Re^n$ is a finite-dimensional linear space, there is an integer N, $0 < N \leqq n$, such that $S_N = S_{N+j}$, $j \geqq 1$. From this we therefore have

$$S_n = S_{n+1} = S_n + AS_n + \sum_{i=1}^{r} D_iS_n.$$

This means that $AS_n \subset S_n$ and $D_iS_n \subset S_n$, $i = 1, 2, \cdots, r$, i.e., S_n is (A, D)-invariant. Thus, we have $S_n \supset S_1^0$. On the other hand, we can also prove $S_n \subset S_1^0$. To do this, we first prove that if an (A, D)-invariant subspace K satisfies $S_j \subsetneqq K \subset S_{j+1}$, then $K = S_{j+1}$. In fact, since $S_{j+1} = S_j + AS_j + \sum_i D_iS_j$, $S_j \subset K$, and K is (A, D)-invariant, we have $S_{j+1} \subset K$. Thus, $S_{j+1} = K$. In general, we can further prove in the same way that, for any $j < l$, if K is an (A, D)-invariant subspace satisfying $S_j \subsetneqq K \subset S_l$, then $K = S_l$. By this fact and the observation $S_1 \subset S_1^0 \subset S_n$, we then have $S_n = S_1^0$.

To complete the proof, we must show that $S_n = \text{span}\{Q_n\}$. In fact, we can prove a more general result: $S_j = \text{span}\{Q_j\}, j \geqq 1$. To do this, we use induction. First observe that this equality is true for $j = 1$ from the definition of $S_1 = \text{span}\{P_1\} = \text{span}\{Q_1\}$. Assume that the equality is true for all positive integers less than j. We now prove it is true for j. From the induction hypothesis we can write

$$S_j = S_{j-1} + AS_{j-1} + \sum_i D_iS_{j-1}$$

$$= \text{span}\{P_1, \cdots, P_{j-1}\} + \text{span}\{AP_1, \cdots, AP_{j-1}\} + \sum_i \text{span}\{D_iP_1, \cdots, D_iP_{j-1}\}.$$

Since

$$\text{span}\{AP_1, \cdots, AP_{j-1}\} = \text{span}\{AP_1, \cdots, AP_{j-2}\} + \text{span}\{AP_{j-1}\},$$

$$\text{span}\{D_iP_1, \cdots, D_iP_{j-1}\} = \text{span}\{D_iP_1, \cdots, D_iP_{j-2}\} + \text{span}\{D_iP_{j-1}\},$$

and

$$\text{span}\{AP_{j-1}\} + \sum_i \text{span}\{D_iP_{j-1}\} = \text{span}\{AP_{j-1}, D_1P_{j-1}, \cdots, D_rP_{j-1}\}$$

$$= \text{span}\{P_j\},$$

we then have

$$S_j = \text{span}\{P_1, \cdots, P_{j-1}\} + \text{span}\{P_j\}$$

$$+ \text{span}\{AP_1, \cdots, AP_{j-2}\} + \sum_i \text{span}\{D_iP_1, \cdots, D_iP_{j-2}\}$$

$$= \text{span}\{P_1, \cdots, P_j\} + \text{span}\{AP_1, \cdots, AP_{j-2}\} + \sum_i \text{span}\{D_iP_1, \cdots, D_iP_{j-2}\}$$

$$= \text{span}\{P_{j-1}, P_j\} + \text{span}\{P_1, \cdots, P_{j-2}\} + \text{span}\{AP_1, \cdots, AP_{j-2}\}$$

$$+ \sum_i \text{span}\{D_iP_1, \cdots, D_iP_{j-2}\}$$

$$= \text{span}\{P_{j-1}, P_j\} + S_{j-1}$$

$$= \text{span}\{P_1, \cdots, P_j\}$$

$$= \text{span}\{Q_j\}. \qquad \square$$

To prove our first main result $S_R = \text{span}\{Q_n\}$, we now need only prove that $S_R = S_1^0$ and appeal to Theorem 4.

THEOREM 5. $S_R = S_1^0$.

Proof. For any real $c \in \Re$, $x(1, 0, c) = \sum_{i=1}^r c^k B_k \in S_R$. Taking r different real numbers $c_1, c_2, \cdots, c_r$, since the Vandermonde determinant

$$\begin{vmatrix} c_1 & c_2 & \cdots & c_r \\ c_1^2 & c_2^2 & \cdots & c_r^2 \\ & \cdots & \cdots & \\ c_1^r & c_2^r & \cdots & c_r^r \end{vmatrix}$$

is not zero, the set of vectors $[c_i, c_i^2, \cdots, c_i^r]$, $i = 1, 2, \cdots, r$, forms a basis in $\Re^r$. For any $[d_1, d_2, \cdots, d_r] \in \Re^r$, we therefore have

$$[d_1, d_2, \cdots, d_r] = \sum_{i=1}^{r} \lambda_i [c_i, c_i^2, \cdots, c_i^r],$$

i.e.,

$$d_k = \sum_{i=1}^{r} \lambda_i c_i^k, \qquad k = 1, 2, \cdots, r.$$

From this we have

$$\sum_{k=1}^{r} d_k B_k = \sum_k \left(\sum_i \lambda_i c_i^k \right) B_k = \sum_i \lambda_i \left(\sum_k c_i^k B_k \right) \in S_R$$

because $\sum_k c_i^k B_k \in S_R$. This means that $S_1 \subset S_R$.

A lengthy calculation shows that S_R is (A, D)-invariant. Therefore, $S_R \supset S_1^0$. On the other hand, for any $x \in S_R$, $x = \sum_{j=1}^{d} \lambda_j x_j$, where $x_1, \cdots, x_d$ forms a basis in S_R, since each x_j is reachable, we then have

$$\begin{aligned} x_j &= x(T_j, 0, u_j) \\ &= \sum_{i=1}^{T_j - 1} W(T_j - 1, \cdots, i) B(i-1) + B(T_j - 1), \end{aligned}$$

where

$$W(T_j - 1, \cdots, i) = \left(A + \sum_k u_j^k (T_j - 1) D_k \right) \left(A + \sum_k u_j^k (T_j - 2) D_k \right) \cdots \left(A + \sum_k u_j^k (i) D_k \right)$$

and

$$B(l) = \sum_k u^k(l) B_k, \qquad l = T_j - 1, \cdots, i - 1.$$

It is clear that $B(i-1)$ and $B(T_j - 1)$ are in S_1, and hence in S_R. Since S_R is (A, D)-invariant, x_j is in S_R. This means $S_R \subset S_1^0$. □

COROLLARY. *The system* (1.1) *is reachable if and only if* rank $(Q_n) = n$.

Now we prove the second main result, $S_N = \ker(T_n)$.

Let $C_1 = C^T$ and $C_j = [A^T C_{j-1}, D_1 C_{j-1}, \cdots, D_r C_{j-1}]$, $j \geqq 2$. From Theorem 4, we then have that the smallest (A^T, D^T)-invariant subspace containing $S_{C^T} = \text{span}\{C^T\}$ is just span $\{C_1, \cdots, C_n\}$. From the duality, we then have the following theorem.

THEOREM 6. *The largest* (A, D)*-invariant subspace contained in* ker (C) *is just* $(\text{span}\{C_1, \cdots, C_n\})^\perp$, *where* $\perp$ *denotes orthogonal complement.*

THEOREM 7. $S_N = \ker(T_n)$.

Proof. From Theorem 6 and the observation

$$(\text{span}\{C_1, \cdots, C_n\})^\perp = \ker(T_n),$$

to prove this theorem we need only to prove that $S_N = K$, where K is the largest (A, D)-invariant subspace contained in ker (C). If $x_0 \in K$, then Ax_0 and $D_i x_0$, $i = 1, 2, \cdots, r$ are in K also. A lengthy calculation shows that $x_0 \in S_N$. Therefore, $K \subset S_N$. On the other hand, for any $x_0 \in S_N$, we have $y(t, x_0, u_1) \equiv y(t, x_0, u_2)$. From this equality, we have $A^t x_0 \in \ker C$ and $D_i^t x_0 \in \ker C$ for all $t \geqq 0$. Thus,

$$CA^t x_0 = 0 \quad \text{for all } t \geqq 0,$$

and

$$CD_i^t x_0 = 0, \quad i = 1, 2, \cdots, r \quad \text{for all } t \geqq C.$$

These mean that $x_0 \in K$. Thus, $S_N \subset K$. □

COROLLARY. *The system* (1.1) *is observable if and only if* rank $(T_n) = n$.

We present our final theorem without proof, since it is entirely analogous to the linear case.

THEOREM 8. *If* $\{W_j\}$ *is realizable and* $\Sigma = (A, D_1, \cdots, D_r, B_1, \cdots, B_r, C)$ *is a realization of* $\{W_j\}$, *then* Σ *is minimal if and only if* Σ *is canonical.*

From this result, it is easy to see that the realization constructed by the algorithm given in § 3 is a minimal realization of $\{W_j\}$.

REFERENCES

[1] G. W. HAYNES AND H. HERMES, *Nonlinear controllability via Lie theory*, SIAM J. Control, 7 (1970), pp. 450–460.

[2] H. J. SUSSMANN AND V. JURDJEVIC, *Controllability of nonlinear systems*, J. Differential Equations, 12 (1972), pp. 95–116.

[3] V. JURDJEVIC AND H. J. SUSSMANN, *Control systems on Lie groups*, J. Differential Equations, 12 (1972), pp. 313–329.

[4] H. JIANXUN, *Chemical Engineering Department*, Tianjin University, P. R. China, personal communication.

[5] R. E. KALMAN, P. L. FALB, AND M. A. ARBIB, *Topics in Mathematical System Theory*, McGraw-Hill, New York, 1969.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 128–133, January 1990

009

ON CENTROHERMITIAN MATRICES*

RICHARD D. HILL†, RONALD G. BATES‡, AND STEVEN R. WATERS§

Abstract. A body of theory for centrohermitian and skew-centrohermitian matrices is developed. Some basic results for these matrices, their spectral properties, and characterizations of linear transformations that preserve them are given.

Key words. linear transformations, hermitian, spectral

AMS(MOS) subject classifications. 15A99, 15A04, 15A18, 15A15

1. Preliminaries. We denote the space of $n \times n$ complex matrices by $\mathcal{M}_n$ and the subset of hermitian [skew-hermitian] matrices by $\mathcal{H}_n$ [$\mathcal{H}_n^-$]. A matrix $A \in \mathcal{M}_n$ is said to be *centrohermitian* [*skew-centrohermitian*] if and only if $a_{ij} = \bar{a}_{n-i+1,n-j+1}$ [$a_{ij} = -\bar{a}_{n-i+1,n-j+1}$] $i, j = 1, \cdots, n$. We denote the set of centrohermitian [skew-centrohermitian] matrices by $\mathcal{CH}_n$ [$\mathcal{CH}_n^-$]. A matrix $A \in \mathcal{M}_n$ is said to be *perhermitian* [*skew-perhermitian*] if and only if $a_{ij} = \bar{a}_{n-j+1,n-i+1}$ [$a_{ij} = -\bar{a}_{n-j+1,n-i+1}$] $i, j = 1, \cdots, n$. We denote the set of perhermitian [skew-perhermitian] matrices by $\mathcal{PH}_n$[$\mathcal{PH}_n^-$]. A matrix $A \in \mathcal{M}_n$ is said to be *perdiagonal* if and only if $a_{ij} = 0$ whenever $i + j \neq n + 1$, $i, j = 1, \cdots, n$. In particular, we shall use $J = (\delta_{i,n-j+1})$ to denote the unit perdiagonal matrix that has 1's on the secondary diagonal (i.e., the diagonal from upper-right to lower-left) and 0's elsewhere.

The concept of a centrosymmetric matrix in the context of its determinant goes back to Muir [13] and Aitken [1]. Ligh [12] attributes the idea to Zehfuss [20] in 1862. The investigation of centrosymmetric matrices by Good [6] and Ray [17] was motivated by the study of certain Toeplitz matrices. Cruse [4] encountered the group of $n \times n$ centrosymmetric permutation matrices in his study of problems from combinatorial theory.

While many papers have discussed bits and pieces of basic theory for centrosymmetric matrices (cf. [3], [6], [12], [16], [18], and [19]), we observe with Lee [11] that centrohermitian matrices have received very little attention. Whereas the notions of centrosymmetric and centrohermitian agree for matrices with real entries, they yield quite different theories in the complex case. The centrohermitian concept appears to be a more natural generalization of real centrosymmetric, just as hermitian appears to be a more natural generalization of real symmetric.

In this paper we develop a body of theory on centrohermitian and skew-centrohermitian matrices. In particular, we develop some basic results for these matrices, discuss their interface with the perhermitian matrices, consider their spectral properties, and characterize linear transformations that leave the set of centrohermitian matrices invariant. This theory at times parallels and at times is quite different from that in our companion paper [8] for perhermitian matrices.

2. Basic results. In this section we shall enumerate many of the basic facts concerning centrohermitian [skew-centrohermitian] matrices beginning with a characterization.

2.1. For $A \in \mathcal{M}_n$, the following are equivalent:

(i) $A \in \mathcal{CH}_n$ [$A \in \mathcal{CH}_n^-$]

(ii) $A = J\bar{A}J$ [$A = -J\bar{A}J$]

* Received by the editors September 6, 1988; accepted for publication (in revised form) March 6, 1989.

† Department of Mathematics, Idaho State University, Pocatello, Idaho 83201.

‡ Department of Mathematics, Hartnell College, Salinas, California 93901.

§ Department of Mathematics, Pacific Union College, Angwin, California 94508.

(iii) $JA \in \mathscr{CH}_n$ $[JA \in \mathscr{CH}^-_n]$

(iv) $AJ \in \mathscr{CH}_n$ $[AJ \in \mathscr{CH}^-_n]$

(v) $iA \in \mathscr{CH}^-_n$ $[iA \in \mathscr{CH}_n]$

We observe that if $A \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$, then so are $\bar{A}$, A^*, and A^{tr}. Also, by (ii), a centrohermitian matrix A is seen to be unitarily similar to its conjugate, $\bar{A}$. Further, by (v), results for skew-centrohermitian matrices may be immediately obtained from those for centrohermitian matrices, and conversely.

2.2. If $A_1, \cdots, A_s \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$ and $c_1, \cdots, c_s \in \mathbb{R}$, then $\sum_{j=1}^{s} c_j A_j \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$. It follows that $\mathscr{CH}_n$ and $\mathscr{CH}^-_n$ are both real vector spaces.

2.3. When n is even, letting $j = 1, \cdots, n/2$ and $k = 1, \cdots, n$, we have that the $n^2/2$ matrices $E_{jk} + E_{n-j+1,n-k+1}$ and the $n^2/2$ matrices $iE_{jk} - iE_{n-j+1,n-k+1}$, form a basis for $\mathscr{CH}_n$ over $\mathbb{R}$. When n is odd, letting $j = 1, \cdots, (n-1)/2$, $k = 1, \cdots, n$, and $l = (n+1)/2$, we have that the $(n(n-1))/2$ matrices $E_{jk} + E_{n-j+1,n-k+1}$, the $(n(n-1))/2$ matrices $iE_{jk} - iE_{n-j+1,n-k+1}$, the $(n-1)/2$ matrices $E_{l,j} + E_{l,n-j+1}$, the $(n-1)/2$ matrices $iE_{l,j} - iE_{l,n-j+1}$, and the 1 matrix $E_{l,l}$ form a basis for $\mathscr{CH}_n$ over $\mathbb{R}$. Thus, $\mathscr{CH}_n$ is of dimension n^2 as a real vector space. Multiplying each of the above matrices by i, we obtain a basis of n^2 matrices for the real vector space of skew-centrohermitian matrices.

2.4. If A is perdiagonal, then $\bar{A}A, A\bar{A} \in \mathscr{CH}_n$.

2.5. If $A, B \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$, then $AB \in \mathscr{CH}_n$. It follows that the set of centrohermitian matrices forms an algebra. While the set of centrosymmetric matrices also forms an algebra (cf. Thm. 7 of [19]), the set of perhermitian matrices does not (cf. result 2.5 of [8]).

2.6. If $A \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$ and $B \in \mathscr{CH}^-_n$ $[\mathscr{CH}_n]$, then $AB \in \mathscr{CH}^-_n$.

2.7. If $A \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$ and A is nonsingular, then $A^{-1} \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$. In conjunction with 2.5 and 2.6, it follows that whenever defined, all integer powers of centrohermitian matrices are centrohermitian, and integer powers of skew-centrohermitian matrices are either centrohermitian (even powers) or skew-centrohermitian (odd powers).

Since $AA^* \in \mathscr{CH}_n$ for all $A \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$, and the Moore–Penrose inverse of A can be written as $A^+ = A^* p(AA^*)$ for some polynomial p with real coefficients (cf. p. 526 of [5]), we immediately obtain the following.

2.8. If $A \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$, then $A^+ \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$.

2.9. If $A \in \mathscr{CH}_n$, then the determinant of A, $\det A$, is real. If $A \in \mathscr{CH}^-_n$, then $\det A$ is real [pure imaginary] if n is even [odd].

2.10. If $A \in \mathscr{CH}_n$, then the adjoint of A, $\operatorname{adj} A \in \mathscr{CH}_n$. If $A \in \mathscr{CH}^-_n$, then $\operatorname{adj} A \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$ if n is odd [even].

2.11. If $A \in \mathscr{M}_n$, then there exist unique $P, Q \in \mathscr{CH}_n$ such that $A = P + iQ$. For this result, $P = \frac{1}{2}(A + J\bar{A}J)$ and $Q = (1/2i)(A - J\bar{A}J)$. Note that this parallels the Toeplitz (Cartesian) decomposition $A = H + iK$ with unique $H, K \in \mathscr{H}_n$ and the decomposition 2.11 of [8].

Our next result relates principal submatrices of a centrohermitian or skew-centrohermitian matrix with principal submatrices of its conjugate. Note that $A[p_1, \cdots, p_s | p_1, \cdots, p_s]$ denotes the principal submatrix of A which retains both the rows and columns indexed by $p_1, \cdots, p_s$.

2.12. If $A \in \mathscr{CH}_n$ $[\mathscr{CH}^-_n]$, then for $s = 1, \cdots, n$, we have

$$A[p_1, \cdots, p_s | p_1, \cdots, p_s] = J\bar{A}[n-p_s+1, \cdots, n-p_1+1 | n-p_s+1, \cdots, n-p_1+1]J$$

$$[= -J\bar{A}[n-p_s+1, \cdots, n-p_1+1 | n-p_s+1, \cdots, n-p_1+1]J].$$

Taking determinants we immediately get relationships between the corresponding minors. In particular, the sum of all principal minors of size s for $A \in \mathscr{CH}_n$ must be real.

An interesting relationship exists among the real vector spaces of centrohermitian, perhermitian, and symmetric matrices; viz., the intersection of any two of these sets is contained in the third. Restating, we have the following.

2.13. If $A \in \mathscr{M}_n$, then any two of the following three conditions imply the third.

(1) $A \in \mathscr{CH}_n$ [$\mathscr{CH}_n^-$]

(2) $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$]

(3) A is symmetric.

The analogous result for centrosymmetric and persymmetric matrices also holds.

3. Spectral results. Since the coefficients of the characteristic polynomial for a matrix are sums of principal minors of the matrix multiplied by ± 1 (cf. p. 157 of [10]), result 2.12 yields the following.

3.1. If $A \in \mathscr{CH}_n$, then the characteristic polynomial of A has all real coefficients.

3.2. If $A \in \mathscr{CH}_n$ [$\mathscr{CH}_n^-$] has an eigenvalue λ of algebraic multiplicity k, then A must also have $\bar{\lambda}$ [$-\bar{\lambda}$] as an eigenvalue of algebraic multiplicity k.

Hermitian matrices are known to be similar to real matrices (cf. Theorem 2 of [2]) as are perhermitian matrices (cf. result 3.3 of [8]). The next result, which is due to Lee [11], gives us that all centrohermitian matrices are *simultaneously* similar to real matrices. Lee observes that any nonsingular matrix Q satisfying $JQ = \bar{Q}$ will give the desired transformation and that infinitely many such Q's exist.

3.3. There exists a nonsingular matrix $Q \in \mathscr{M}_n$ for which $A \in \mathscr{CH}_n$ [$\mathscr{CH}_n^-$] if and only if $Q^{-1}AQ \in \mathscr{M}_n(\mathbb{R})$ [$\mathscr{M}_n(i\mathbb{R})$].

3.4. If $A \in \mathscr{CH}_n$, then all the elementary symmetric functions of A are real.

As in the perhermitian case, we find that there is no proper subset of the complex plane that contains the spectrum of all centrohermitian matrices. Indeed, given any number $z \in \mathbb{C}$, the matrix diag $(z, \bar{z})$ is centrohermitian with z as an eigenvalue. Clearly, a similar construction gives matrix examples of any order greater than 2. Also, centrohermitian matrices need not be normal, nor even diagonalizable (e.g., $E_{12} + E_{32} \in \mathscr{M}_3$).

Weaver [19] has studied the eigenvector structure for centrosymmetric matrices and has shown that (λ, x) is an eigenvalue, eigenvector pair of a centrosymmetric matrix if and only if (λ, Jx) is also. The corresponding result for centrohermitian matrices is as follows.

3.5. If $A \in \mathscr{CH}_n$, then (λ, x) is an eigenvalue, eigenvector pair of A if and only if $(\bar{\lambda}, J\bar{x})$ is also.

4. Centrohermitian-preserving linear transformations. We now address the problem of characterizing centrohermitian-preserving linear transformations; i.e., linear transformations on $\mathscr{M}_n$ that leave $\mathscr{CH}_n$ invariant. We utilize the notation of Poluikis and Hill [15] and Oxenrider and Hill [14]. Two bijections are defined from $\{(i,j): i,j = 1, \cdots, n\}$ to $\{1, \cdots, n^2\}$ by $[i,j] = (i-1)n + j$ and $\langle i,j \rangle = (j-1)n + i$. These correspond to the lexicographical ordering ($(i,j) < (r,s)$ if and only if $i < r$ or ($i = r$ and $j < s$)) and the antilexicographical ordering ($(i,j) < (r,s)$ if and only if $j < s$ or ($j = s$ and $i < r$)), respectively, on $\{(i,j): i,j = 1, \cdots, n\}$.

The following equalities, which may be verified by calculation, will prove useful in establishing many of the equivalences in the characterization theorem.

$$(*) \qquad \begin{aligned} n^2 - [k,l] + 1 &= [n-k+1, n-l+1] \\ n^2 - \langle k,l \rangle + 1 &= \langle n-k+1, n-l+1 \rangle \end{aligned} \qquad k,l = 1, \cdots, n.$$

As in [15], if $\mathscr{T}$ is a linear transformation on $\mathscr{M}_n$, then we let $\langle \mathscr{T} \rangle \in \mathscr{M}_{n^2}$ be the matrix representation of $\mathscr{T}$ with respect to the basis of unit matrices $\{E_{ij}\}_{i,j=1,\cdots,n} \subset \mathscr{M}_n$ ordered antilexicographically. Intuitively this order may be thought of as transforming a matrix $A \in \mathscr{M}_n$ into vec $A \in \mathbb{C}^{n^2}$ by stacking the columns of A into one big column vector (cf. [9] and [10]). We then have vec $\mathscr{T}(A) = \langle \mathscr{T} \rangle$ vec A.

It is also useful to write $T \in \mathscr{M}_{n^2}$ in the block form $T = (T_{ij}) \in \mathscr{M}_n(\mathscr{M}_n)$, where $T_{ij} = (t_{rs}^{ij}) \in \mathscr{M}_n (i, j, r, s = 1, \cdots, n)$. With this notation, we note that T is centrohermitian if and only if $t_{rs}^{ij} = \bar{t}_{n-r+1,n-s+1}^{n-i+1,n-j+1}$ $(i, j, r, s = 1, \cdots, n)$.

Oxenrider and Hill [14] have studied eight element reorderings of matrices in $\mathscr{M}_n(\mathscr{M}_n)$ that naturally arise from rearranging the rows or columns, lexicographically or antilexicographically, into $n \times n$ blocks ordered lexicographically or antilexicographically. These reorderings are defined by

$$\Gamma(T)_{rs}^{ij} = t_{[i,j],[r,s]}, \quad \Psi(T)_{rs}^{ij} = t_{\langle r,s\rangle,\langle i,j\rangle}, \quad \Xi(T)_{rs}^{ij} = t_{[i,j],\langle r,s\rangle}, \quad \Upsilon(T)_{rs}^{ij} = t_{\langle i,j\rangle,[r,s]},$$
$$\Theta(T)_{rs}^{ij} = t_{\langle i,j\rangle,\langle r,s\rangle}, \quad \Lambda(T)_{rs}^{ij} = t_{\langle r,s\rangle,[i,j]}, \quad \Delta(T)_{rs}^{ij} = t_{[r,s]\langle i,j\rangle}, \quad \Omega(T)_{rs}^{ij} = t_{[r,s],[i,j]}.$$

As noted in [14], these reorderings do not preserve the matrix properties of determinant, rank, and trace, nor do they preserve normal, hermitian, or perhermitian matrices. Also, only Γ, Ψ, Ω, and Θ yield characterizations of hermitian-preserving and perhermitian-preserving linear transformations (cf. [15] and [8]). It is remarkable then, that all eight reorderings preserve centrohermitian matrices and give characterizations of centrohermitian-preserving linear transformations.

We now state the main result of this section, noting the similarity to Theorems 1 and 2 of [15] and Theorem 4.1 of [8].

THEOREM 4.1. *Let $\mathscr{T}$ be a linear transformation on $\mathscr{M}_n$. Then the following are equivalent*:

(1) *$\mathscr{T}$ is centrohermitian preserving.*

(2) *$\mathscr{T}$ is skew-centrohermitian preserving.*

(3) *There exist $A_1, \cdots, A_t \in \mathscr{M}_n$ with $JA_kJ = \bar{A}_{t-k+1}$ $(k = 1, \cdots, t)$ and $G = (g_{ij}) \in \mathscr{CH}_t$ for which*

$$\mathscr{T}(X) = \sum_{i,j=1}^{t} g_{ij} A_i X A_j^*.$$

(4) $t_{rs}^{ij} = \bar{t}_{n-r+1,n-s+1}^{n-i+1,n-j+1}$ $(i, j, r, s = 1, \cdots, n)$ *where* $\langle \mathscr{T} \rangle = ((t_{rs}^{ij}))$.

(5) $\langle \mathscr{T} \rangle$ *is centrohermitian.*

(6) $\Gamma(\langle \mathscr{T} \rangle)$ *is centrohermitian.*

(7) $\Psi(\langle \mathscr{T} \rangle)$ *is centrohermitian.*

(8) $\Omega(\langle \mathscr{T} \rangle)$ $(=\Gamma(\langle \mathscr{T} \rangle^{\mathrm{tr}}))$ *is centrohermitian.*

(9) $\Theta(\langle \mathscr{T} \rangle)(=\Psi(\langle \mathscr{T} \rangle^{\mathrm{tr}}))$ *is centrohermitian.*

(10) $\Upsilon(\langle \mathscr{T} \rangle)$ *is centrohermitian.*

(11) $\Xi(\langle \mathscr{T} \rangle)$ *is centrohermitian.*

(12) $\Lambda(\langle \mathscr{T} \rangle)$ *is centrohermitian.*

(13) $\Delta(\langle \mathscr{T} \rangle)$ *is centrohermitian.*

(14) *The block matrix $(\mathscr{T}(E_{ij}))$ is centrohermitian.*

(15) *$\mathscr{T}^*$ is centrohermitian preserving.*

Proof. Result 2.1(v) immediately gives (1) $\Leftrightarrow$ (2).

Using a proof technique analogous to Theorem 1 of [7] (viz. by computing $\mathscr{T}(B)$ for each of the basis elements in result 2.3, and forcing these to be centrohermitian) we get (1) $\Leftrightarrow$ (4).

The equivalence of (4) and (5) follows immediately from the definition of centrohermitian and the fact that $t_{rs}^{ij} = t_{(i-1)n+r,(j-1)n+s}$.

Making use of (*) and the result analogous to (4) $\Leftrightarrow$ (5), we have that $\Delta(\langle \mathscr{T} \rangle)$ is centrohermitian

$$\text{iff } \Delta(\langle \mathscr{T} \rangle)_{rs}^{ij} = \overline{\Delta(\langle \mathscr{T} \rangle)}_{n-r+1,n-s+1}^{n-i+1,n-j+1} \qquad (i,j,r,s = 1, \cdots, n)$$

$$\text{iff } t_{[r,s],\langle i,j\rangle} = \bar{t}_{[n-r+1,n-s+1],\langle n-i+1,n-j+1\rangle} \qquad (i,j,r,s = 1, \cdots, n)$$

$$\text{iff } t_{[r,s],\langle i,j\rangle} = \bar{t}_{n^2-[r,s]+1,n^2-\langle i,j\rangle+1} \qquad (i,j,r,s = 1, \cdots, n)$$

$$\text{iff } \langle \mathscr{T} \rangle \text{ is centrohermitian,}$$

thus establishing (13) $\Leftrightarrow$ (5). A similar argument establishes the equivalences (6) $\Leftrightarrow$ (5) through (12) $\Leftrightarrow$ (5).

By Lemma 2 of [15] we have that the block matrix $(\mathscr{T}(E_{ij})) = \Psi(\langle \mathscr{T} \rangle)$, which gives (7) $\Leftrightarrow$ (14). Also, since $\{E_{ij}\}$ is an orthonormal basis for $\mathscr{M}_n$, we have that the matrix representation of the Hilbert adjoint of $\mathscr{T}$ is $\langle \mathscr{T}^* \rangle = \langle \mathscr{T} \rangle^*$, thus yielding (1) $\Leftrightarrow$ (15).

For (3) $\Rightarrow$ (1), suppose that $\mathscr{T}$ has the form indicated in (3) and that C is centrohermitian. We then have

$$\begin{aligned} J(\overline{\mathscr{T}(C)})J &= \sum_{i,j=1}^{t} \bar{g}_{ij}(J\bar{A}_iJ)(J\bar{C}J)(J\bar{A}_jJ)^* \\ &= \sum_{i,j=1}^{t} g_{t-i+1,t-j+1}A_{t-i+1}CA_{t-j+1}^* \\ &= \sum_{i,j=1}^{t} g_{ij}A_iCA_j^* \end{aligned}$$

yielding $\mathscr{T}(C)$ centrohermitian whenever C is; thus, $\mathscr{T}$ is centrohermitian preserving.

For (5) $\Rightarrow$ (3), let $\langle \mathscr{T} \rangle$ be centrohermitian. Then by (5) $\Leftrightarrow$ (7), $\Psi^{-1}(\langle \mathscr{T} \rangle)$ is also centrohermitian. Letting $t = n^2$, $G = \Psi^{-1}(\langle \mathscr{T} \rangle)$, and $A_1, \cdots, A_t$ be the unit matrices E_{ij} ordered antilexicographically (i.e., $A_{\langle i,j\rangle} = E_{ij}$), we have by (*) that $JA_kJ = A_{t-k+1}$ $(k = 1, \cdots, t)$ and

$$\langle \mathscr{T} \rangle = \sum_{i,j=1}^{t} g_{ij}A_j \otimes A_i$$

where $\otimes$ denotes the Kronecker product. Since $E_{ij}^* = E_{ij}^{\text{tr}}$, Proposition 12.1.4 of [10] gives us that

$$\mathscr{T}(X) = \sum_{i,j=1}^{t} g_{ij}A_iXA_j^*$$

and the result is established. □

Acknowledgment. The authors thank Professor E. E. Underwood, Utah State University, for suggesting the topics of centrosymmetric (centrohermitian) and persymmetric (perhermitian) matrices.

REFERENCES

[1] A. C. AITKEN, *Determinants and Matrices*, Interscience Publishers, New York, 1939.

[2] D. H. CARLSON, *On real eigenvalues of complex matrices*, Pacific J. Math., 15 (1965), pp. 1119–1129.

[3] A. R. COLLAR, *On centrosymmetric and centro-skew matrices*, Quart. J. Mech. Appl. Math., 25 (1962), pp. 265–281.
[4] A. B. CRUSE, *Some combinatorial properties of centrosymmetric matrices*, Linear Algebra Appl., 16 (1977), pp. 65–77.
[5] H. P. DECELL, JR., *An application of the Cayley-Hamilton theorem to generalized matrix inversion*, SIAM Rev., 7 (1965), pp. 526–528.
[6] I. J. GOOD, *The inverse of a centrosymmetric matrix*, Technometrics, 12 (1970), pp. 925–928.
[7] R. D. HILL, *Linear transformations which preserve hermitian matrices*, Linear Algebra Appl., 6 (1973), pp. 257–262.
[8] R. D. HILL, R. G. BATES, AND S. R. WATERS, *On perhermitian matrices*, SIAM J. Math Anal. Appl., 11 (1990), to appear.
[9] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, to appear.
[10] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, Second edition, Academic Press, Orlando, 1985.
[11] A. LEE, *Centrohermitian and skew-centrohermitian matrices*, Linear Algebra Appl., 29 (1980), pp. 205–210.
[12] S. LIGH, *Centro-symmetric matrices*, Delta, 3 (1972/73), pp. 33–37.
[13] T. MUIR, *Theory of Determinants*, Longmans, Green and Company, 1933.
[14] C. J. OXENRIDER AND R. D. HILL, *On the matrix reorderings* Γ *and* Ψ, Linear Algebra Appl., 69 (1985), pp. 205–212.
[15] J. A. POLUIKIS AND R. D. HILL, *Completely positive and hermitian-preserving linear transformations*, Linear Algebra Appl., 35 (1981), pp. 1–10.
[16] W. D. PYE, T. L. BOULLION, AND T. A. ATCHISON, *The pseudoinverse of a centrosymmetric matrix*, Linear Algebra Appl., 6 (1973), pp. 201–204.
[17] W. D. RAY, *The inverse of a finite Toeplitz matrix*, Technometrics, 12 (1970), pp. 153–156.
[18] P. A. ROEBUCK AND S. BARNETT, *A survey of Toeplitz and related matrices*, Internat. J. Systems Sci., 9 (1978), pp. 921–934.
[19] J. R. WEAVER, *Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues, and eigenvectors*, Amer. Math. Monthly, 92 (1985), pp. 711–717.
[20] G. ZEHFUSS, *Zwei sätze über determinanten*, Z. Math. Phys., 7 (1862), pp. 436–439.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 1, pp. 134–172, January 1990

010

THE ROLE OF ELIMINATION TREES IN SPARSE FACTORIZATION*

JOSEPH W.H. LIU†

Abstract. In this paper, the role of elimination trees in the direct solution of large sparse linear systems is examined. The notion of elimination trees is described and its relation to sparse Cholesky factorization is discussed. The use of elimination trees in the various phases of direct factorization are surveyed: in reordering, sparse storage schemes, symbolic factorization, numeric factorization, and different computing environments.

Key words. elimination tree, sparse matrix, Cholesky factor, reordering, symbolic factorization, numeric factorization

AMS(MOS) subject classifications. 65F50, 65F25

1. Introduction.

The elimination tree plays an important role in many aspects of sparse matrix factorization. It provides structural information relevant to the sparse factorization process. The purpose of this paper is to provide a unified study of this important structure and to survey its uses in various phases of sparse factorization.

Throughout this paper, unless otherwise specified, A is assumed to be a large sparse n-by-n symmetric positive-definite matrix. We consider direct methods for the solution of the linear system

$$A x = b .$$

The matrix A is factored into LL^T, where L is lower triangular and is the Cholesky factor of A. The solution vector x is then obtained by forward and backward substitution using L. The elimination tree of A is defined using the structure of the Cholesky factor L of A. It can therefore be characterized using only the structure of the given matrix A.

The structure of an elimination tree was used implicitly long before its importance was recognized. The term elimination tree was used by Duff [6], although the actual structure studied by him is slightly different from the one in this paper. Jess and Kees [34] use this term to refer to a tree structure introduced by them for studying parallel elimination. Their structure is, in fact, a special case of the elimination tree studied here. Schreiber [55] is perhaps the first one to formally define the elimination tree structure. In [36], Liu uses the term "elimination tree" to refer to the structure introduced by Schreiber. It is this tree structure that we are going to consider in this paper.

Variants of the basic elimination tree structure appear in the literature under different names. It is used as a *dissection tree* to study nested dissection [18], [31] in the context of optimal sparse matrix reordering. The *element merge tree* introduced by Eisenstat, Schultz, and Sherman [13] in their element model has a structure close to the elimination tree. Duff and Reid [10] use an *assembly tree* to determine the assembly order in the multifrontal method, and its structure is a generalized version

* Received by the editors December 10, 1987; accepted for publication (in revised form) May 10, 1989. This research was partially supported by the Natural Sciences and Engineering Research Council of Canada under grant A5509, by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems Inc., and by the U.S. Air Force Office of Scientific Research under contract AFOSR-ISSA-86-00012.

† Department of Computer Science, York University, North York, Ontario, Canada M3J 1P3 (joseph@yuyetti.bitnet or NA.JLIU@na-net.stanford.edu).

of the elimination tree. The *row merge tree* used by Liu in [38] can be viewed as an extension of the elimination tree for the study of sparse orthogonal factorization. This paper gives a unified treatment of this structure, which has proved to be a valuable tool in sparse elimination.

The reader is assumed to be familiar with the graph-theoretic terminology used in the study of sparse elimination: fill, ordering and permutation, elimination graph model, chordal graphs, and other related concepts. All the necessary material can be found in [8] and [22]. Moreover, the basic terminology and concepts in the study of trees will be assumed: parent/child, ancestor/descendant, paths, root, subtree, leaf, and others. The reader is referred to [1].

In this survey paper, we use many results that have already appeared in the literature. Such results are quoted and properly referenced, and their proofs are usually omitted. However, results that are generally known among sparse matrix researchers, but have not been formally dealt with, are treated in greater detail. This is to provide a good foundation for future works on elimination trees.

Another objective of this paper is to point out the relevance of the elimination tree in many existing sparse algorithms. The connections may not be stated in previous discussions of these algorithms in the literature. Here, we provide the direct link to this tree structure. The paper also contains some new results on the elimination tree, one of which is its use in the intersection graph representation of chordal graphs in §4.2.

An outline of this paper follows. In §2, the notion of an elimination tree is formally defined in terms of the Cholesky factor L of a sparse matrix A. A simple matrix example is introduced, and it will be used throughout the remainder of this paper. We also offer two new views of the elimination tree: one as the transitive reduction of the directed graph of L, and another as a depth-first search tree of the filled graph.

Section 3 provides properties of the elimination tree that are relevant to the sparse Cholesky factor L. Nonzeros in the factor matrix can be characterized in terms of paths in the elimination tree. Both the row and column structures of L can also be expressed in terms of structures of this tree. Section 4 gives some observations on the use of elimination trees for chordal graphs.

In §5, we consider efficient ways of determining the elimination tree for a given sparse matrix A. An efficient algorithm can be formulated in terms of the basic set union operations, introduced by Tarjan [58]. Some experimental results are provided to offer some practical observations in this regard.

Sections 6–10 examine the role of the elimination tree in various aspects and different phases of sparse factorization. Section 6 considers the use of elimination trees for finding equivalent matrix reorderings. It includes reorderings that preserve the elimination tree structure, and those that completely restructure it. In §7, we study various sparse storage schemes that makes use of elimination tree structures.

Section 8 examines the connection of the elimination tree with the symbolic factorization phase. Existing symbolic factorization algorithms are shown to use this tree structure implicitly. Many numerical factorization schemes also use this tree structure. In §9, we consider its role in the multifrontal method, the minimal storage scheme, the general row merging scheme, and sparse indefinite factorization. In specific computing environments, the elimination tree can also be used to improve on the basic factorization scheme, and we consider it in §10. Finally, §11 contains some remarks on future research directions.

2. On elimination trees.

2.1. Trees in sparse matrices. Undirected graphs are useful tools in the study of symmetric matrices. See, for example, [8], [22], and [52]. A given symmetric sparse matrix A can be structurally represented by its associated graph $G(A) = (X(A), E(A))$, where nodes in $X(A)$ correspond to rows and columns of the matrix and edges in $E(A)$ correspond to nonzero entries. We shall use $\{u, v\}$ to indicate an edge between two nodes u and v.

In this section, we briefly review properties of sparse matrices whose associated graphs are *trees*, in preparation for the introduction of elimination trees in the next section. Without loss of generality, we assume that the given matrix is irreducible so that its graph is connected. Let A be an n-by-n symmetric positive-definite irreducible matrix and let $G(A)$ be its associated graph, which is a tree. It is well known that such a matrix A is a *perfect elimination matrix*; that is, it has a permutation P such that the permuted matrix PAP^T does not suffer any fill in its Cholesky factorization [52].

Orderings that will produce no extra fill are sometimes referred to as *perfect elimination orderings* [52]. Such orderings for tree structures are easy to obtain. Indeed, the popular *minimum degree ordering* on $G(A)$ will be appropriate. Or, more simply, take any node in the associated tree as the root, any ordering that numbers children nodes before their parent node will introduce no fill. In particular, a postordering [1] of the rooted tree will be one such ordering.

For a given rooted tree, we define a *topological ordering* of the tree to be an ordering that numbers children nodes before their parent node. This is consistent with the notion of topological orderings for directed acyclic graphs (directed graphs without cycles) used in the literature [58]. A topological ordering of a directed acyclic graph is one such that for every directed edge from a node u to v, u is ordered before v. In the case of a rooted tree, if we treat each tree edge as a directed edge that goes from a child to its parent, our definition of a topological ordering of a rooted tree is the same as that used for directed acyclic graphs.

Assume the matrix A (whose associated graph is a tree) has already been ordered by a topological ordering. Let $x_1, x_2, \cdots, x_n$ denote the nodes in the associated graph/tree, where node x_j corresponds to the jth row/column of the matrix A. Note in the topological ordering, except the root, each node at its elimination is connected to only one uneliminated node (namely, its parent). It follows then that in the lower triangular part of A, each column has exactly *one* off-diagonal nonzero except the last column.

Such matrix structure can be represented by the subscripts of the off-diagonal nonzeros in the columns. For each column $j < n$ of A, let

$$PARENT[j] = p, \text{ where } a_{pj} \neq 0 \text{ and } p > j.$$

For completeness, we let $PARENT[n] = 0$. The function $PARENT[*]$ uniquely characterizes the associated tree $G(A)$ of A. If we consider the tree $G(A)$ as rooted at x_n, for each node x_j other than the root, its parent node in the tree is given by $x_{PARENT[j]}$.

2.2. Notion of elimination trees. Trees form a class of data structure that is easy to store and manipulate. However, in practice, it is rare that sparse matrices have associated graphs in the form of trees. In this section, for a general symmetric positive-definite sparse matrix, we introduce a tree structure that is useful in the study of sparse factorization. It may be viewed as a generalization of the tree in §2.1.

Let A be a given n-by-n sparse symmetric positive-definite irreducible matrix. Consider its Cholesky factorization $A = LL^T$. As before, let $G(A)$ be the undirected graph associated with A, and let $x_1, x_2, \cdots, x_n$ be the sequence of nodes. Moreover, let $G(F)$ be the associated filled graph, where $F = L + L^T$ is the filled matrix of A. It is well known that $G(F)$ has the same set of nodes and is a supergraph of $G(A)$.

Consider the Cholesky factor L and the filled matrix F of A. Since the matrix A is irreducible, it can be readily verified that each of the first $n-1$ columns of L has at least one off-diagonal nonzero. For each column $j < n$ of L, remove all the nonzeros in this column except the first nonzero below the diagonal. Let L_t be the resulting matrix and $F_t = L_t + L_t^T$. The graph $G(F_t)$ is a tree structure, and it depends entirely on the structure of the original sparse matrix A and its initial ordering. We use $T(A)$ to denote this tree structure and refer to it as the *elimination tree* of A.

An example is given in Fig. 2.1, which illustrates the structures of the matrices A, F, and F_t. Each diagonal entry is labeled by the corresponding node in the graph. Off-diagonal nonzeros are indicated by "•" while "∘" is used to denote a fill in the matrix. The corresponding graphs of $G(A)$, $G(F)$, and $T(A)$ are given in Fig. 2.2. A dotted line in $G(F)$ is used to indicate a filled edge in the graph. Note that some of the tree edges in $T(A)$ are filled edges of $G(F)$ (for example, the tree edge between nodes g and h). This same example will be used throughout the remainder of this paper.

The elimination tree $T(A)$ has the same node set as $G(A)$ and is a *spanning tree* [1] of the filled graph $G(F)$ of A. We define the node x_n to be the root of this tree $T(A)$. The elimination tree structure can be conveniently represented by the $PARENT[*]$ vector of F_t. In terms of the triangular factor L, we have, for $j < n$,

$$PARENT[j] = \min \{ i > j \mid \ell_{ij} \neq 0 \}.$$

It should be pointed out that this is a generalization of the $PARENT[*]$ vector introduced in §2.1; since when $G(A)$ is a tree and is ordered with no fill, we have $a_{ij} \neq 0$ if and only if $\ell_{ij} \neq 0$. Indeed, in this case, $G(A) = G(F) = T(A)$.

In general, $G(A)$ is a subgraph of $G(F)$ due to fills. However, the two elimination trees $T(A)$ and $T(F)$ are identical (since F is a perfect elimination matrix with no additional fill). Often, we simply use T to refer to this tree. The next result contains another simple property of the elimination tree. It follows directly from the definition of the $PARENT$ vector.

PROPOSITION 2.1. *If x_i is a proper ancestor node of x_j in the elimination tree, then $i > j$.* □

To facilitate discussions in subsequent sections, we introduce the subtree notation. We use $T[x]$ to represent the subtree of $T(A)$ rooted at the node x, which includes all descendants of x in the tree T. Since x_n is the root of the tree, we have $T(A) = T[x_n]$. If $y \in T[x]$, then the node y is a *descendant* of x, and x an *ancestor* of y. Note that every node is an ancestor and a descendant of itself.

To simplify discussion, we shall use $T[x]$ to denote both the subtree itself and the set of nodes in this subtree. This means that, on the one hand, $T[x]$ is a subtree of the elimination tree $T(A)$ and, on the other hand, the node set $T[x]$ defines a subgraph of the graph $G(A)$. No attempt shall be made to distinguish between a set of nodes of $G(A)$ and the subgraph of $G(A)$ that it induces. That is, we shall use $T[x]$ as a node subset and as a subgraph of $G(A)$ interchangeably. Note that the node set $T[x]$ also induces a subgraph in the filled graph $G(F)$.

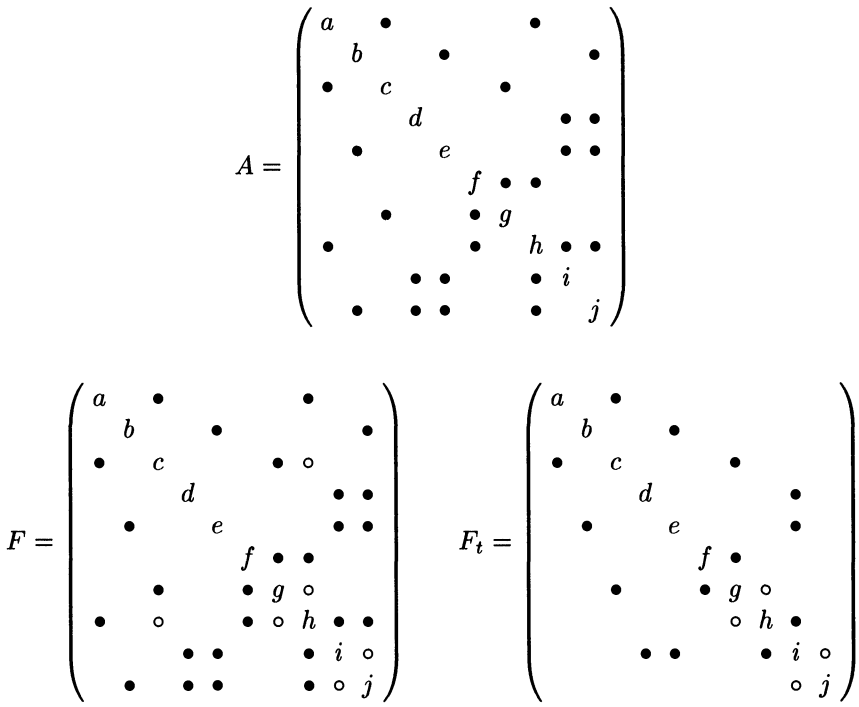


FIG. 2.1. *An example of matrix structures.*

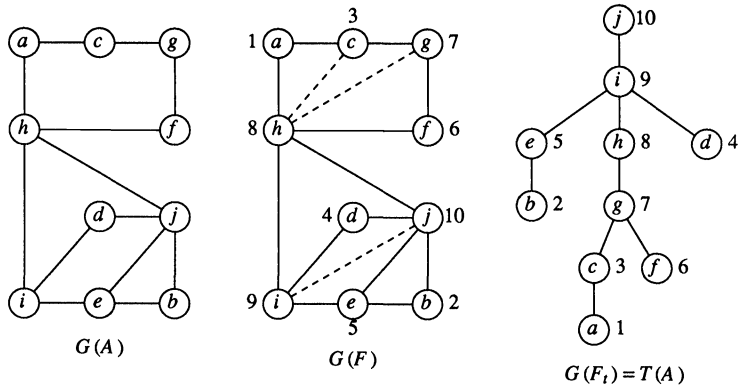


FIG. 2.2. *Graph structures of the example in Fig.* 2.1.

2.3. Motivation from column dependencies. It is interesting to provide an alternative view to introduce the notion of elimination trees. This motivation comes from the dependency in numerical values among columns of the Cholesky factor. We state the following relation without proof.

PROPOSITION 2.2. *For $i > j$, the numerical values of column L_{*i} depend on column L_{*j} if and only if $\ell_{ij} \neq 0$.* □

The immediate consequence of this observation is that the filled graph $G(F)$ captures the column dependencies among the Cholesky columns. To be precise, if we use a directed edge from node x_j to node x_i to indicate that column i depends on column j, a directed filled graph will give the exact relation. Indeed, for each edge in the filled graph, we simply provide a direction that points from the node with a lower subscript to the higher one. The resulting directed graph is actually the graph of the (unsymmetric) Cholesky factor L^T.

If we are interested only in the column dependency relation, we can simplify this directed graph by a process commonly known in graph theory as *transitive reduction* [1]. That is, if there is a directed path of length greater than one from x_j to x_i, and a directed edge from x_j to x_i, the edge from x_j to x_i is deemed redundant and can be removed. The removal of all such redundant edges from the directed filled graph gives its transitive reduction.

The transitive reduction of the directed filled graph generates precisely the elimination tree structure. We can therefore view the elimination tree as providing the minimal amount of information on column dependencies in the Cholesky factor. To better illustrate this connection, we provide in Fig. 2.3 the directed filled graph of the example in Fig. 2.2. We also include the transitive reduction of this directed graph and the final elimination tree structure.

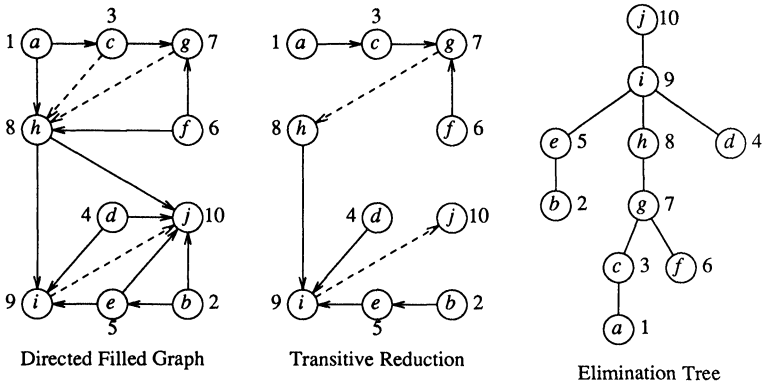


FIG. 2.3. *Directed filled graph and its transitive reduction of the example in Fig.* 2.2.

2.4. Interpretation as a depth-first search tree. As noted earlier, the elimination tree is a spanning tree of the filled graph $G(F)$ of the given matrix A. It can be obtained from the transitive reduction of the directed graph of L^T. In this section, we show that this tree structure can also be obtained from a depth-first search exploration of the undirected filled graph $G(F)$.

Depth-first search is a standard technique of systematically exploring nodes in a graph. It serves as a fundamental tool in devising many efficient graph algorithms (see, for example, [33], [57]). The search starts with an initial node x; and x is marked as *visited.* Then each unvisited node adjacent to x is searched in turn, using depth-first search recursively. The readers are referred to [1] and [58] for details. The edges

that lead to new (unmarked) nodes during a depth-first search of a connected graph G form a rooted tree, called a *depth-first search tree*. In other words, if the edge $\{u, v\}$ leads the search from the marked node u to the unmarked node v, then u is the parent node of v in the depth-first search tree. This means the first node visited is the root of this search tree.

An edge of G that connects a node to one of its proper ancestors, except its parent, in a depth-first search tree is called a *back edge*. (Hopcroft and Tarjan [33], [57] use the term *fronds* to refer to back edges). An edge of G that connects two nodes that are not ancestors of each other is called a *cross edge*. The following important property of a depth-first search tree is well known, and its proof can be found in [57].

THEOREM 2.3. [57]. *For a depth-first search tree of an undirected connected graph G, each edge of G is either a tree edge or a back edge of the tree. (That is, there is no cross edge).* □

We provide a different view of the elimination tree in terms of a depth-first search tree in the next theorem. A direct constructive approach is used in the proof, since it gives better insight into the connection.

THEOREM 2.4. *The elimination tree $T(A)$ of a connected graph $G(A)$ is a depth-first search tree of the filled graph $G(F)$ of A.*

Proof. Let $x_1, x_2, \cdots, x_n$ be the node ordering of the filled graph $G(F)$. Consider the depth-first search of $G(F)$ subject to the following tie-breaking rule: when there is a choice of more than one node to explore next, always pick the one with the largest subscript. The rule implies that the search will start with x_n as the initial node.

It remains to show that the resulting depth-first search tree is the same as the elimination tree $T(A)$. Since both are spanning trees of the filled graph $G(F)$, it is sufficient to show that every tree edge of the depth-first search tree is also an edge in the elimination tree. Consider a tree edge from x_p to x_j in the depth-first search tree; that is, during the search, the edge $\{x_p, x_j\}$ in the filled graph leads to the node x_j. We leave it to the reader to verify that the subscript p is greater than j and is indeed the same as $PARENT[j] = \min \{ i > j \mid \ell_{ij} \neq 0 \}$. □

It follows from Theorem 2.4 that properties of depth-first search trees also apply to elimination trees. In particular, by Theorem 2.3, every edge in the filled graph is either a back edge or a tree edge of the elimination tree. In his thesis [48], Peters establishes this observation and refers to the filled graph as a *palm*. (The notion of a palm tree is introduced by Hopcroft and Tarjan [33] to refer to a directed graph with an underlying rooted spanning tree such that every directed edge in the graph connects a node to its ancestor in the tree.)

To illustrate the result of Theorem 2.4, we use the filled graph example of Fig. 2.2. It is easy to verify that

$$j,\ i,\ h,\ g,\ f,\ c,\ a,\ e,\ b,\ d$$

is the sequence of node visits during a depth-first search subject to the tie-breaking strategy in the proof of Theorem 2.4. For example, after the node j has been visited, we can choose any one of b, d, e, h, or i as the next node. Since $i = x_9$ has the largest subscript, it is selected next in the search. After node i, node $h = x_8$ will be selected out of the three possible candidates: d, e, and h. Note that after node f has been visited, there is no unvisited node adjacent to the node f, so that the search backs up to the node g and finds an unvisited neighbor c of g.

In Fig. 2.4, we display the filled graph $G(F)$ and the resulting depth-first search tree. Back edges are also included in the figure and they are represented by curve

directed lines. Without the back edges, the underlying depth-first search tree is identical to the elimination tree of Fig. 2.2.

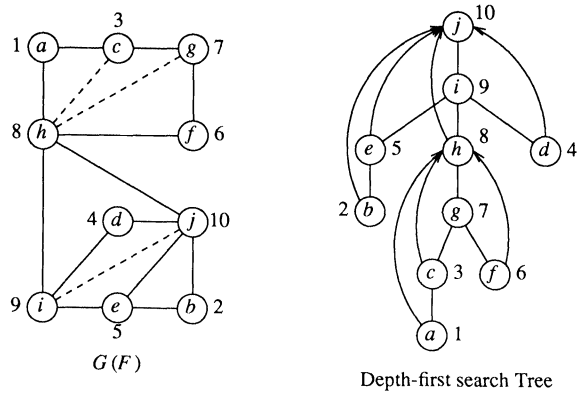


FIG. 2.4. *A depth-first search tree of the filled graph in Fig. 2.2.*

3. Elimination trees and Cholesky factorization.

3.1. Path characterization of filled edges. The elimination tree $T(A)$ is defined in terms of the Cholesky factor L of the matrix A. This tree structure contains a lot of information pertinent to the sparse factorization process. In [55], Schreiber establishes a number of interesting properties of elimination trees that are relevant to the study of sparse Gaussian elimination. The next result is restated here in our terminology without proof. In what follows, unless otherwise stated, we assume that i, j, and k are subscripts and they satisfy $i > j > k$.

THEOREM 3.1. [55]. *If $\ell_{ij} \neq 0$, then the node x_i is an ancestor of x_j in the elimination tree.* □

COROLLARY 3.2. *Let $T[x_i]$ and $T[x_j]$ be two disjoint subtrees of the elimination tree. Then for all $x_s \in T[x_i]$ and $x_t \in T[x_j]$, $\ell_{st} = 0$.*

Proof. Assume for contradiction that there exist nodes $x_s \in T[x_i]$ and $x_t \in T[x_j]$ such that $\ell_{st} \neq 0$. Without loss of generality, let $s < t$. By Theorem 3.1, the node x_t is an ancestor of x_s. This implies that $x_s \in T[x_t] \subseteq T[x_j]$. Therefore, the subtrees $T[x_i]$ and $T[x_j]$ have the node x_s in common and cannot be disjoint. □

Theorem 3.1 provides a necessary condition in terms of the ancestor-descendant relation in the elimination tree for an entry to be nonzero in the filled matrix. Indeed, the result is implicit from the connection between the elimination tree and the depth-first search tree in Theorem 2.4. Every nonzero ℓ_{ij} corresponds to an edge $\{x_i, x_j\}$ in the filled graph so that by Theorem 2.3, it is either a tree edge or a back edge of the elimination tree. Every such edge, therefore, always connects an ancestor-descendant pair in the elimination tree. Corollary 3.2 is simply a restatement of the fact that there is no cross edge in the elimination tree.

We now consider a necessary and sufficient condition for an entry to be nonzero in the filled graph. Rose, Tarjan, and Lueker [53] characterize edges in the filled graph based on the special type of paths in the original graph $G(A)$. We quote the following "path theorem" from them.

THEOREM 3.3. [53]. *Let $i > j$. Then $\ell_{ij} \neq 0$ if and only if there exists a path*

$$x_i,\ x_{p_1}, \cdots,\ x_{p_t},\ x_j$$

in the graph $G(A)$ such that all subscripts in $\{p_1, \cdots, p_t\}$ are less than j. □

Note that the path length in this "path theorem" can be one (that is, $t = 0$). The subscript condition implies that any intermediate nodes along this path must be eliminated before x_j (and x_i). We now extend this path condition in terms of subtrees in the elimination tree.

THEOREM 3.4. *Let $i > j$. Then $\ell_{ij} \neq 0$ if and only if there exists a path*

$$x_i,\ x_{p_1}, \cdots,\ x_{p_t},\ x_j$$

in the graph $G(A)$ such that $\{x_{p_1}, \cdots, x_{p_t}\} \subseteq T[x_j]$.

Proof. If part. Assume that such a path exists. Since x_j is a proper ancestor of every node in $\{x_{p_1}, \cdots, x_{p_t}\}$, by Proposition 2.1, each subscript is less than j. Then by Theorem 3.3, $\ell_{ij} \neq 0$.

Only if part. Assume $\ell_{ij} \neq 0$. By Theorem 3.3, there exists a path

$$x_i = x_{p_0},\ x_{p_1}, \cdots,\ x_{p_t},\ x_{p_{t+1}} = x_j$$

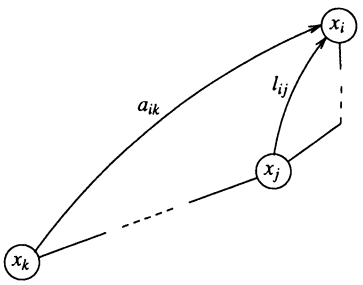
in the graph $G(A)$ such that all subscripts in $\{p_1, \cdots, p_t\}$ are less than j. It remains to show that each node in $\{x_{p_1}, \cdots, x_{p_t}\}$ belongs to $T[x_j]$. There is nothing to prove if $t = 0$. Consider the case when $t > 0$. Assume for contradiction that not every intermediate node belongs to the subtree $T[x_j]$. Let $s > 0$ be the largest subscript such that x_{p_s} does not belong to $T[x_j]$. By the choice of the subscript s, we have $x_{p_{s+1}} \in T[x_j]$.

Since the path is in $G(A)$, this implies that $\{x_{p_s}, x_{p_{s+1}}\}$ is an edge in $G(A)$ and hence also in $G(F)$. But $x_{p_{s+1}}$ cannot be an ancestor of x_{p_s}, for otherwise x_{p_s} would belong to the subtree $T[x_j]$. Therefore, by Theorem 3.1, x_{p_s} must be an ancestor of $x_{p_{s+1}}$. Now that both x_j and x_{p_s} are ancestor nodes of $x_{p_{s+1}}$, and x_j is not an ancestor of x_{p_s}, the node x_{p_s} must be a proper ancestor of x_j. This contradicts the fact that $p_s < j$ and Proposition 2.1. □

3.2. Row structure of the Cholesky factor. Theorems 3.3 and 3.4 characterize edges in the filled graph in terms of paths. In [36], this author extends a property established by Schreiber [55], and the result provides a different necessary and sufficient condition for entries in the Cholesky factor L to be nonzero. We quote this alternative characterization below.

THEOREM 3.5. [36]. *$\ell_{ij} \neq 0$ if and only if the node x_j is an ancestor of some node x_k in the elimination tree, where $a_{ik} \neq 0$.* □

The same result was also given by Tarjan and Yannakakis [59, page 570]. They credited the result to Whitten [60], who presented it at the 1978 SIAM Symposium on Sparse Matrix Computations. Pictorially, the result of Theorem 3.5 can be depicted as follows:



The result of Theorem 3.5 can be used to characterize the row structure of the Cholesky factor. Let us define $T_r[x_i]$ to be the structure of the ith row of the Cholesky

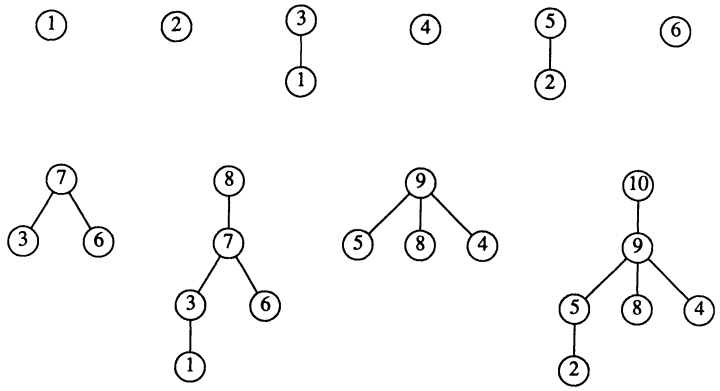


FIG. 3.1. *Row subtrees for the matrix example of Fig.* 2.1.

factor L, that is,

$$T_r[x_i] = \{ x_j \mid \ell_{ij} \neq 0 , j \leq i \} .$$

It follows from Theorem 3.1 that

$$T_r[x_i] \subseteq T[x_i] ,$$

and from Theorem 3.3 that if $a_{ik} \neq 0$, the row structure $T_r[x_i]$ includes all nodes on the path from x_k to x_i in the elimination tree. Schreiber [55] shows that the row structure $T_r[x_i]$ is a pruned subtree rooted at the node x_i in the elimination tree. We shall refer to $T_r[x_i]$ as the ith *row subtree* of L.

Each row subtree $T_r[x_i]$ is completely determined by its set of leaves, since the leaves specify the locations in which the subtree $T[x_i]$ should be pruned. The next corollary characterizes such leaves, and it follows immediately from Theorem 3.5.

COROLLARY 3.6. [36]. *The node x_j is a leaf in the row subtree $T_r[x_i]$ if and only if $a_{ij} \neq 0$, and for every proper descendant x_k of x_j, $a_{ik} = 0$.* □

Note that each leaf x_j in the row subtree $T_r[x_i]$ corresponds to an edge $\{x_i, x_j\}$ in the original graph $G(A)$. In Fig. 3.1, we illustrate the sequence of row subtrees for the example of Fig. 2.1. For convenience, in the figure, each node is labeled by its subscript. Consider the row subtree $T_r[x_{10}]$, which has three leaf nodes $\{x_2, x_4, x_8\}$. Since the node x_7 is the only child of these leaves, the row subtree $T_r[x_{10}]$ can be obtained by pruning the subtree $T[x_7]$ from $T[x_{10}]$.

Therefore, the entire row structure for L is characterized by the elimination tree $T(A)$ and the structure of the original matrix A. The observation that each row structure is a pruned subtree of the elimination tree has a number of interesting applications. It will be used later in this paper to compute the number of nonzeros in L (§3.4), to test if an ordering is perfect elimination (§4.1), to design a compact row storage scheme (§7.2), and to perform symbolic factorization by rows (§8.2).

3.3. Column structure of the Cholesky factor. The previous section characterizes the row structures of the Cholesky factor L in terms of the elimination tree. The result of Theorem 3.1 partially characterizes the column structure of the Cholesky factor. For column j of L, the subscript set of nonzeros in this column is contained in the ancestor set of x_j in the elimination tree. In the next theorem, we provide a complete characterization of the column structures using the elimination tree.

For a graph G and a node v in G, we use $Adj_G(v)$ to denote the set of nodes adjacent to v in the graph. We further extend this Adj operator to subsets. For a subset S of nodes, we define the adjacent set of S in G to be

$$Adj_G(S) = \{ x \notin S \mid x \in Adj_G(v) \text{ for some } v \in S \} .$$

Note that the adjacent subset $Adj_G(S)$ does not include any nodes in S.

THEOREM 3.7. *The structure of column j of the Cholesky factor is given by*

$$Adj_{G(A)}(T[x_j]) \cup \{x_j\} = \{ x_i \mid \ell_{ij} \neq 0 , i \geq j \} .$$

Proof. Consider column j of the Cholesky factor. Since the diagonal entry $\ell_{jj} \neq 0$, the node x_j belongs to the subset. If $\ell_{ij} \neq 0$ with $i > j$, by Theorem 3.5, this is equivalent to the fact that the node x_j is an ancestor of some node x_k with $a_{ik} \neq 0$. This means $x_i \in Adj_{G(A)}(x_k)$, where x_k belongs to the subtree $T[x_j]$. Or simply, this is equivalent to $x_i \in Adj_{G(A)}(T[x_j])$. □

Consider node c in the example of Fig. 2.2. The subtree $T[c]$ contains nodes a and c, so that

$$Adj_{G(A)}(T[c]) = \{ g, h \} ,$$

which gives precisely the locations of off-diagonal nonzeros in the column associated with the node c in the Cholesky factor of A. We now provide some observations based on the result of Theorem 3.7.

COROLLARY 3.8. *$Adj_{G(A)}(T[x_j]) \cup \{x_j\}$ is a clique in the filled graph $G(F)$.* □

COROLLARY 3.9. *For the subset of nodes $T[x_j]$,*

$$Adj_{G(A)}(T[x_j]) = Adj_{G(F)}(T[x_j]) .$$

Proof. The result follows from Theorem 3.7 and the fact that F is a perfect elimination matrix with no additional fill. □

By Corollary 3.9, we can use the notation $Adj(T[x_j])$ to refer to both $Adj_{G(A)}(T[x_j])$ and $Adj_{G(F)}(T[x_j])$. We do so if the underlying graph is clear from the context. For convenience, we also use $Adj(x_j)$ to refer to $Adj_{G(A)}(x_j)$, which is in general a subset of $Adj_{G(F)}(x_j)$. In other words, the adjacent operator Adj, by default, applies to the graph $G(A)$.

3.4. Nonzero counts of the Cholesky factor. The result of Theorem 3.5 can be used to devise efficient algorithms for counting nonzeros in the Cholesky factor matrix L. Let $\eta(L_{i*})$ and $\eta(L_{*j})$ be the number of nonzeros in the ith row and jth column of the factor L, respectively. These quantities can be computed efficiently using the following algorithm. Here, $marker[*]$ is a working integer vector used in the algorithm to mark nodes that have been considered in the current row structure. A similar description of this algorithm is also given by Zmijewski and Gilbert [62]. In [4], Bank and Smith provide an algorithm that essentially computes the nonzero count of L and the structure of the elimination tree simultaneously.

Algorithm 3.1. Nonzero Count.

```
for j := 1 to n do
    η ( L_* j ) := 1 ;

for i := 1 to n do
begin
```

```
η ( L_i * ) := 1 ;
marker [ x_i ] := i ;
 for k < i  and  a_ik ≠ 0  do
 begin  { traverse and mark nodes in the row subtree T_r [x_i ] }
    j := k ;
     while marker [ x_j ] ≠ i  do
     begin
        η ( L_i * ) := η ( L_i * ) + 1 ;
        η ( L_* j ) := η ( L_* j ) + 1 ;
        marker [ x_j ] := i ;
        j := PARENT[j]
     end
 end
end.
```

It is easy to verify that Algorithm 3.1 takes time proportional to the number of nonzeros in the Cholesky factor L and space proportional to the number of nonzeros in the original matrix A. It traverses through nodes in each row subtree $T_r[x_i]$ to obtain the count $\eta(L_{i*}) = \mid T_r[x_i] \mid$.

The column nonzero counts $\{\eta(L_{*j})\}$ are also computed in Algorithm 3.1 during the traversal of the row subtrees. These quantities are useful in setting up columnwise storage schemes. This implies that the storage requirement for values of L can be obtained with only the structure of A and the elimination tree. Moreover, the number of arithmetic operations can also be computed using the column nonzero counts. Indeed, if $\eta(L)$ is the number of nonzeros in the factor L and $\mu(L)$ is the number of multiplicative operations to perform the Cholesky factorization, we have [22]

$$\eta(L) = \sum_{j=1}^{n} \eta(L_{*j}) ,$$

$$\mu(L) = \sum_{j=1}^{n} [\eta(L_{*j}) - 1] [\eta(L_{*j}) + 2] / 2 .$$

Therefore the storage and computational cost for the factor L can be determined without the formation of the actual structure of L. This will be a useful utility routine for any sparse matrix package.

The row nonzero counts $\{\eta(L_{i*})\}$ are also useful even in the context of numerical sparse Cholesky factorization by columns. For $1 \leq i \leq n$, $\eta(L_{i*}) - 1$ represents the number of column modifications required to transform this column of A to that of L (see Proposition 9.1 in §9). Some implementations of parallel sparse factorization make use of this information [19], [62].

3.5. Connectivity consideration. The path characterizations of filled edges in Theorems 3.3 and 3.4 provide some relation between fills and connectivity. In this section, we present a more thorough consideration of subtrees in the elimination tree and connected components for subgraphs in $G(A)$ and $G(F)$. As before, we assume that the graph $G(A)$ is connected (or equivalently, the matrix A is irreducible). Recall the subtree notation $T[x]$ introduced at the end of §2. For a given node x, the subtree $T[x]$ identifies a node subset and hence also defines a subgraph of $G(A)$ and of the filled graph $G(F)$.

THEOREM 3.10. *For each node x_j, the subgraph of $G(A)$ $\{G(F)\}$ consisting of nodes in $T[x_j]$ is connected.*

Proof. We prove by induction on the number of nodes t in $T[x_j]$. The result is obviously true if there is only one node in $T[x_j]$. Assume that the result holds true for all subtrees of size less than t, and $t > 1$. Let $x_{s_1}, \cdots, x_{s_m}$ be the children nodes of x_j. By the inductive assumption, each subgraph consisting of nodes in $T[x_{s_k}]$, for $1 \leq k \leq m$ has fewer than t nodes and is hence connected in $G(A)$. Moreover, for each k, $\{x_j, x_{s_k}\}$ is an edge in the filled graph $G(F)$. By Theorem 3.4, there exists a path from x_{s_k} to x_j through nodes in $T[x_{s_k}]$. This proves the claim that the node subset $T[x_j]$ is a connected subgraph in $G(A)$. Since $G(F)$ is a supergraph of $G(A)$, $T[x_j]$ must also be a connected subgraph in $G(F)$. □

COROLLARY 3.11. *For each node x_j, the set of nodes in $T[x_j]$ forms a connected component in the subgraph of $G(A)$ $\{G(F)\}$ consisting of all nodes except those in $Adj(T[x_j])$.* □

COROLLARY 3.12. *For each node x_j, the set of nodes in $T[x_j]$ forms a connected component in the subgraph of $G(A)$ $\{G(F)\}$ consisting of all nodes except proper ancestors of x_j.*

Proof. Since $Adj(T[x_j])$ is a subset of proper ancestors of the node x_j, the result follows from Corollary 3.11. □

To illustrate these results, consider the example in Fig. 2.2. The subtree $T[g]$ consists of nodes $\{a, c, f, g\}$. This set is a connected subgraph of $G(A)$ (as proved in Theorem 3.10). Furthermore, the adjacent set $Adj(T[g])$ is given by $\{h\}$, whose removal from $G(A)$ gives two connected components, one of which is $\{a, c, f, g\}$. The removal of the set $\{h, i, j\}$ of proper ancestors of g also leaves $T[g]$ as one of the remaining components.

It follows from Corollary 3.11 that if

$$T[x_j] \cup Adj(T[x_j]) \neq X(A),$$

then the subset $Adj(T[x_j])$ separates the nodes of $T[x_j]$ from those of $X(A) - \{T[x_j] \cup Adj(T[x_j])\}$. This observation is used in [45] to devise an effective graph partitioning algorithm by node separators.

4. Elimination trees and chordal graphs.

4.1. Testing chordality. In this section, we shall present some uses of the notion of elimination tree in the study of chordal (or triangulated) graphs. An edge is said to be a *chord* of a cycle if it joins two nonconsecutive nodes on the cycle. An undirected graph is said to be *chordal* if every cycle of length at least four has a chord. It is well known that chordal graphs are exactly those graphs with perfect elimination orderings. Filled graphs are examples of such chordal graphs. A thorough treatment of chordal graphs as a subclass of *perfect graphs* is given by Golumbic [32].

We first consider the use of elimination trees for testing if a given undirected graph is chordal. Since chordal graphs are exactly those with perfect elimination orderings, the chordality of a graph can be established by finding a perfect elimination ordering. Rose, Tarjan, and Lueker [53] and Tarjan and Yannakakis [59] have provided linear algorithms for testing chordality.

One key step in their chordality test algorithms is a linear algorithm that determines if a given ordering is a perfect elimination ordering for a graph $G(A)$. We shall describe this step using the connection of an elimination tree. Let $x_1, x_2, \cdots, x_n$

be the ordering of the graph $G(A)$. For each j, if column j of A does not have any nonzero entry under the diagonal, define $p_j = 0$. Otherwise, let

$$p_j = \min \{ i > j \mid a_{ij} \neq 0 \} .$$

Note that $\{p_j\}$ is defined in the similar way as $PARENT[*]$, except that entries in A are used instead of those in L. Of course, if A does not suffer any fill in its Cholesky factorization, then $PARENT[j] = p_j$.

THEOREM 4.1. *$x_1, x_2, \cdots, x_n$ is a perfect elimination ordering for $G(A)$ if and only if for every row i of A, $a_{i,p_k} \neq 0$ for every nonzero a_{ik} with $k < i$.*

Proof. If part. Assume for contradiction that there exist i and k with $k < i$ such that $a_{ik} \neq 0$ and $a_{i,p_k} = 0$. By the definition of p_k, we have

$$k < p_k \leq i \text{ and } a_{p_k,k} \neq 0 .$$

Therefore, the given ordering cannot be a perfect elimination ordering, since the elimination of the node x_k will create a fill at location ℓ_{i,p_k}.

Only if part. Assume that $x_1, \cdots, x_n$ is a perfect elimination ordering for $G(A)$. Let $T(A)$ be the elimination tree of A. This implies that $PARENT[i] = p_i$ for $i = 1, \cdots, n$. Consider any nonzero a_{ik} with $k < i$. The node x_k belongs to the row subtree $T_r[x_i]$ rooted at x_i. Therefore, the parent node $x_{PARENT[k]} = x_{p_k}$ of x_k also belongs to this row subtree. It follows from Theorem 3.5 that ℓ_{i,p_k} is nonzero. Since the ordering is a perfect elimination ordering for $G(A)$, a_{i,p_k} must also be nonzero. □

Theorem 4.1 contains a simple test to see if an ordering is a perfect elimination ordering for a given graph $G(A)$. In essence, it is comparing to see if the ith row structure of A is the same as the ith row structure of L (which is the row subtree $T_r[x_i]$), for $i = 1, \cdots, n$. The following algorithm is from Tarjan and Yannakakis [59].

Algorithm 4.1. Test for Perfect Elimination Ordering.

```
for i := 1 to n do
begin
   marker [ x_i ] := i ;
   for k < i and a_ik ≠ 0 do
      marker [ x_k ] := i ;
   for k < i and a_ik ≠ 0 do
      if marker[x_{p_k} ] ≠ i then return false ;
   p_i := max { 0, min { s > i | a_si ≠ 0 } }
end;
return true.
```

It is easy to see that Algorithm 4.1 determines if an ordering is perfect elimination for a graph $G(A)$ in time proportional to the number of nodes and number of edges in $G(A)$. The chordality-testing algorithms of Rose, Tarjan, and Lueker [53] and Tarjan and Yannakakis [59] make use of Algorithm 4.1. Both consist of two steps:

Step 1. Find an ordering P for the graph $G(A)$ such that P is a perfect elimination ordering for A if and only if $G(A)$ is chordal.

Step 2. Apply Algorithm 4.1 to see if the ordering P is a perfect elimination ordering for $G(PAP^T)$.

It is clear from definitions that a *minimal fill ordering*[1] for A is a perfect elimination ordering if and only if $G(A)$ is chordal. Therefore, any minimal fill ordering will serve the purpose for Step 1. The *lexicographic ordering*, a special form of breadth-first search, by Rose, Tarjan, and Lueker [53] can be used to determine a minimal fill ordering in linear time. Tarjan and Yannakakis [59] provide a simpler linear algorithm based on what they call *maximum cardinality search*, to determine an ordering satisfying the requirement in Step 1. Combining Algorithm 4.1 with either the lexicographic ordering or the maximum cardinality search, we have therefore an overall linear algorithm for testing the chordality of undirected graphs. It is interesting to note the implicit role played by the elimination tree in these chordality-testing algorithms.

4.2. Intersection graph representations of chordal graphs. It is known that a graph is chordal if and only if it is the *intersection graph* of a family of subtrees of a tree [17], [32]. In this section, we show how to construct such a tree representation explicitly using the elimination tree of a chordal graph. The approach of using elimination trees in arriving at this tree representation is new.

For completeness, we first define intersection graphs. Let Γ be a family of nonempty sets. The *intersection graph* of Γ is obtained by representing each set in Γ by a node and connecting two nodes by an edge if and only if their corresponding sets intersect (see, for example, Golumbic [32]).

Let A be a symmetric matrix with graph structure $G(A)$ and filled graph $G(F)$. We assume that $x_1, x_2, \cdots, x_n$ is the node elimination sequence. Let $T(A)$ be the corresponding elimination tree, with $T_r[x_1], \cdots, T_r[x_n]$ as the sequence of row subtrees.

LEMMA 4.2. *$\ell_{ij} \neq 0$ if and only if $T_r[x_i] \cap T_r[x_j] \neq \emptyset$.*

Proof. For definiteness, assume $i > j$. It follows from definition that $\ell_{ij} \neq 0$ if and only if $x_j \in T_r[x_i]$. Therefore, if $\ell_{ij} \neq 0$, then $x_j \in T_r[x_i] \cap T_r[x_j]$ and hence $T_r[x_i] \cap T_r[x_j] \neq \emptyset$.

On the other hand, if the intersection of the two row subtrees is nonempty, say $x_s \in T_r[x_i] \cap T_r[x_j]$, this implies that the node x_j lies on the path from x_s to x_i in the elimination tree. Hence $x_j \in T_r[x_i]$ and that implies that $\ell_{ij} \neq 0$. □

THEOREM 4.3. *The chordal graph $G(F)$ is the intersection graph of the row subtrees in the elimination tree $T(A)$.*

Proof. The result follows directly from Lemma 4.2. □

The result in Theorem 4.3 gives a constructive approach to determine the intersection graph representation of any chordal graph. Indeed, for a given chordal graph G, we can first find a perfect elimination sequence $x_1, \cdots, x_n$ for G. Obtain the elimination tree of this node sequence and the associated row subtrees $T_r[x_1], \cdots, T_r[x_n]$. Then the chordal graph G is given by the intersection graph of these row subtrees in the elimination tree.

4.3. Separators for chordal graphs. A *separator* of a connected graph is a subset of nodes whose removal renders the remaining subgraph disconnected. In [30], Gilbert, Rose, and Edenbrandt have devised an efficient algorithm to determine $O(\sqrt{m})$-separators for chordal graphs, where m is the number of edges in the graph. Their algorithm uses implicitly the structure of an elimination tree. In this section, we explore this connection. Let A be a sparse matrix with $G(A)$ as a chordal graph and assume that it is ordered with no fill. Let $T(A)$ be its corresponding elimination tree.

[1] A minimal fill ordering P on the matrix A is one such that no ordering of A will generate a filled graph that is a proper subgraph of the filled graph of PAP^T [52].

The separator algorithm by Gilbert, Rose, and Edenbrandt [30] finds a set of size at most $O(\sqrt{m})$, whose removal divides the graph into connected components, each with no more than $n/2$ nodes. Their approach can be interpreted as first obtaining from the elimination tree $T(A)$ the smallest subscript j such that $T[x_j]$ has more than $n/2$ nodes. Then, the desired $O(\sqrt{m})$-separator is given by the set $S = Adj(T[x_j]) \cup \{x_j\}$.

It follows from Corollary 3.8 that this set S forms a clique in the chordal graph. Being a clique, the set must have size no greater than $O(\sqrt{m})$. Furthermore, by the construction of S, the removal of S leaves no component with more than $n/2$ nodes. Our discussion here presents a different view of the separator result in the paper [30] and interprets the algorithm in terms of the elimination tree.

5. Determination of elimination trees.

5.1. Basis of the algorithm. For an n-by-n symmetric matrix A, we consider the problem of determining the structure of its elimination tree $T(A)$, that is, computing the parent vector $PARENT[i]$, $i = 1, \cdots, n$, of $T(A)$. We can obtain the parent vector using the following code.

```
for i := 1 to n do
begin
   PARENT[i] := 0 ;
    for k < i and ℓ_ik ≠ 0 do
        if PARENT[k] = 0 then PARENT[k] := i ;
end.
```

Note that the inner **for**-loop uses the row structures of the Cholesky factor L. For this purpose, we can generate the location of each row nonzero of L as needed, without storing the entire structure of L. The row structure characterization of the factor matrix in §3.2 is applicable here. Indeed, for each row i, we can generate the structure of row i of L using the structure of the original matrix A and the current values of $PARENT[k]$, $k = 1, \cdots, i-1$. That will result in an algorithm for computing the parent vector in time proportional to the number of nonzeros in L and in space proportional to the number of nonzeros in A.

5.2. The algorithm using set union operations. In [36], Liu further extends the row approach of the last section to obtain more efficient ways of finding the elimination tree structure. The key observation is that the problem of computing the $PARENT[*]$ vector of $T(A)$ from the graph $G(A)$ can be expressed in terms of basic set operations for the set union problem. Following Tarjan [58, Chap. 2], we consider the three set operations:

makeset(x): create a new singleton set with element x;
find(x): return the representative of the set containing x;
link(x, y): form the union of the two sets containing x and y, and return the new representative of the union set.

Implicitly assumed here is that each set is represented by a rooted tree, where each tree node corresponds to a member of the set, and the root is the representative.

We now describe the determination of the elimination tree from the graph $G(A)$. The following algorithm finds the $PARENT[*]$ vector for the elimination tree $T(A)$ using the above three basic set operations. A temporary vector $realroot[*]$ is used to store the actual root of the subtree under consideration in the partially formed elimination tree.

Algorithm 5.1. Elimination Tree.

```
for i := 1 to n do
begin
   makeset( i ) ;
   realroot[i] := i ;
   PARENT[i] := 0 ;
   vroot := i ;
   for k < i and a_ik ≠ 0 do
   begin
      u := find(k);
      t := realroot[u];
      if PARENT[t] = 0 and t ≠ i then
      begin
         PARENT[t] := i ;
         vroot := link(vroot, u) ;
         realroot[vroot] := i
      end
   end
end.
```

Within this framework, the various enhancement techniques as described by Tarjan in [58] can be applied. In particular, the notion of *path compression* to change the structure of the tree during a "*find*" by moving nodes closer to the root can be used to reduce the amount of time for later "*find*"s. In that case, two structures are maintained throughout the course of the algorithm, one for the actual elimination tree and the other for the compressed tree structure. Moreover, the use of *union by ranks* in the "*link*" operation keeps the resulting tree shallow by choosing a more appropriate representative for the union set. It has the desirable effect of balancing the tree.

In [36], Liu provides a detailed description of the algorithm to determine the elimination tree structure using path compression. He also notes that path compression with balancing will produce a theoretically more efficient scheme. An explicit description of this scheme then appears in the work of Zmijewski and Gilbert [62]. We refer the reader to these two papers.

A direct use of the analysis by Tarjan [58] provides the following complexity bounds. Let m be the number of nonzeros in the matrix A.

THEOREM 5.1. *Algorithm 5.1, using only path compression, determines the elimination tree in* $O(m \log_2 n)$ *time.* □

THEOREM 5.2. *Algorithm 5.1, using path compression with balancing, determines the elimination tree in* $O(m\alpha(m,n))$ *time, where* $\alpha(*,*)$ *is a functional inverse of Ackerman's function.* □

In [16], Fischer defines a class of trees in order to establish a lower bound for the set union algorithm without balancing. It is interesting to note that we can use the same example to construct sparse matrix structures so that Algorithm 5.1 with only path compression will require time proportional to $m \log_2 n$. In other words, the time bound in Theorem 5.1 is the best possible.

It is appropriate here to mention two parallel algorithms to determine the elimination tree structure. Zmijewski and Gilbert [62] provide a distributed version of Algorithm 5.1 using both path compression and balancing. Gilbert and Hafsteinsson

[29] use the divide-and-conquer approach to devise an efficient parallel algorithm to determine the tree structure.

5.3. Experimental results on balancing. The time bound in Theorem 5.2 involves the inverse Ackerman function α, which is an extremely slow growing function. For all practical purposes, it can be regarded as a constant value of 5. (For more details on this function, the readers are referred to [58].) From the theoretical complexity bound, Algorithm 5.1 using path compression with balancing is superior to the one without balancing.

Both schemes (with and without balancing) were implemented to compute the structure of elimination trees of large sparse matrices. We applied them to the model square grid problem (with 9-point difference operator), where the nodes were ordered by the minimum degree ordering algorithm. The time in CPU seconds on a SUN 3/50 is tabulated in Table 5.1.

TABLE 5.1
Time to determine elimination trees.

Grid	Algorithm 5.1 using path compression	
	without balancing	with balancing
30×30	0.10	0.18
50×50	0.26	0.50
80×80	0.82	1.30
100×100	1.08	2.08
180×180	3.82	6.66

We note from Table 5.1 that the scheme without balancing runs almost twice as fast as the one with balancing, even when the grid size is 180 ($n = 32,400$). Runs on practical sparse matrix examples from the Harwell–Boeing collection give similar results. Therefore, in practice, we recommend the simpler code of using Algorithm 5.1 without balancing.

6. Elimination trees and matrix reordering.

6.1. Topological orderings. In this section, we consider *equivalent* reorderings based on the structure of the elimination tree. Let A be a given symmetric matrix. Following [43], we call two orderings P and Q equivalent if the structures of the filled graphs of PAP^T and QAQ^T are the same (that is, the filled graphs are isomorphic). For convenience, we shall call P an equivalent reordering of the matrix A if the filled graph of A has the same structure as that of PAP^T. It is known that equivalent orderings require the same amount of arithmetic for the sparse Cholesky decomposition of their permuted matrices (see, for example, [11], [52]). Therefore, in terms of both storage and computational costs, equivalent reorderings are as good as the original ordering. However, we may use them to take advantage of other aspects of elimination, some of which will be illustrated in later sections.

The structure of an elimination tree provides some flexibility and freedom in reordering the nodes. We first consider the class of topological orderings. As defined in §2, a topological ordering on a rooted tree is one that numbers the children nodes before their parent node. Given an initial node ordering on the matrix A and its corresponding elimination tree $T(A)$, let P be a permutation matrix for A that corresponds to a topological ordering of the nodes in $T(A)$.

THEOREM 6.1. *As unlabeled graphs, the filled graph of* $G(PAP^T)$ *and the filled graph of* $G(A)$ *are isomorphic.*

Proof. Let F be the filled matrix of A. Put $\tilde{A} = PAP^T$ and let $\tilde{F}$ be the corresponding filled matrix of $\tilde{A}$. We want to show that $G(F)$ and $G(\tilde{F})$ are structurally identical.

Let $x_1, x_2, \cdots, x_n$ be the node elimination sequence for the matrix A and let $\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n$ be that for the matrix $\tilde{A}$. Consider two nodes x_i and x_j in A. Let $x_i = \tilde{x}_{\tilde{r}}$ and $x_j = \tilde{x}_{\tilde{s}}$ in the sequence for $\tilde{A}$. It is sufficient to show that $\{x_i, x_j\}$ is an edge in the filled graph $G(F)$ if and only if $\{\tilde{x}_{\tilde{r}}, \tilde{x}_{\tilde{s}}\}$ is an edge in $G(\tilde{F})$.

Assume that $\{x_i, x_j\}$ is an edge in $G(F)$ and $i > j$. By Theorem 3.1, x_i is a proper ancestor of x_j in the elimination tree $T(A)$. Since P is a topological ordering, the node $\tilde{x}_{\tilde{r}}$ is labeled after $\tilde{x}_{\tilde{s}}$ so that $\tilde{r} > \tilde{s}$. Furthermore, by Theorem 3.4, there exists a path in the graph $G(A)$

$$\tilde{x}_{\tilde{r}} = x_i, x_{p_1}, \cdots, x_{p_t}, x_j = \tilde{x}_{\tilde{s}}$$

such that $\{x_{p_1}, \cdots, x_{p_t}\} \subseteq T[x_j]$. By the property of the topological ordering of P, these nodes $x_{p_1}, \cdots, x_{p_t}$ are labeled before $\tilde{x}_{\tilde{s}}$ in the matrix $\tilde{A}$. Therefore, by Theorem 3.3, $\{\tilde{x}_{\tilde{r}}, \tilde{x}_{\tilde{s}}\}$ is also an edge in the filled graph $G(\tilde{F})$.

Conversely, let $\{\tilde{x}_{\tilde{r}}, \tilde{x}_{\tilde{s}}\}$ be an edge in $G(\tilde{F})$. For definiteness, let $\tilde{r} > \tilde{s}$. Note first that x_i does not belong to the subtree $T[x_j]$, for otherwise, it contradicts the topological ordering property of P. By Theorem 3.3, there exists a path in $G(A)$

$$\tilde{x}_{\tilde{r}}, \tilde{x}_{\tilde{p}_1}, \cdots, \tilde{x}_{\tilde{p}_t}, \tilde{x}_{\tilde{s}}$$

such that all subscripts $\tilde{p}_1, \cdots, \tilde{p}_t$ are less than $\tilde{s}$. By the property of the topological ordering of P, the nodes in $\{\tilde{x}_{\tilde{p}_1}, \cdots, \tilde{x}_{\tilde{p}_t}\}$ cannot be ancestors of x_j. From the above path, we know that all the nodes on the path belong to the connected component containing the node x_j in the subgraph of $G(A)$ excluding the set of proper ancestors of x_j in the tree. By Corollary 3.12, these nodes all belong to the subtree $T[x_j]$. In other words, we have a path in $G(A)$ from $\tilde{x}_{\tilde{s}} = x_j$ to $\tilde{x}_{\tilde{r}} = x_i$ through nodes in the subtree $T[x_j]$ and x_i is outside of $T[x_j]$. Again by Corollary 3.12, this means that x_i is a proper ancestor of x_j so that $i > j$. Therefore, using Theorem 3.4, $\{x_i, x_j\}$ is also an edge in the filled graph $G(F)$. □

COROLLARY 6.2. *As unlabeled trees, the elimination tree* $T(PAP^T)$ *and the elimination tree* $T(A)$ *are isomorphic.* □

The result of Theorem 6.1 implies that every topological ordering of the elimination tree is an equivalent reordering of the given sparse matrix. The amount of fills and the number of arithmetic operations for the factorization will be preserved by every topological ordering of the elimination tree. Indeed, by Corollary 6.2, even the structure of the elimination tree is preserved.

For a given elimination tree, there are many possible topological orderings. One important example is the class of *postorderings* [1]. In a postordering, the nodes within every subtree of the elimination tree will be numbered consecutively. The root of a subtree will always be labeled last among nodes in the subtree. For the elimination tree of Fig. 2.2, the ordering P

$$d,\ f,\ a,\ c,\ g,\ h,\ b,\ e,\ i,\ j$$

is a postordering on the tree $T(A)$. In Fig. 6.1, we display the structure of the filled matrix $\tilde{F}$ of PAP^T, the graph $G(\tilde{F})$ and the tree $T(PAP^T)$.

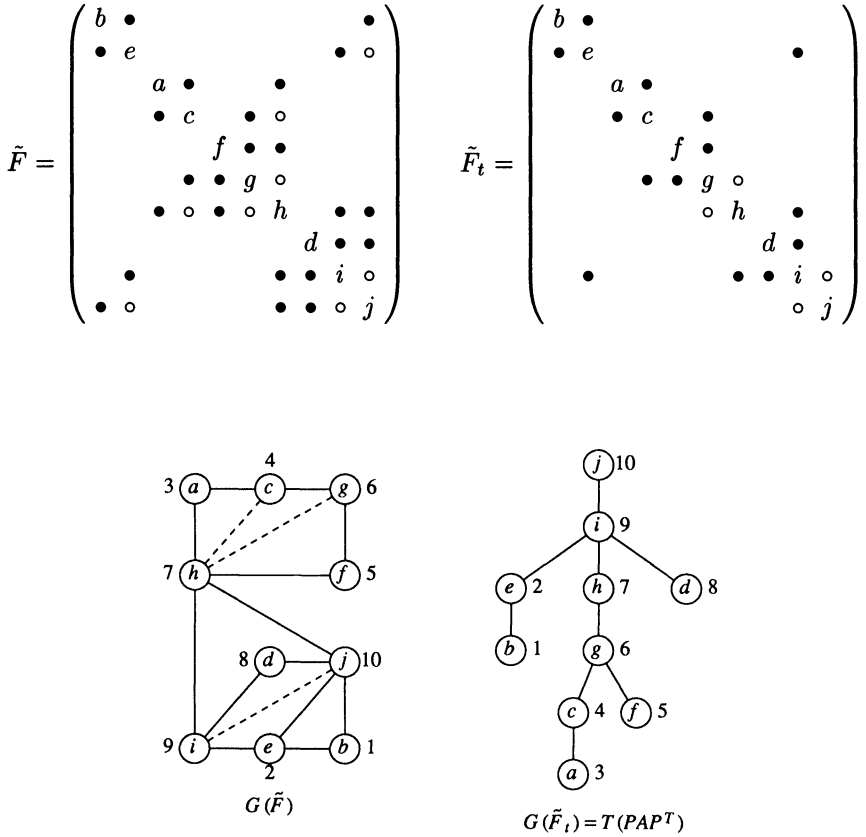


FIG. 6.1. *A postordering of the elimination tree in Fig. 2.2.*

Topological orderings on an elimination tree are notions similar to *consistent* orderings introduced by Peters [48]. He also provides a proof of a result equivalent to Theorem 6.1. It leads him to observe that if $A = LL^T$ and P is a topological ordering of $T(A)$, then PLP^T is lower triangular and is the Cholesky factor of PAP^T. Furthermore, his concept of *preserving palm* can also be expressed in terms of topological orderings on the elimination tree.

6.2. Root selection of elimination trees by reorderings. In the previous section, for a given sparse matrix A, we have shown that every topological ordering of its elimination tree $T(A)$ is an equivalent reordering for A. Furthermore, the structure of the elimination tree is preserved by such topological orderings. However, in general, equivalent reorderings for A may not retain the elimination tree structure.

We now show that for any node x in $G(A)$, there is an (essentially) equivalent reordering such that the resulting elimination tree is rooted at x. Let F be the filled matrix of A. We need the following result from Rose [52].

THEOREM 6.3. [52]. *For any node x in $G(F)$, there is a perfect elimination ordering on $G(F)$ such that the node x is numbered last.* □

COROLLARY 6.4. *For any node x in $G(A)$, there is an ordering P on $G(A)$ such that the node x is numbered last, and such that the filled graph of $G(PAP^T)$ is a subgraph of the filled graph of $G(A)$.*

Proof. Let $G(F)$ be the filled graph of $G(A)$. By Theorem 6.3, there is a perfect elimination ordering P on $G(F)$ such that the node x is numbered last. In other words, the factorization of the permuted filled matrix PFP^T does not create additional fill. This implies that fills created in the factorization of PAP^T will be accounted for by $G(PFP^T)$. Therefore the filled graph of $G(PAP^T)$ is a subgraph of $G(PFP^T) = G(F)$. □

In Corollary 6.4, the ordering P is at least as good as the initial ordering for A in terms of fill. In the case when the matrix A has been ordered initially by a minimal fill ordering, then the filled graph of $G(PAP^T)$ will be the same as the filled graph of $G(A)$. Otherwise, the ordering P may be a better ordering so that the filled graph of $G(PAP^T)$ is a proper subgraph of the filled graph of $G(A)$. The following theorem is a direct consequence of Corollary 6.4.

THEOREM 6.5. *For any node x in $G(A)$, there is an ordering P on $G(A)$ such that the node x is the root of the elimination tree $T(PAP^T)$, and that the filled graph of $G(PAP^T)$ is a subgraph of the filled graph of $G(A)$.* □

For a better understanding of the result, it is instructive to consider the graph example of Fig. 2.2 (or Fig. 6.1). Say, we want a reordering on $G(A)$ such that node b is the root of the elimination tree. One such reordering is

$$a,\ c,\ f,\ g,\ h,\ d,\ i,\ e,\ j,\ b\,.$$

For this reordering, we display the new labeling and its corresponding elimination tree in Fig. 6.2. Note that the structure of the resulting elimination tree is quite different from the original one in Fig. 2.2. For this example, the filled graph structure is preserved, since the original ordering is a minimal fill ordering (in fact, a minimum fill ordering).

6.3. Tree restructuring by elimination tree rotations. Reordering so that a given node x becomes the root of the resulting elimination tree is a special case of a more general technique, called *elimination tree rotation*, introduced by the author [43]. The essence of elimination tree rotation can be described as follows. For any given

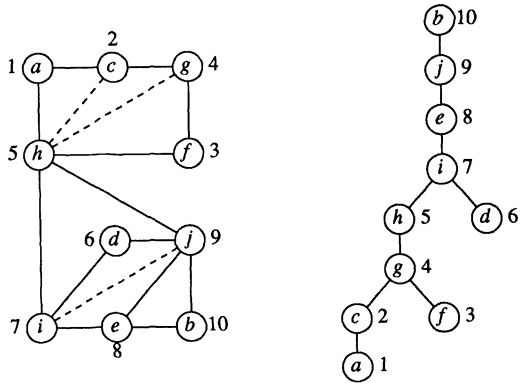


FIG. 6.2. *An equivalent reordering of the graph in Fig.* 2.2.

node x in the elimination tree, the ordering obtained from an elimination tree rotation (actually a composite of basic rotations) at x will maintain the relative ordering of the nodes that are not ancestors of x. But the ancestors nodes of x will be reordered such that the nodes in $Adj(T[x]) \cup \{x\}$ are numbered last.

The overall effect of such a reordering will retain the structure of subtrees outside the ancestor set of x. Moreover, the nodes in $Adj(T[x]) \cup \{x\}$ will be "promoted" up, one of which will form the root of the resulting elimination tree. Since this set is a clique in the filled graph (by Corollary 3.8), they can be renumbered among themselves in any order so long as they are labeled last. Picking x to be the last node will give a tree rooted at x.

The matrix reordering example of Fig. 6.2 is obtained by an elimination tree rotation of the tree in Fig. 6.1 at the node b. Therefore, the nodes in the clique of $G(F)$

$$Adj(T[b]) \cup \{b\} = \{ b, e, j \}$$

are labeled last in the reordering. Of these three nodes, b is selected as the last so that the resulting elimination tree will be rooted at b. The relative ordering of the remaining seven nodes is maintained in the new reordering.

Orderings from elimination tree rotations form an important class of equivalent reorderings in restructuring the tree. The criterion on the tree structure for the best equivalent reordering depends on the application. For example, having a balanced (or unbalanced) tree structure may be a desirable criterion. Another one would be to reduce the height of the elimination tree. Some applications will be discussed in later sections. For more details on elimination tree rotations, the reader is referred to [43] and [46].

7. Elimination trees in sparse storage schemes.

7.1. Relative row-index scheme for Cholesky factors. The most commonly used storage scheme for sparse Cholesky factors is the one proposed by Sherman [56]. His compressed column storage scheme stores the nonzeros in the Cholesky factor column by column. The subscript information is provided in an auxiliary vector in a compressed form by taking advantage of columns whose initial row subscripts are final subsequences of those in the previous column. The scheme maintains *absolute* row index information for the column structures. The relevance of this storage scheme in sparse factorization is also addressed in [15] and [54].

In [55], Schreiber suggests the use of *relative* row indices to store the structural information for columns of the Cholesky factor. The possible advantages, as pointed out by Schreiber, are: improved efficiency in performing column updates, better utilization of vectorized machines, reduced overhead storage requirement (by packing more relative row indices into a word). One important use of relative indices is in some implementations of the multifrontal method [2].

We now provide the basis for his scheme using the terminology established in this paper. As before, let A be a given sparse symmetric positive-definite matrix initially ordered by some fill-reducing ordering. Let $T(A)$ be the corresponding elimination tree.

THEOREM 7.1. *Let x_p be the parent of the node x_j in the elimination tree. Then*

$$Adj(T[x_j]) \subseteq Adj(T[x_p]) \cup \{x_p\} .$$

Proof. The result follows from the fact that $T[x_j] \subseteq T[x_p]$. □

By Theorem 3.7, the set $Adj(T[x_j]) \cup \{x_j\}$ provides the structure of column j of the Cholesky factor. Theorem 7.1 implies that the structure of column j without the diagonal element is contained in the column structure of p. (Note that in matrix terms, p is the row subscript of the first off-diagonal nonzero in column j of the factor L.) The relative row-index scheme of Schreiber makes use of this observation. For an off-diagonal nonzero ℓ_{ij} in column j, it follows from Theorem 7.1 that ℓ_{ip} is also nonzero. Instead of storing the absolute row index i for the nonzero ℓ_{ij}, Schreiber suggests storing the relative location of the corresponding entry ℓ_{ip} in the structure of column p.

Let us use the matrix example in Fig. 2.1 to illustrate Schreiber's scheme. Using the given ordering, we consider column $j = 2$ or node b. The parent of b in the elimination tree is the node e so that $p = 5$. The structure of column 2 is given by

$$2, 5, 10,$$

while that of column 5 is

$$5, 9, 10.$$

These absolute row subscripts will be stored in a compressed form by Sherman's storage scheme. However, in Schreiber's relative row-index scheme, the structure of column 2 is maintained implicitly by the following sequence of relative indices:

$$-, 1, 3,$$

since the absolute indices of 5 and 10 appear in locations 1 and 3 of column $p = 5$. It is important to realize that the structure of the parent column p will also be stored in the relative row-index form.

Schreiber [55] provides a complete set of numerical Cholesky factorization and forward/backward substitution algorithms using this relative row-index storage scheme. He employs the fact that all column modifications of column j in the Cholesky factorization are from columns $k < j$, where $x_k \in T_r[x_j]$, the row subtree of x_j (see Proposition 9.1 in §9). Moreover, the actual numerical values from such a column k can be passed onto column j through the columns associated with nodes on the path from x_k to x_j in the elimination tree. Therefore, the relative row indices are sufficient to accumulate contributions to column j. The reader is referred to [55] for details.

7.2. Compact row scheme for Cholesky factors. The compressed column storage scheme by Sherman is especially suited for the column-Cholesky factorization algorithm. Moreover, the compressed subscript approach is demonstrated to be very

effective in reducing the amount of overhead storage for the structure of the Cholesky factor. The amount of subscript overhead used is problem dependent.

In [36], Liu proposes an alternative row storage scheme, and the integer overhead for subscripts is shown to be no more than the number of nonzeros in the original matrix A. The scheme is based on the result on the row structure of the Cholesky factor established in §3.2. Since the structure of row i of the factor is given by the ith row subtree (see, for example, Fig. 3.1), it is sufficient to store the *leaves* of the row subtree. But each such leaf corresponds to one nonzero entry in the original matrix (by Theorem 3.5). This implies that for row i, its row subtree structure can be represented by no more than the number of nonzeros in row i of the matrix A.

In this row storage scheme, whenever the structure of row i is needed, its entire set of subscripts will be generated from the row subtree. This is accomplished by a *postorder* tree traversal of the subtree. But in order to obtain the set of (column) subscripts for the row in ascending sequence, we perform an equivalent reordering of the matrix using a postordering (as described in §6.1). Subsequent postorder traversal of any subtree will always produce subscripts in ascending order. It is instructive to note that this observation is not applicable to the row subtrees in Fig. 3.1, since the ordering used is not a postordering of the elimination tree (see Fig. 2.2). However, it works for the reordering used in Fig. 6.1. For more details of the row storage scheme, the reader is referred to the paper [36].

Bank and Smith [4] propose a row scheme, which is quite similar to the one described in [36]. The techniques of postorder reordering and elimination tree computation are also applicable to their scheme.

7.3. Data structures for sparse QR and LU factors. In [27], George and Ng describe a novel implementation of sparse Gaussian elimination with partial pivoting. For a given sparse matrix M (not necessarily symmetric), their scheme determines from the structure of M a *static* data structure that will accommodate nonzeros in the factor matrices for all possible partial pivoting sequences. Their static data structure consists of two parts: one for the lower triangular factor and the other for the upper.

Define the symmetric matrix $A = M^T M$. The static structure for the upper triangular factor of M is given by the Cholesky factor of the symmetric matrix A. Indeed, for any permutation P, if the row-permuted matrix PM is decomposed by Gaussian elimination into LU, George and Ng show that the structure of U is contained in that of the Cholesky factor of $A = M^T M$.

As for the lower triangular part, it turns out that its structure can be determined in terms of the elimination tree $T(A)$ of the matrix product A. It is shown in [26] that each row structure of the lower triangular factor is contained in the structure given by a chain of tree edges in the elimination tree $T(A)$, that is, a sequence of nodes with subscripts

$$i_1,\ i_2 = PARENT[i_1]\,, \cdots,\ i_t = PARENT[i_{t-1}]\,.$$

This not only saves overhead storage in implementing the static data structure, but also reduces the factorization time. It is beyond the scope of this paper to consider the details. The reader is referred to [26]. It should be pointed out that the same static data structure is also useful in the sparse orthogonal decomposition of the matrix M.

8. Elimination trees in symbolic factorization.

8.1. Symbolic factorization of A by columns. Efficient algorithms have been devised to perform the symbolic factorization of a given large sparse symmetric matrix

A. In the literature, symbolic factorization refers to the determination of the column structures of the Cholesky factor L of A. Rose, Tarjan, and Lueker [53] provide an algorithm (called "FILL_IN") for this purpose. In [21], George and Liu describe a similar symbolic factorization scheme, the output of which is tailored for the compressed column storage scheme of Sherman.

In this section, we relate the basic algorithm for symbolic factorization in [21] and [53] to the structure of an elimination tree. Indeed, both approaches are based on the following result, which can be regarded as an extension of Theorem 7.1.

THEOREM 8.1. *For any node x_j in the elimination tree, $Adj(T[x_j])$ is given by*

$$(Adj(x_j) - \{x_1, \cdots, x_{j-1}\}) \cup \{ Adj(T[x_s]) - \{x_j\} \mid x_s \text{ is a child of } x_j \} .$$

Proof. By definition of $T[x_j]$, we have

$$Adj(x_j) - \{ x_1, \cdots, x_{j-1} \} \subseteq Adj(T[x_j]) .$$

By Theorem 7.1, for each child node x_s of x_j,

$$Adj(T[x_s]) - \{x_j\} \subseteq Adj(T[x_j]) .$$

On combining the two, we therefore have shown that the given set is contained in $Adj(T[x_j])$.

On the other hand, consider any $x_i \in Adj(T[x_j])$. From Proposition 2.1, note that x_i cannot be one of $\{x_1, \cdots, x_j\}$. By Theorem 3.7, $\{x_i, x_j\}$ is an edge in the filled graph of A. By Theorem 3.5, there exists a node x_k such that the node x_j is an ancestor of x_k and $\{x_k, x_i\}$ is an edge in $G(A)$. If $x_k = x_j$, then clearly $x_i \in Adj(x_j)$. Otherwise, let x_s be the child node of x_j, which is also an ancestor of x_k. Then by Theorem 3.5, $\{x_i, x_s\}$ is also an edge in the filled graph. Therefore, by Theorem 3.7, $x_i \in Adj(T[x_s])$. In other words, in both cases, x_i must also belong to the given set. This completes the proof. □

Theorem 8.1 provides the basis for the formulation of an efficient symbolic factorization algorithm to compute the column structure $\{Adj(T[x_j])\}$. Such formulation requires the structure of the elimination tree $T(A)$. However, since the parent of the node x_j $(j < n)$ in the elimination tree can be obtained readily from $Adj(T[x_j])$, a predetermination of the elimination tree structure is not necessary. We can therefore formulate the following symbolic factorization algorithm.

Algorithm 8.1. Symbolic Factorization by Columns.

```
for j := 1 to n do
begin
   Adj(T[x_j]) := Adj(x_j) - { x_1 , ··· , x_{j-1} } ;
   for x_s in the children list of x_j do
      Adj(T[x_j]) := Adj(T[x_j]) ∪ Adj(T[x_s]) - {x_j} ;
   if Adj(T[x_j]) ≠ ∅ then
   begin
      p := min { i | x_i ∈ Adj(T[x_j]) } ;
      add x_j to the children list of x_p
   end
end.
```

Algorithm 8.1 uses implicitly the fact that $x_1, \cdots, x_n$ is a topological ordering of the elimination tree $T(A)$. Indeed, when the structure of column j is determined,

we assume that the column structures of its children nodes have all been computed. This is always the case if we compute the column structures in increasing order of the subscript j (from Proposition 2.1).

In terms of the actual implementation of the children lists, we need to use only one extra integer n-vector $CLIST[*]$. It can be arranged so that at the beginning of step j, the list of children nodes of the node x_j in the elimination tree is given by the sequence

$$CLIST[j],\ CLIST[CLIST[j]], \cdots .$$

The space for this list can be reused for children lists of subsequent nodes. In other words, at step j, $CLIST$ stores the complete children list for the node x_j and partial children lists for nodes x_i, with $i > j$. This makes it possible to represent the n children lists during the course of the algorithm using only an n-vector. The symbolic factorization routine in SPARSPAK [22] uses this observation.

8.2. Symbolic factorization of A by rows. In an unpublished manuscript [60], Whitten describes a symbolic factorization algorithm that will determine the structure of the Cholesky factor L by rows. The algorithm can be viewed as using the row structure characterization of L in terms of row subtrees of the elimination tree (Theorem 3.5). For each row i of L, we traverse the row subtree $T_r[x_i]$ as in Algorithm 3.1 for computing the number of nonzeros in L. The nodes visited are collected to form the ith row structure of L. A detailed description of this approach is given by Tarjan and Yannakakis [59].

An interesting feature of this algorithm is its potential for parallelism. Assume that the parent vector of the elimination tree has been precomputed. Then the structures of all n rows can be computed totally independent of each other. Zmijewski and Gilbert [62] use this observation to design a parallel symbolic factorization scheme for message-passing multiprocessors.

8.3. Symbolic factorization of $A = M^TM$ using M. In the sparse orthogonal factorization of a matrix M [20], [38] or the static storage approach of sparse partial pivoting [27] of a square matrix M, it is necessary to compute the structure of the Cholesky factor of the symmetric matrix M^TM. One obvious approach is to construct explicitly the structure of $A = M^TM$ and apply the symmetric symbolic factorization algorithm to A.

In [28], George and Ng provide an efficient algorithm to perform the symbolic factorization of M^TM using the structure of the matrix M. This removes the redundant step of determining the structure of M^TM. Although their algorithm description does not involve elimination trees, their scheme can also be formulated nicely in terms of such trees. Here, we provide this alternative formulation.

To use the symbolic factorization of Algorithm 8.1 for the structure of the matrix $A = M^TM$, we need to know the adjacent set of each node x_j in $G(A)$. But this is related to the structure of M by the next result.

PROPOSITION 8.2.

$$Adj_{G(A)}(x_j) = \{x_i \neq x_j \mid m_{rj} \neq 0, m_{ri} \neq 0 \ \textit{for some row}\ r\}. \ \square$$

In words, the structure of row/column j of A is given by the union of all the row structures M_{r*} of M where m_{rj} is nonzero. A direct application of Algorithm 8.1 to $Adj_{G(A)}(x_j)$ will determine the symbolic factorization of $A = M^TM$. But forming the entire structures of $Adj_{G(A)}(x_j)$ (for $j = 1, \cdots, n$) from M could be expensive.

In what follows, we shall show that for the purpose of performing the symbolic factorization of M^TM, we only need to form part of $Adj_{G(A)}(x_j)$. Define

$$FAdj_{G(A)}(x_j) = \{x_i \neq x_j \mid m_{r1} = \cdots = m_{r,j-1} = 0, m_{rj} \neq 0, m_{ri} \neq 0 \text{ for some row } r\}.$$

To simplify our notation, we shall use $Adj(x_j)$ and $FAdj(x_j)$ to refer to $Adj_{G(A)}(x_j)$ and $FAdj_{G(A)}(x_j)$, where $A = M^TM$. It follows from the definition that

$$FAdj(x_j) \subseteq Adj(x_j)\,.$$

However, the next theorem says that as far as adapting Algorithm 8.1 to the symbolic factorization of M^TM is concerned, it is sufficient to use $\{FAdj(x_j)\}$ instead of $\{Adj(x_j)\}$.

THEOREM 8.3. *For any node x_j in the elimination tree of $T(M^TM)$, $Adj(T[x_j])$ is given by*

$$FAdj(x_j) \cup \{\, Adj(T[x_s]) - \{x_j\} \mid x_s \text{ is a child of } x_j \,\}\,.$$

Proof. It follows from Theorem 8.1 that it is sufficient to show that

$$Adj(x_j) - \{x_1, \cdots, x_{j-1}\} \subseteq FAdj(x_j) \cup \{Adj(T[x_s]) - \{x_j\} \mid x_s \text{ is a child of } x_j\}.$$

Consider $x_i \in Adj(x_j)$ with $i > j$. If $x_i \in FAdj(x_j)$, there is nothing to prove. Otherwise, by Proposition 8.2, there exists a row r of M and a subscript $k < j$ such that

$$m_{rk} \neq 0\,,\ m_{rj} \neq 0\,,\ m_{ri} \neq 0\,.$$

This implies that a_{jk}, a_{ik}, and a_{ij} are nonzeros. By Theorem 3.1 on the elimination tree of $A = M^TM$, the node x_i is an ancestor of x_j, which in turn is an ancestor of x_k. Let x_s be the child of x_j, which is also an ancestor of x_k. Since $a_{ik} \neq 0$, by Theorem 3.5, $\ell_{is} \neq 0$. By Theorem 3.7, $x_i \in Adj(T[x_s]) - \{x_j\}$. □

The following algorithm performs the symbolic factorization scheme using the structure of M. Its correctness follows from Theorem 8.3.

Algorithm 8.2. Symbolic Factorization of $A = M^TM$.

```
for j := 1 to n do
begin
   Adj(T[x_j]) := ∅ ;
   for each row r of M with first nonzero in location j do
      Adj(T[x_j]) := Adj(T[x_j]) ∪ {x_i ≠ x_j | m_ri ≠ 0} ;
   for x_s in the children list of x_j do
      Adj(T[x_j]) := Adj(T[x_j]) ∪ Adj(T[x_s]) − {x_j} ;
   if Adj(T[x_j]) ≠ ∅ then
   begin
      p := min { i | x_i ∈ Adj(T[x_j]) } ;
      add x_j to the children list of x_p
   end
end.
```

Note that Algorithm 8.2 is essentially the same as Algorithm 8.1. The only difference is the use of the **for**-loop to initialize the structure of the factor column to

$FAdj(x_j)$ instead of $Adj(x_j)$ as in Algorithm 8.1. This formulation connects the algorithm by George and Ng [28] with the elimination tree structure and the original symmetric symbolic factorization scheme. The use of an extra integer vector $CLIST[*]$ to keep track of the children nodes during the algorithm can also be adapted from Algorithm 8.1.

9. Elimination trees in numerical factorization.

9.1. Sparse column-Cholesky factorization. Most implementations of sparse Cholesky factorization use some variants of the *column-Cholesky* scheme, whereby the sparse Cholesky factor is computed and accessed by columns. They include the Harwell MAxx series of routines for sparse solvers, the Waterloo SPARSPAK [24], and the Yale sparse matrix package [12]. Specifically, we can express the column scheme algorithmically as follows:

for $i := 1$ **to** n **do**
begin

$$\begin{pmatrix} t_i \\ \vdots \\ t_n \end{pmatrix} := \begin{pmatrix} a_{ii} \\ \vdots \\ a_{ni} \end{pmatrix} - \sum_{k<i} \ell_{ik} \begin{pmatrix} \ell_{ik} \\ \vdots \\ \ell_{nk} \end{pmatrix}$$

$$\begin{pmatrix} \ell_{ii} \\ \vdots \\ \ell_{ni} \end{pmatrix} := \frac{1}{\sqrt{t_i}} \begin{pmatrix} t_i \\ \vdots \\ t_n \end{pmatrix}$$

end

In this formulation, the temporary vector $(t_i, \cdots, t_n)^T$ is used here only for clarity. Its storage can overlap with that of $(\ell_{ii}, \cdots, \ell_{ni})^T$. This formulation is applicable to both dense and sparse matrices. In the sparse case, contributions to column i come from those preceding columns of L with nonzero ℓ_{ik}. These are given precisely by the row structure of L_{i*}, which is shown to be a subtree in the elimination tree. Recall from §3 that $T_r[x_i]$ is the row subtree, and it is a pruned subtree of $T[x_i]$ rooted at the node x_i. Although the elimination tree has no direct role in the sparse column-Cholesky approach, it does offer the numerical dependencies on columns of the Cholesky factor. The column dependency in Proposition 2.2 can be justified as follows.

PROPOSITION 9.1. *The column L_{*i} can be computed using A_{*i} and those columns L_{*k}, where $x_k \in T_r[x_i]$.* □

9.2. The multifrontal method for symmetric systems.

9.2.1. Frontal matrices and the elimination tree. The *multifrontal method* by Duff and Reid [10] is an important advance in direct solution of sparse systems. To factor a given large sparse matrix, the method uses a novel way of reorganizing the computation so that the entire sparse Cholesky factorization is performed through the partial factorization of a sequence of dense and smaller submatrices. The reorganization is based on the structure of the elimination tree (Duff and Reid use a slight variant, called an *assembly tree*). As before, let $x_1, x_2, \cdots, x_n$ be the ordering of the nodes in the graph $G(A)$. Recall that the node x_j corresponds to the jth row/column of the matrix A.

In the multifrontal method, each node x_j is associated with a *frontal matrix* $\bar{F}_j$, which is dense and much smaller. The size of the frontal matrix $\bar{F}_j$ is given by the

number of nonzeros in column j of L

$$\eta(L_{*j}) = |\, Adj(T[x_j]) \cup \{x_j\} \,|\,;$$

and its *first* row/column corresponds to the node x_j. Once fully formed, the first row/column in $\bar{F}_j$ can be eliminated from this matrix. Let F_j denote the remaining frontal matrix after the elimination of x_j from $\bar{F}_j$. We now describe the relation among these frontal matrices $\{\bar{F}_j\}$ and $\{F_j\}$ using the structure of the elimination tree.

For a given row/column j, consider the matrix B obtained by first removing all the rows and columns of A except those in $T[x_j] \cup Adj(T[x_j])$, and then zeroing out all the entries except those in the rows and columns of $T[x_j]$. For example, consider the matrix and elimination tree of Fig. 6.1. For node $c = x_4$, the corresponding B matrix is given by

$$\begin{pmatrix} a & \bullet & & \bullet \\ \bullet & c & \bullet & \\ & \bullet & 0 & 0 \\ \bullet & & 0 & 0 \end{pmatrix}$$

Note that the last two rows/columns correspond to the nodes g and h. Here, we use "0" to emphasize that zero value is assigned to the location under consideration. We provide the following observations without proof.

PROPOSITION 9.2. *The frontal matrix $\bar{F}_j$ is the same as the remaining matrix after the rows/columns of $T[x_j] - \{x_j\}$ have been eliminated from B.* □

PROPOSITION 9.3. *The matrix F_j is the same as the remaining matrix after the rows/columns of $T[x_j]$ have been eliminated from B.* □

From Proposition 9.3, the matrix F_j contains updates from the elimination of columns associated with $T[x_j]$. Moreover, these observations imply that the frontal matrix $\bar{F}_j$ can be assembled (or formed) using column A_{*j} of the given matrix and the remaining frontal matrices $\{F_s\}$ of its children nodes (if any) in the elimination tree. The algorithmic form of the multifrontal method can then be described in terms of the elimination tree as follows.

Algorithm 9.1. Multifrontal Method.

```
for j := 1 to n do
begin
   allocate space for the frontal matrix F̄_j ;
    for  each child x_s of x_j in the elimination tree T(A)   do
        assemble the remaining frontal matrix F_s into F̄_j ;
   assemble column A_{* j} into F̄_j ;
   perform one step of Cholesky factorization on F̄_j to give L_{* j} and F_j
end.
```

The assembly of the remaining frontal matrix F_s or column A_{*j} into $\bar{F}_j$ is performed by simply adding nonzero entries from F_s or A_{*j} into appropriate locations of $\bar{F}_j$. Note that the first row/column of the frontal matrix $\bar{F}_j$ corresponds to the node x_j, whose elimination from $\bar{F}_j$ gives the nonzero entries of L_{*j}. Therefore, the first column of the frontal matrix $\bar{F}_j$ is (logically) full. This implies that the remaining submatrix F_j is also full. This is perhaps the main characteristic of the multifrontal

method, which allows sparse factorization be performed via a sequence of full submatrices.

The multifrontal method was originally developed for the direct solution of symmetric indefinite sparse linear systems [10]. But it is already clear from this paper of Duff and Reid that the scheme can be used for sparse Cholesky factorization. In [51], Reid adapts this method in his TREESOLVE package, an out-of-core factorization scheme for large, sparse finite-element systems. Duff and Reid [9] demonstrate that the multifrontal method is especially effective on vector machines, such as the CRAY. Recently, Liu [41] showed that the method even has significant advantage over conventional sparse Cholesky schemes on virtual memory paging systems. The readers are referred to [2] and [3] for more recent development of the multifrontal method.

The role of the elimination tree in the multifrontal method should be clear from Algorithm 9.1. It provides the assembly connection among the frontal matrices. Indeed, each tree edge in the elimination tree corresponds to one assembly operation of some F_s into its parent's frontal matrix.

9.2.2. Matrix reordering for the multifrontal method. Implicitly assumed in Algorithm 9.1 is that in the assembly of the frontal matrix $\bar{F}_j$, the remaining frontal matrices $\{F_s\}$ of its children nodes are readily available. In other words, after performing one step of Cholesky factorization on $\bar{F}_j$, we need to store the remaining submatrix F_j for the assembly of its parent's frontal matrix. To reduce the amount of such storage, it is important to use F_j as early as possible to release its storage space for subsequent remaining frontal submatrices. This can be achieved by using a postordering of the elimination tree $T(A)$ of the matrix A. With a postordering, the logistics can be handled nicely by the use of a stack of remaining frontal matrices, as suggested by Duff and Reid [10]. Each F_s when required is always at the top of the stack. On the other hand, when F_j is formed, it is simply pushed into this stack. This stack of full triangular matrices (each F_j is symmetric) represents the working storage overhead for the implementation of the multifrontal method.

In [39], Liu provides an analysis of this working storage requirement in terms of the elimination tree structure. Based on the analysis, it is observed that the way in which children nodes are arranged can also affect the amount of working storage. But the reordering obtained from a rearrangement of children nodes still corresponds to a topological ordering of the elimination tree. By Theorem 6.1, it is an equivalent reordering (that is, it preserves the filled graph). Therefore, we are free to rearrange children nodes in the elimination tree.

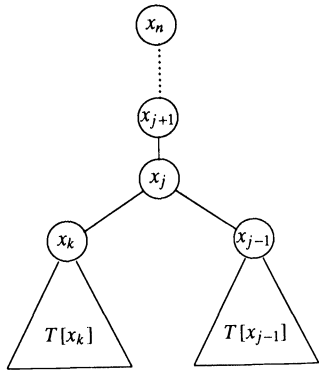
Liu [39] shows that an optimal child sequence can be obtained by arranging the children nodes in descending order of some easily computed quantities. Significant reduction in working storage is achieved with a small investment in reordering time. This technique is recommended for any implementation of the multifrontal method.

This storage reduction technique uses a reordering that preserves not only the filled graph, but also the elimination tree. If we remove the restriction on preserving the elimination tree, further working storage reduction is possible. In [43], Liu applies reorderings that restructure the elimination tree (as discussed in §6.3) to achieve more saving in this working storage overhead. The technique used will obtain an equivalent reordering (preserving the filled graph), that will give an "*unbalanced*" elimination tree. For the justification of this, we refer the reader to the paper. For an illustration, we note that the reordering given in Fig. 6.2 is a more desirable ordering than the one in Fig. 6.1 in terms of working storage requirement.

9.3. Minimal storage sparse elimination algorithm. In reference [14], Eisenstat, Schultz, and Sherman propose an interesting variant of sparse Gaussian elimination for machines with limited core storage (without the use of auxiliary storage). The method trades an increase in computation for a decrease in storage by recomputing rather than saving most nonzeros in the triangular factor matrix. They have appropriately called it the *minimal storage sparse elimination* algorithm.

Central to their algorithm is a structure called the *element merge tree.* The tree structure can be interpreted as a variant of the elimination tree. We shall describe this minimal storage algorithm in terms of the elimination tree. Let A be the given n-by-n sparse symmetric positive-definite matrix and let $T(A)$ be its elimination tree. We assume that the ordering for A is already a postordering on the elimination tree $T(A)$.

Let x_j be the *last* node that has two or more children nodes. To simplify the exposition, we assume that x_j has two children nodes, one of them must be x_{j-1} (a property of a postordering). Let x_k be the other child. Pictorially, we can view that tree as follows:



The crucial observation in the scheme by Eisenstat, Schultz, and Sherman is that if solutions to the variables $\{x_j, \cdots, x_n\}$ are known, the problem can be reduced into two smaller independent systems, one involving variables in the subtree $T[x_k]$ and the other $T[x_{j-1}]$.

In matrix terms, we note that the given linear system can be partitioned as follows:

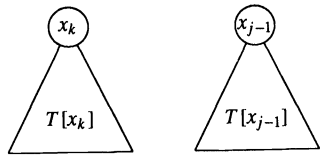
$$\begin{pmatrix} A_u & 0 & E_u^T \\ 0 & A_v & E_v^T \\ E_u & E_v & A_w \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \begin{pmatrix} b_u \\ b_v \\ b_w \end{pmatrix} ,$$

where the solution vector x is partitioned into three parts u, v, and w. The subvector u corresponds to unknowns $x_1, \cdots, x_k$ in the subtree $T[x_k]$, v to $x_{k+1}, \cdots, x_{j-1}$ in $T[x_{j-1}]$, and w to $x_j, \cdots, x_n$. The matrix A and right-hand vector b are partitioned accordingly. The zero block submatrix in A follows from the fact that the two subtrees $T[x_k]$ and $T[x_{j-1}]$ are disjoint and Corollary 3.2. It is easy to see that if the solution w is known, the linear system is reduced to two smaller subsystems:

$$A_u\, u \;=\; b_u - E_u^T\, w\,, \quad A_v\, v \;=\; b_v - E_v^T\, w\,.$$

The elimination trees for the submatrices A_u and A_v are given by the subtrees $T[x_k]$ and $T[x_{j-1}]$, respectively. The scheme can then be interpreted in terms of the elimi-

nation tree as a removal of the nodes $\{x_j, \cdots, x_n\}$ from the elimination tree $T(A)$ to yield two trees for the independent subsystems:



Note that these two subsystems can be solved by applying the same approach recursively. The objective then becomes the determination of the variables $\{x_j, \cdots, x_n\}$ in w with limited core storage. The scheme by Eisenstat, Schultz, and Sherman can be best described using frontal matrices as described in §9.1. Indeed, [14] is perhaps the first paper that uses the multifrontal principle. We describe their overall scheme as follows.

Step 1. Determine the remaining frontal matrix F_k by eliminating nodes in the subtree $T[x_k]$.

Step 2. Determine the remaining frontal matrix F_{j-1} by eliminating nodes in the subtree $T[x_{j-1}]$.

Step 3. Subtract F_k and F_{j-1} from appropriate entries in A_w to give the modified submatrix $\bar{A}_w$. Let b_w be correspondingly modified to become $\bar{b}_w$. Then solve the linear subsystem $\bar{A}_w w = \bar{b}_w$.

Step 4. Solve $A_u u = b_u - E_u^T w$, and $A_v v = b_v - E_v^T w$.

The important point here is that during the formation of the frontal matrices F_k and F_{j-1}, the columns of L are discarded as they are computed. This saves storage for the factor matrix; and in exchange, part of columns of L have to be recomputed when solving for the unknowns in u and v.

9.4. Row merging scheme for orthogonal decomposition. In [38], Liu introduces the notion of a *row merge tree* for sparse orthogonal decomposition by Givens rotations. In that paper, the connection between a row merge tree and an elimination tree is addressed, and the relation of row merging for sparse QR factorization with the multifrontal method for sparse Cholesky factorization is also briefly mentioned. Here, we give a more detailed account of the relation among elimination tree, row merging, and the multifrontal method.

Let M be an m-by-n large sparse matrix with full column rank. The problem is to determine the triangular factor R in the orthogonal decomposition of M into

$$Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where Q is an m-by-m orthogonal matrix. It is well known that the factor R is mathematically equivalent to the Cholesky factor of the symmetric matrix $A = M^T M$.

Assume that the columns of M have been prearranged by some fill-reducing ordering on $A = M^T M$ (as suggested by George and Heath [20]). Let $T(A)$ be the elimination tree of A. The general row merging scheme [38] will organize computation for the sparse orthogonal decomposition using the structure of the elimination tree. The way computation is organized is similar to the multifrontal method for sparse Cholesky factorization. Indeed, it is appropriate to view the general row merging scheme as one that performs the multifrontal method on $A = M^T M$ without ever forming A. Instead, the given matrix M is used.

In the multifrontal method (Algorithm 9.1), each node x_j is associated with a frontal matrix $\bar{F}_j$, and a logically full submatrix F_j. At step j, the numerical computation consists of the formation of the frontal matrix $\bar{F}_j$ through assembly, and the elimination of a variable from this frontal matrix to give F_j.

In the case of the general row merging scheme for the given matrix M, the computation associated with each node x_j of the elimination tree $T(M^TM)$ is a merging process involving a set of full upper trapezoidal submatrices together with some original rows of the matrix M. The original rows from M are those with the first nonzero in location j; while each child node of x_j will contribute one such upper trapezoidal submatrix to the merging operation. The merging operation uses a sequence of Givens rotations to reduce the set of upper trapezoidal submatrices and selected rows from M, to form another full upper trapezoidal submatrix, part of which will in turn be passed to the parent node of x_j.

Let us denote by $\bar{Z}_j$ the full upper trapezoidal submatrix for node x_j and by Z_j the remaining submatrix after the removal of the first row from $\bar{Z}_j$. We state without proof the following observations on $\bar{Z}_j$ and Z_j.

PROPOSITION 9.4. *The size of the submatrix $\bar{Z}_j$ $\{Z_j\}$ is the same as that of $\bar{F}_j$ $\{F_j\}$.* □

PROPOSITION 9.5. *If QR is the orthogonal decomposition of the matrix M, then the jth row of the upper triangular factor matrix is given by the first row of $\bar{Z}_j$.* □

The general row merging scheme can be described in terms of the submatrices $\bar{Z}_j$ and Z_j. Note the similar logic structure as in the multifrontal method of Algorithm 9.1.

Algorithm 9.2. General Row Merging Scheme.

```
for j := 1 to n do
begin
   allocate space for the upper trapezoidal submatrix Z̄_j;
   for each child x_s of x_j in the elimination tree T(M^T M) do
      merge the remaining trapezoidal submatrix Z_s into Z̄_j;
   for each row r of M with first nonzero at location j do
      merge row M_{r *} into Z̄_j;
   remove the first row of Z̄_j to give the jth row of R and Z_j;
end.
```

The similarity between Algorithm 9.2 and Algorithm 9.1 is clear. Essentially, the "assemble" operation is replaced by the "merge" operation. It is also interesting to point out that Algorithm 9.2 can be viewed simply as an extension of the symbolic factorization scheme for $A = M^TM$ in Algorithm 8.2. Logical manipulation of the structure of M is now replaced by the actual numerical computation involved in forming rows of the upper triangular factor R. In this paper, we are concerned with the exposition of the connection between the general row merging scheme and the elimination tree. For details of the merging scheme, the reader is referred to [23] and [38].

9.5. Factorization of symmetric indefinite systems. In [10], Duff and Reid introduce the multifrontal method as an efficient scheme to solve sparse symmetric indefinite systems. Let A be an n-by-n symmetric indefinite matrix. The elimination tree structure $T(A)$ is still defined. Assume that the ordering on A is already a postordering on the tree $T(A)$.

If the matrix A were positive definite, the multifrontal method would form the frontal matrix $\bar{F}_j$ at step j and proceed to eliminate variable x_j from this frontal matrix. However, since A is now indefinite, the variable x_j may not be a suitable numerical pivot in the frontal matrix $\bar{F}_j$ (though it is a good structural pivot). In such case, Duff and Reid allow the frontal matrix $\bar{F}_j$ to be extended to include more rows/columns, thereby enlarging the set of pivot candidates.

As noted by Liu [44], the scheme of Duff and Reid uses the technique of *delayed elimination.* When a node is deemed as unsuitable to be a numerical pivot, its elimination will be delayed to a later stage. This is different from the conventional approach for factoring dense indefinite matrices, where *advanced* elimination is used to eliminate the suitable node as part of a 2×2 block pivot [5].

The connection between the elimination tree and delayed elimination is discussed in [44]. The elimination tree represents a class of ideal elimination sequences in which no pivoting is performed. With the added stability requirement, we can still use the elimination tree structure to provide a selection guide for 1×1 and 2×2 pivots. At step j, pivots can only be selected from the uneliminated nodes of the subtree $T[x_j]$. If no satisfactory pivots can be found, the scheme will proceed to work on the next node x_{j+1}. On the other hand, if uneliminated nodes from this subtree are selected as pivots, the amount of structural damage to the resulting tree will be contained. In this way, the resulting sequence of stable pivots will form an elimination tree that deviates as little as possible from the original tree. We shall refer the interested readers to [44] for more details.

10. Elimination trees in computing environments.

10.1. Factorization on a paging environment. In §6, we consider equivalent matrix reorderings on a given sparse matrix A using the structure of its elimination tree $T(A)$. They all preserve the structure of the filled graph of A, and will require the same amount of storage and computational cost in factoring A. Those based on topological orderings on the tree $T(A)$ will also preserve the elimination tree structure as considered in §6.1. There are also some that will restructure the tree, as given by those from rotations in §6.2.

For a given computing environment, a meaningful question to ask is to find the "best" equivalent reordering for the factorization of the sparse matrix A. Even finding a "good" equivalent reordering for a particular environment will be of practical significance. In this section, we consider some common computing environments and offer some suggestions for obtaining "good" reorderings. The recommendations are based on intuitive understanding of the environment and the factorization process, and are further substantiated by numerical experiments. Further work is still needed for theoretical justification of these recommendations. The recent work by Pothen [49] is on the complexity of finding optimal elimination trees.

We first consider the sparse Cholesky factorization on a virtual memory paging system. In this environment, it is desirable to have columns of the matrix organized so as to reduce the amount of paging activity. This obviously depends on the numerical factorization algorithm used. In [42], Liu considers the conventional sparse column-Cholesky factorization scheme as discussed in §9.1.

From Proposition 9.1, in the factorization scheme by columns, each column of the factor matrix is computed by a number of column modifications, governed by the locations of the nonzeros in its corresponding row. Each row structure is itself a (pruned) subtree of the elimination tree $T(A)$. Therefore, a reasonable recommendation is to order nodes in each subtree consecutively, so that columns for each subtree

will be stored in contiguous memory locations. In this way, columns required for the computation of L_{*j} will reside relatively close to column j. This tends to improve on the *locality* of the factorization scheme.

A postordering on the elimination tree is an appropriate choice. It is demonstrated that for some practical problems, the amount of CPU time for factorization can differ by almost 100 percent for two equivalent reorderings [42].

10.2. Out-of-core sparse factorization. In the situation when the storage requirement of the matrix factor L of A exceeds the space available in main memory, it is necessary to exploit the use of auxiliary storage. The multifrontal method as described in Algorithm 9.1 lends itself readily to such an environment. After performing one step of Cholesky factorization on the frontal matrix $\bar{F}_j$ to give L_{*j} and F_j, the out-of-core version will store the factor column L_{*j} into secondary storage. This will not affect subsequent steps of the factorization, since this column L_{*j} is no longer necessary. Indeed, contributions from it have already been accumulated in the remaining frontal matrix F_j.

The amount of primary storage to perform the entire factorization of this out-of-core multifrontal method is precisely the same as the working storage requirement for the in-core version. Therefore, the use of tree rotations to give an "unbalanced" tree structure and the technique of optimal children resequencing on the resulting elimination tree can also be applied here to reduce the amount of storage for the out-of-core scheme. The relevance of the elimination tree is clear for this situation.

In [40], Liu proposes an alternative general sparse out-of-core scheme, which is based on reorganization of the data storage vector during the course of sparse factorization by columns. Column segments of the factor matrix that are no longer needed will be discarded to make room for new columns. The elimination tree is again a relevant structure in determining the primary storage requirement. The techniques of unbalanced tree and optimal children resequencing are also applicable here, although the formula for storage requirement is slightly different. Interested readers can consult [40] for the explicit formula.

10.3. Models for parallel factorization. Parallel machines offer a different computing environment. The use of multiprocessors in the parallel Cholesky factorization of large sparse matrices has attracted the attention of many researchers. Jess and Kees [34] use the elimination tree structure as a large-grained task model for sparse factorization. In [37], Liu presents a systematic and unified treatment of computational task models for parallel sparse factorization. It is shown that the elimination tree is an appropriate large grained model irrespective of whether computation is performed by row, by column or by submatrix. The next result follows directly from Proposition 9.1.

PROPOSITION 10.1. *Let $T[x_i]$ and $T[x_j]$ be two disjoint subtrees of the elimination tree $T(A)$. The columns L_{*i} and L_{*j} can be computed in parallel with no overlap in data access.* □

COROLLARY 10.2. *All the leaf nodes in the elimination tree can be eliminated in parallel.* □

Nodes that can be eliminated in parallel are called *parallel pivots.* In this context, it is desirable to maximize the number of parallel pivots for each step without sacrificing the fill-reducing quality of the ordering. Here, each step refers to the elimination of a set of parallel pivots. The approach that Jess and Kees [34] take is to consider *equivalent* matrix reorderings that will maximize the number of parallel pivots. They provide an algorithm to determine such an equivalent ordering using the filled graph

of the matrix. In [47], a linear algorithm is given that achieves the same result. Furthermore, it is shown in [46] that the resulting elimination tree has minimum height among all trees from the class of equivalent reorderings. Recent works by Lewis and Peyton [35] and Pothen [50] provide further insight into the algorithm by Jess and Kees and contain improvement to [47].

Another recent use of elimination tree in the context of parallel factorization is in the assignment of computational tasks to individual processors in a multiprocessor environment. For shared memory architectures, Liu [37] shows how the structure of the elimination tree can be used to schedule computational tasks to the processors. In the case of local memory systems, it is desirable to assign tasks to processors to achieve load balancing and to reduce communication traffic between processors. In [25], George, Liu, and Ng show how an assignment of tasks to processor nodes of a hypercube architecture can be performed to satisfy these objectives. The key idea used is to map tasks associated with a subtree of the elimination tree to a subcube of the hypercube.

Duff [7] considers the parallel version of the multifrontal method. The way in which elimination can be performed in multiple independent fronts is governed by the structure of the elimination tree. Other recent works on parallel sparse factorization that involve the elimination tree include George et al [19], Gilbert and Hafsteinsson [29], and Zmijewski [61].

11. Other related notions and future research directions. It is quite evident that the elimination tree is a valuable structure in the study of sparse matrix factorization. Our exploration of its usage in this paper is by no means exhaustive. Some useful block partitionings have been defined based on the elimination tree structure. One of them has been referred to as a *supernodal partitioning* [2], [3]. In its simplest form, a supernodal partitioning can be defined as follows. Let x_p be the parent node of x_j in the elimination tree. These two nodes x_j and x_p belong to the same block in the partitioning if

$$Adj(T[x_j]) \;=\; Adj(T[x_p]) \cup \{x_p\}\,,$$

and x_j is the only child of x_p. (Note that by Theorem 7.1 the above condition is equivalent to $\mid Adj(T[x_j]) \mid = \mid Adj(T[x_p]) \mid +1$.) The set of nodes in each block is collectively referred to as a *supernode*. It can be readily verified that each supernode defines a clique in the filled graph and they share the same set of adjacent nodes outside the clique. Furthermore, the nodes of each supernode corresponds to a chain in the elimination tree. The notion of supernodal partitioning plays a central role in devising highly efficient multifrontal codes on vector machines [2], [3].

The compressed column storage scheme by Sherman [56] can also be viewed as using a generalization of this notion of supernodes. If we number the nodes of each supernode consecutively, each supernode corresponds to a full diagonal block of the Cholesky factor. Since the columns associated with each supernode have identical column structures outside the diagonal block, we need to store only one copy of the column structures for each supernode. The success of Sherman's compressed column storage scheme can be attributed mostly to this property of supernodes.

In his thesis [48], Peters considers a different partitioning based on the elimination tree structure. If x_p is the parent node of x_j, these two nodes belong to the same partition if x_j is the only child of x_p. He refers to it as a *proper perfect preserving partition.* His motivation is to use the partition to devise a solution method that only requires full envelope (or profile) solves.

In this paper, we have not discussed the notion of *clique trees* [35] defined for chordal graphs (or filled graphs). A clique tree has a structure closely related to that of an elimination tree. An application of it to sparse matrix reordering for parallel elimination can be found in [35].

We have discussed some of the roles of elimination trees in sparse QR and LU factorizations in §§7.3, 8.3, 9.4, and 9.5. The direct solution of unsymmetric sparse linear systems is a relatively less developed area. With more advances in sparse QR and LU algorithms for unsymmetric systems in the future, we expect to see more impact from this tree structure.

The elimination tree structure provides information on data dependency in the factorization process. It captures the essential ingredient for parallel elimination. With the explosion in research work on parallel algorithms, this tree will definitely play a central role in future development. The height of the elimination tree represents an effective but crude measure for the amount of work in parallel elimination. Finding practical and more refined criteria in terms of the tree will be a fruitful research area.

Another important research direction is in the characterizations of the best elimination tree structure for a given computational environment, be it on a parallel architecture, a vector machine, or a virtual memory system. Current recommendations on desirable structures are mostly based on intuition and experience. A vigorous approach to provide theoretical justification seems to be an interesting and important area.

Acknowledgments. The author would like to thank John Gilbert and the two referees for their careful and critical reading of the original manuscript.

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *Data Structures and Algorithms*, Addison-Wesley, Reading, MA, 1983.

[2] 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.

[3] C. ASHCRAFT AND R. GRIMES, *The influence of relaxed supernode partitions on the multifrontal method*, Tech. Report ETA-TR-60, Engineering Technology Applications Division, Boeing Computer Services, Seattle, WA, 1987.

[4] R. E. BANK AND R. K. SMITH, *General sparse elimination requires no permanent integer storage*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 574–584.

[5] J. R. BUNCH AND L. KAUFMAN, *Some stable methods for calculating inertia and solving symmetric linear equations*, Math. Comp., 31 (1977), pp. 163–179.

[6] I. S. DUFF, *Full matrix techniques in sparse Gaussian elimination*, in Lecture Notes in Mathematics (912), G. Watson, ed., Springer-Verlag, Berlin, New York, 1982, pp. 71–84.

[7] ———, *Parallel implementation of multifrontal schemes*, Parallel Comput., 3 (1986), pp. 193–204.

[8] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 1987.

[9] I. S. DUFF AND J. K. REID, *Experience of sparse matrix codes on the CRAY-1*, Comput. Phys. Comm., 26 (1982), pp. 293–302.

[10] ———, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[11] ———, *A note on the work involved in no-fill sparse matrix factorization*, IMA J. Numer. Anal., 3 (1983), pp. 37–40.

[12] S. C. EISENSTAT, M. C. GURSKY, M. H. SCHULTZ, AND A. H. SHERMAN, *The Yale sparse matrix package, 1. The symmetric code*, Internat. J. Numer. Meth. Engrg., 18 (1982), pp. 1145–1151.

[13] S. C. EISENSTAT, M. H. SCHULTZ, AND A. H. SHERMAN, *Applications of an element model for Gaussian elimination*, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 85–96.

[14] ———, *Software for sparse Gaussian elimination with limited core storage*, in Sparse Matrix Proceedings, I. S. Duff and G. W. Stewart, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979, pp. 135–153.

[15] ———, *Algorithms and data structures for sparse symmetric Gaussian elimination*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 225–237.

[16] M. J. FISCHER, *Efficiency of equivalence algorithms*, in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 153–167.

[17] F. GAVRIL, *The intersection graphs of subtrees in trees are exactly the chordal graphs*, J. Combin. Theory Ser. B, 16 (1974), pp. 47–56.

[18] J. A. GEORGE, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.

[19] J. A. GEORGE, M. HEATH, J. W. H. LIU, AND E. NG, *Sparse Cholesky factorization on a local-memory multiprocessor*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 327–340.

[20] J. A. GEORGE AND M. T. HEATH, *Solution of sparse linear least squares problems using Givens rotations*, Linear Algebra Appl., 34 (1980), pp. 69–83.

[21] J. A. GEORGE AND J. W. H. LIU, *An optimal algorithm for symbolic factorization of symmetric matrices*, SIAM J. Comput., 9 (1980), pp. 583–593.

[22] ———, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[23] ———, *Householder reflection versus Givens rotations in sparse orthogonal decomposition*, Linear Algebra Appl., 88/89 (1987), pp. 223–238.

[24] J. A. GEORGE, J. W. H. LIU, AND E. NG, *User guide for SPARSPAK, Waterloo Sparse Linear Equations Package*, Tech. Report Res. CS-78-30 (revised 1980), Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1980.

[25] ———, *Communication reduction in parallel sparse Cholesky factorization on a hypercube*, in Hypercube Multiprocessors 1987, M. T. Heath, ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987, pp. 576–586.

[26] ———, *A data structure for sparse QR and LU factorization*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 100–121.

[27] J. 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.

[28] ———, *Symbolic factorization for sparse Gaussian elimination with partial pivoting*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 877–898.

[29] J. R. GILBERT AND H. HAFSTEINSSON, *A parallel algorithm for finding fill in a sparse symmetric matrix*, Tech. Report TR 86-789, Department of Computer Science, Cornell University, Ithaca, NY, 1986.

[30] J. R. GILBERT, D. J. ROSE, AND A. EDENBRANDT, *A separator theorem for chordal graphs*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 306–313.

[31] J. R. GILBERT AND R. E. TARJAN, *The analysis of a nested dissection algorithm*, Numer. Math., 50 (1987), pp. 377–404.

[32] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[33] J. E. HOPCROFT AND R. E. TARJAN, *Dividing a graph into triconnected components*, SIAM J. Comput., 2 (1973), pp. 135–158.

[34] J. A. G. JESS AND H. G. M. KEES, *A data structure for parallel L/U decomposition*, IEEE Trans. Comput., C-31 (1982), pp. 231–239.

[35] J. G. LEWIS AND B. W. PEYTON, *A fast implementation of the Jess and Kees algorithm*, Tech. Report ETA-TR-90, Engineering and Scientific Services Division, Boeing Computer Services, Seattle, WA, 1988.

[36] J. W. H. LIU, *A compact row storage scheme for Cholesky factors using elimination trees*, ACM Trans. Math. Software, 12 (1986), pp. 127–148.

[37] ———, *Computational models and task scheduling for parallel sparse Cholesky factorization*, Parallel Comput., 3 (1986), pp. 327–342.

[38] ———, *On general row merging schemes for sparse Givens transformations*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1190–1211.

[39] ———, *On the storage requirement in the out-of-core multifrontal method for sparse factorization*, ACM Trans. Math. Software, 12 (1986), pp. 249–264.

[40] ———, *An adaptive general sparse out-of-core Cholesky factorization scheme*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 585–599.

[41] ———, *The multifrontal method and paging in sparse Cholesky factorization*, Tech. Report CS-87-09, Department of Computer Science, York University, North York, Ontario, Canada, 1987.

[42] ———, *A note on sparse factorization in a paging environment*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 1085–1088.

[43] ———, *Equivalent sparse matrix reordering by elimination tree rotations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 424–444.

[44] ———, *A tree model for sparse symmetric indefinite matrix factorization*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 26–39.

[45] ———, *A graph partitioning algorithm by node separators*, ACM Trans. Math. Software, 15 (1989), pp. 198–219.

[46] ———, *Reordering sparse matrices for parallel elimination*, Parallel Comput., 11 (1989), pp. 73–91.

[47] J. W. H. LIU AND A. MIRZAIAN, *A linear reordering algorithm for parallel pivoting of chordal graphs*, SIAM J. Discrete Math., 2 (1989), pp. 100–107.

[48] F. J. PETERS, *Sparse matrices and substructures*, Tech. Report Mathematical Centre Tracts 119, Matematisch Centrum, Amsterdam, the Netherlands, 1979.

[49] A. POTHEN, *The complexity of optimal elimination trees*, Tech. Report CS-88-16, Department of Computer Science, The Pennsylvania State University, University Park, PA, 1988.

[50] ———, *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.

[51] J. K. REID, TREESOLVE, *a Fortran package for solving large sets of linear finite element equations*, Tech. Report CSS 155, Computer Sciences and Systems Division, AERE Harwell, Oxfordshire, 1984.

[52] 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. Read, ed., Academic Press, New York, 1972, pp. 183–217.

[53] 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.

[54] D. J. ROSE, G. F. WHITTEN, A. H. SHERMAN, AND R. E. TARJAN, *Algorithms and software for in-core factorization of sparse symmetric positive definite matrices*, Computers & Structures, 11 (1980), pp. 597–608.

[55] R. SCHREIBER, *A new implementation of sparse Gaussian elimination*, ACM Trans. Math. Software, 8 (1982), pp. 256–276.

[56] A. H. SHERMAN, *On the efficient solution of sparse systems of linear and nonlinear equations*, PhD thesis, Department of Computer Science, Yale University, New Haven, CT, 1975.

[57] R. E. TARJAN, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.

[58] ———, *Data Structures and Network Algorithms*, CBMS–NSF Regional Conference Series in Applied Mathematics 44, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[59] 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.

[60] G. F. WHITTEN, *Computation of fill-in for sparse symmetric positive definite matrices*, Tech. Report Unpublished manuscript, Computer Science Department, Vanderbilt University, Nashville, TN, 1978.

[61] E. ZMIJEWSKI, *Sparse Cholesky factorization on a multiprocessor*, PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1987.

[62] E. ZMIJEWSKI AND J. R. GILBERT, *A parallel algorithm for large sparse Cholesky factorization on a multiprocessor*, Tech. Report TR 86-733, Department of Computer Science, Cornell University, Ithaca, NY, 1986.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 173–179, April 1990

ON PERHERMITIAN MATRICES*

RICHARD D. HILL†, RONALD G. BATES‡, AND STEVEN R. WATERS§

Abstract. A body of theory for perhermitian and skew-perhermitian matrices is developed. Some basic results for these matrices, their spectral properties, and characterizations of linear transformations that preserve them are given.

Key words. hermitian, linear transformation, spectral

AMS(MOS) subject classifications. 15A04, 15A18, 15A57, 15A99

1. Preliminaries. We denote the space of $n \times n$ complex matrices by $\mathcal{M}_n$ and the subset of hermitian [skew-hermitian] matrices by $\mathcal{H}_n$ [$\mathcal{H}_n^-$]. A matrix $A \in \mathcal{M}_n$ is said to be *perhermitian* [*skew-perhermitian*] if and only if $a_{ij} = \bar{a}_{n-j+1,n-i+1}$ [$a_{ij} = -\bar{a}_{n-j+1,n-i+1}$], $i, j = 1, \cdots, n$. We denote the set of perhermitian [skew-perhermitian] matrices by $\mathcal{PH}_n$ [$\mathcal{PH}_n^-$]. A matrix $A \in \mathcal{M}_n$ is said to be *perdiagonal* if and only if $a_{ij} = 0$ whenever $i + j \neq n + 1$, $i, j = 1, \cdots, n$. In particular, we shall use $J = (\delta_{i,n-j+1})$ to denote the unit perdiagonal matrix that has 1's on the secondary diagonal (i.e., the diagonal from upper-right to lower-left) and 0's elsewhere.

Perhermitian matrices are a natural generalization of the real persymmetric matrices that are discussed by Golub and Van Loan (cf. p. 125 of [10]). Cantoni and Butler [2], Goldstein [8], [9], and Lee [17] have investigated the eigenstructure and other properties of certain proper subsets of $\mathcal{PH}_n$. Real Toeplitz matrices, another subset of $\mathcal{PH}_n$, have been studied in a wide variety of contexts (cf. [5], [7], [11], [21], [22], and [23]).

In this paper we develop a body of theory for perhermitian and skew-perhermitian matrices. In particular, we develop some basic results for these matrices, consider their spectral properties, and characterize linear transformations that leave the set of perhermitian matrices invariant.

A companion paper [13] on centrohermitian matrices follows. (A matrix $A \in \mathcal{M}_n$ is said to be *centrohermitian* if $a_{ij} = \bar{a}_{n-i+1,n-j+1}$, $i, j = 1, \cdots, n$.) While in some cases the theory for centrohermitian matrices parallels the development here, in others it is strikingly different. The interface between perhermitian and centrohermitian matrices is also discussed in [13].

2. Basic results. In this section we shall enumerate many of the basic facts concerning perhermitian [skew-perhermitian] matrices, beginning with a characterization.

2.1. For $A \in \mathcal{M}_n$, the following are equivalent:

(i) $A \in \mathcal{PH}_n$ [$A \in \mathcal{PH}_n^-$]
(ii) $A = JA^*J$ [$A = -JA^*J$]
(iii) $JA \in \mathcal{H}_n$ [$JA \in \mathcal{H}_n^-$]
(iv) $AJ \in \mathcal{H}_n$ [$AJ \in \mathcal{H}_n^-$]
(v) $iA \in \mathcal{PH}_n^-$ [$iA \in \mathcal{PH}_n$]

We observe that if $A \in \mathcal{PH}_n$ [$\mathcal{PH}_n^-$], then so are $\bar{A}$, A^*, and A^{tr}. Also, by (ii), a perhermitian matrix A is seen to be unitarily similar to its conjugate transpose, A^*. Further, by (v), results for skew-perhermitian matrices may be immediately obtained from those for perhermitian matrices, and conversely.

* Received by the editors August 1, 1988; accepted for publication (in revised form) May 12, 1989.
† Department of Mathematics, Idaho State University, Pocatello, Idaho 83201.
‡ Department of Mathematics, Hartnell College, Salinas, California 93901.
§ Department of Mathematics, Pacific Union College, Angwin, California 94508.

2.2. If $A_1, \cdots, A_s \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$] and $c_1, \cdots, c_s \in \mathbb{R}$, then $\sum_{j=1}^{s} c_j A_j \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$]. It follows that $\mathscr{PH}_n$ and $\mathscr{PH}_n^-$ are both real vector spaces.

2.3. Letting $j, k = 1, \cdots, n$, we have that the $(n(n-1))/2$ matrices $\{E_{jk} + E_{n-k+1,n-j+1}\}_{j+k \leqq n}$, the $(n(n-1))/2$ matrices $\{iE_{jk} - iE_{n-k+1,n-j+1}\}_{j+k \leqq n}$, and the n matrices $\{E_{j,n-j+1}\}$ form a basis for $\mathscr{PH}_n$ over $\mathbb{R}$. Thus, $\mathscr{PH}_n$ is of dimension n^2 as a real vector space. Analogously, the n^2 matrices $\{E_{jk} - E_{n-k+1,n-j+1}\}_{j+k \leqq n}$, $\{iE_{jk} + iE_{n-k+1,n-j+1}\}_{j+k \leqq n}$, and $\{iE_{j,n-j+1}\}$ form a basis for $\mathscr{PH}_n^-$ over $\mathbb{R}$.

2.4. If A is perdiagonal, then $\bar{A}A, A\bar{A} \in \mathscr{PH}_n$.

2.5. If $A, B \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$], then $AB \in \mathscr{PH}_n$ if and only if $AB = BA$. This precludes any multiplicative structure (ring or algebra) on $\mathscr{PH}_n$ [$\mathscr{PH}_n^-$]. Note the difference from the set of centrohermitian matrices which does form an algebra (cf. result 2.5 of [13]).

2.6. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$] and $B \in \mathscr{PH}_n^-$ [$\mathscr{PH}_n$], then $AB \in \mathscr{PH}_n^-$ if and only if $AB = BA$.

2.7. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$] and A is nonsingular, then $A^{-1} \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$]. In conjunction with 2.5 and 2.6, it follows that whenever defined, all integer powers of perhermitian matrices are perhermitian, and integer powers of skew-perhermitian matrices are either perhermitian (even powers) or skew-perhermitian (odd powers).

In general, perhermitian [skew-perhermitian] matrices are not normal. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$] is normal, however, then we have $AA^* \in \mathscr{PH}_n$. Since the Moore–Penrose inverse can be written as $A^+ = A^* p(AA^*)$ for some polynomial p with real coefficients (cf. p. 526 of [6]), we obtain the following.

2.8. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$] is normal, then $A^+ \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$].

2.9. If $A \in \mathscr{PH}_n$, then the determinant of A, det A, is real. If $A \in \mathscr{PH}_n^-$, then det A is real [pure imaginary] if n is even [odd].

2.10. If $A \in \mathscr{PH}_n$, then the adjoint of A, adj $A \in \mathscr{PH}_n$. If $A \in \mathscr{PH}_n^-$, then adj $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$] if n is odd [even].

2.11. If $A \in \mathscr{M}_n$, then there exist unique $P, Q \in \mathscr{PH}_n$ such that $A = P + iQ$. For this result, $P = (1/2)(A + JA^*J)$ and $Q = (1/2i)(A - JA^*J)$. Note that this parallels the Toeplitz (Cartesian) decomposition $A = H + iK$ with unique $H, K \in \mathscr{H}_n$ and the decomposition 2.11 of [13].

Our next result relates principal submatrices of a perhermitian or skew-perhermitian matrix with principal submatrices of its conjugate transpose. Note that $A[p_1, \cdots, p_s | p_1, \cdots, p_s]$ denotes the principal submatrix of A that retains both the rows and columns indexed by $p_1, \cdots, p_s$.

2.12. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$], then for $s = 1, \cdots, n$, we have

$$A[p_1, \cdots, p_s | p_1, \cdots, p_s] = JA^*[n-p_s+1, \cdots, n-p_1+1 | n-p_s+1, \cdots, n-p_1+1]J$$

$$[= -JA^*[n-p_s+1, \cdots, n-p_1+1 | n-p_s+1, \cdots, n-p_1+1]J].$$

Taking determinants, we immediately get relationships between the corresponding minors. In particular, the sum of all principal minors of size s for $A \in \mathscr{PH}_n$ must be real.

3. Spectral and perspectral results. Since the coefficients of the characteristic polynomial for a matrix are sums of principal minors of the matrix multiplied by ± 1 (cf. p. 157 of [16]), result 2.12 yields the following.

3.1. If $A \in \mathscr{PH}_n$, then the characteristic polynomial of A has all real coefficients.

3.2. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$] has an eigenvalue λ of algebraic multiplicity k, then A must also have $\bar{\lambda}$ [$-\bar{\lambda}$] as an eigenvalue of algebraic multiplicity k.

3.3. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$], then A is similar to a matrix with all elements real [pure imaginary].

This follows from Carlson's Theorem 2 of [3].

3.4. If $A \in \mathscr{PH}_n$, then all the elementary symmetric functions of A are real.

3.5. If $A \in \mathscr{PH}_n$ is nonsingular and $\pi(AJ) = 0$, then

$$\pi(A) = \begin{cases} \dfrac{n}{2} & \text{if } n \text{ is even} \\ \dfrac{n-1}{2} & \text{if } n \text{ is odd.} \end{cases}$$

This follows from Johnson's Corollary 3 of [15]; it can alternately be derived from result 3.17. (Note that $\pi(X)$ denotes the number of eigenvalues of X that have positive real part.)

While many spectral results for $\mathscr{PH}_n$ are analogous to those of $\mathscr{H}_n$, others are not. For example, the spectrum of any hermitian matrix is a subset of the real line, but there is no proper subset of the complex plane that contains the spectrum of all perhermitian matrices. Indeed, given any number $z \in \mathbb{C}$, the matrix diag $(z, \bar{z})$ is perhermitian with z as an eigenvalue. Clearly, a similar construction gives matrix examples of any order greater than 2. Also, while all hermitian matrices are normal, perhermitian matrices may not even be diagonalizable (e.g., $E_{12} \in \mathscr{M}_2$).

This leads us to seek an alternate "perspectral" theory. Its construction begins by defining a *pereigenvalue* λ of a matrix $A \in \mathscr{M}_n$ to be a zero of $\det(\lambda J - A)$. We shall refer to $\lambda J - A$, $\det(\lambda J - A)$, and $\det(\lambda J - A) = 0$, as the *percharacteristic matrix*, *polynomial*, and *equation* of A, respectively, and denote the set of all pereigenvalues of A as $\sigma_p(A)$.

If λ is a pereigenvalue of A, we define a corresponding *pereigenvector* to be a nonzero $x \in \mathbb{C}^n$ for which $(\lambda J - A)x = 0$. We note that this is equivalent to $Ax = \lambda Jx$, yielding a specialization of the generalized eigenvalue problem $Ax = \lambda Bx$ (cf. [19] and pp. 251–265 of [10]). Also, since $J^2 = I$, we have $Ax = \lambda Jx$ if and only if $JAx = \lambda x$, so that (λ, x) is a pereigenvalue, pereigenvector pair of A if and only if it is also an eigenvalue, eigenvector pair of JA. This immediately yields the following results.

3.6. If $A \in \mathscr{PH}_n$ [$\mathscr{PH}_n^-$], then $\sigma_p(A) \subset \mathbb{R}$ [$i\mathbb{R}$].

3.7. Pereigenvectors corresponding to distinct pereigenvalues of a matrix are linearly independent.

3.8. The pereigenvalues of an upper or lower pertriangular matrix are the secondary diagonal elements. (A matrix $A \in \mathscr{M}_n$ is said to be *upper* [*lower*] *pertriangular* if $a_{jk} = 0$ whenever $j + k > n + 1$ [$j + k < n + 1$].)

3.9. Every matrix $A \in \mathscr{M}_n$ satisfies the percharacteristic equation of JA, and JA satisfies the percharacteristic equation of A. (These results follow directly from the Cayley–Hamilton theorem.)

Similarity transformations $P^{-1}AP$ are an integral part of matrix spectral theory. Defining $A, B \in \mathscr{M}_n$ to be *persimilar* if and only if there exists a nonsingular $P \in \mathscr{M}_n$ for which $B = JP^{-1}JAP$ yields corresponding perspectral results. Note that A is persimilar to B via P if and only if JA is similar to JB via P.

3.10. Persimilar matrices share the same percharacteristic polynomial, and hence, the same perspectrum.

3.11. A matrix $A \in \mathcal{M}_n$ is persimilar to a perdiagonal matrix with the pereigenvalues of A on the secondary diagonal if and only if A has n linearly independent pereigenvectors. In fact, $JP^{-1}JAP = D$ with D perdiagonal if and only if the columns of P are linearly independent pereigenvectors of A.

3.12. If $A \in \mathcal{PH}_n$ [$\mathcal{PH}_n^-$], then A is persimilar to a real [pure imaginary] perdiagonal matrix.

We define a *perjordan block* to be a matrix of the form JB, where B is a Jordan block. A *perjordan form* for a matrix is then a block perdiagonal matrix with each secondary diagonal block being a perjordan block.

3.13. If $A \in \mathcal{M}_n$, then A is persimilar to a perjordan form with the pereigenvalues of A of the secondary diagonal. Note that in fact, A is persimilar to J times the Jordan form for JA.

Since pereigenvectors of A have their elements reversed when multiplied by A (i.e., $A(x_1 \cdots x_n)^{\mathrm{tr}} = \lambda(x_n \cdots x_1)^{\mathrm{tr}}$), it is natural to extend the concept of quadratic forms as follows. Given $A \in \mathcal{M}_n$, the function $p_A(x) = (\overline{x_n} \cdots \overline{x_1})A(x_1 \cdots x_n)^{\mathrm{tr}}$ is said to be a *perquadratic form* in $\{x_1, \cdots, x_n\}$. This is easily seen to be equivalent to $p_A(x) = x^*JAx$; hence, the perquadratic form for A is just the quadratic form for JA.

3.14. The perquadratic form $p_A(x)$ is real for all $x \in \mathbb{C}^n$ if and only if $A \in \mathcal{PH}_n$.

A perhermitian matrix A is said to be *positive perdefinite* [*positive persemidefinite*] if and only if the generated perquadratic form $p_A(x)$ is positive [nonnegative] for all nonzero $x \in \mathbb{C}^n$. We also define $\det A[n - p_s + 1, \cdots, n - p_1 + 1 \mid p_1, \cdots, p_s]$ to be a *perprincipal minor* of A.

3.15. A matrix $A \in \mathcal{PH}_n$ is positive perdefinite [positive persemidefinite] if and only if all pereigenvalues of A are positive [nonnegative].

3.16. A matrix $A \in \mathcal{PH}_n$ is positive perdefinite [positive persemidefinite] if and only if all perprincipal minors of A are positive [nonnegative].

3.17. If $A \in \mathcal{PH}_n$ is positive perdefinite, then A is diagonalizable with all real eigenvalues and the inertia of A (i.e., the integer triple indicating the number of eigenvalues with positive, negative, and zero real part) is given by

$$\operatorname{In} A = \begin{cases} \left(\frac{n}{2}, \frac{n}{2}, 0\right) & \text{if } n \text{ is even} \\ \left(\frac{n+1}{2}, \frac{n-1}{2}, 0\right) & \text{if } n \text{ is odd.} \end{cases}$$

This follows from Carlson's Corollary 3 of [3] and the observation that A is positive perdefinite if and only if JA is positive definite.

Finally, we note that with the natural definitions, results may be obtained for perunitary and pernormal matrices that are analogous to those for unitary and normal matrices.

4. Perhermitian-preserving linear transformations. We now address the problem of characterizing perhermitian-preserving linear transformations; i.e., linear transformations on $\mathcal{M}_n$ that leave $\mathcal{PH}_n$ invariant. As in [20], if $\mathcal{T}$ is a linear transformation on $\mathcal{M}_n$, then we let $\langle \mathcal{T} \rangle \in \mathcal{M}_{n^2}$ be the matrix representation of $\mathcal{T}$ with respect to the basis of unit matrices $\{E_{ij}\}_{i,j=1,\cdots,n} \subset \mathcal{M}_n$ ordered antilexicographically; i.e., with respect to the order defined by $(i, j) < (r, s)$ if and only if $j < s$ or ($j = s$ and $i < r$). Intuitively this order may be thought of as transforming a matrix $A \in \mathcal{M}_n$ into $\operatorname{vec} A \in \mathbb{C}^{n^2}$ by stacking the columns of A into one big column vector (cf. [14] and [16]). We then have $\operatorname{vec} \mathcal{T}(A) = \langle \mathcal{T} \rangle \operatorname{vec} A$.

It is also useful to write $T \in \mathcal{M}_{n^2}$ in the block form $T = (T_{ij}) \in \mathcal{M}_n(\mathcal{M}_n)$, where $T_{ij} = (t_{rs}^{ij}) \in \mathcal{M}_n$ $(i, j, r, s = 1, \cdots, n)$. We note that the bijections Γ and Ψ on

$\mathscr{M}_n(\mathscr{M}_n)$ used in [20], [1], and [18] can be represented as $\Gamma(T)^{ij}_{rs} = t^{ir}_{js}$ and $\Psi(T)^{ij}_{rs} = t^{sj}_{ri}$ (cf. p. 4 of [20]). Intuitively, Γ rearranges the n^2 rows of T into $n \times n$ blocks ordered lexicographically, and Ψ rearranges the n^2 columns into $n \times n$ blocks antilexicographically.

Hermitian-preserving linear transformations have been characterized in the literature as transformations that can be represented in the form

$$\mathscr{T}_{\mathscr{A},D}(H) = \sum_{i,j=1}^{s} d_{ij} A_i H A_j^*$$

for some $A_1, \cdots, A_s \in \mathscr{M}_n$ and $D = (d_{ij}) \in \mathscr{M}_s$, where D is (1) hermitian, (2) diagonal with real entries, or (3) diagonal with each diagonal entry being 1, -1, or 0 (cf. [12] and [20]). Similarly, the completely positive linear transformations have been characterized as transformations of the same general form where D is (1) positive semidefinite, (2) diagonal with nonnegative entries, or (3) diagonal with each diagonal entry being 1 (cf. [4] and [20]). It is natural to hope then, that the perhermitian-preserving linear transformations could be characterized as transformations of the same general form where D is (1) perhermitian, (2) perdiagonal with real entries, or (3) perdiagonal with each secondary diagonal entry being 1, -1, or 0. This hope, along with other characterizations analogous to Theorems 1 and 2 of [20] and Theorem 4.1 of [13], is realized in the following theorem.

THEOREM 4.1. *Let $\mathscr{T}$ be a linear transformation on $\mathscr{M}_n$. Then the following are equivalent*:

(1) *$\mathscr{T}$ is perhermitian-preserving.*

(2) *$\mathscr{T}$ is skew-perhermitian-preserving.*

(3) *There exist $A_1, \cdots, A_t \in \mathscr{M}_n$ with $JA_kJ = A_{t-k+1}$ $(k = 1, \cdots, t)$ and $G = (g_{ij}) \in \mathscr{PH}_t$ for which*

$$\mathscr{T}(X) = \sum_{i,j=1}^{t} g_{ij} A_i X A_j^*.$$

(4) *There exist $A_1, \cdots, A_t \in \mathscr{M}_n$ with $JA_kJ = A_{t-k+1}$ $(k = 1, \cdots, t)$ and $\gamma_1, \cdots, \gamma_t \in \mathbb{R}$ for which*

$$\mathscr{T}(X) = \sum_{i=1}^{t} \gamma_i A_{t-i+1} X A_i^*.$$

(5) *There exist $A_1, \cdots, A_t \in \mathscr{M}_n$ with $JA_kJ = A_{t-k+1}$ $(k = 1, \cdots, t)$ and $\gamma_1, \cdots, \gamma_t \in \{-1, 1, 0\}$ for which*

$$\mathscr{T}(X) = \sum_{i=1}^{t} \gamma_i A_{t-i+1} X A_i^*.$$

(6) *$t^{ij}_{rs} = \bar{t}^{\,n-r+1,n-s+1}_{n-i+1,n-j+1}$ $(i, j, r, s = 1, \cdots, n)$ where $\langle \mathscr{T} \rangle = ((t^{ij}_{rs}))$.*

(7) *$\Gamma(\langle \mathscr{T} \rangle)$ is perhermitian.*

(8) *$\Psi(\langle \mathscr{T} \rangle)$ is perhermitian.*

(9) *$\Omega(\langle \mathscr{T} \rangle)$ $(= \Gamma(\langle \mathscr{T} \rangle^{\mathrm{tr}}))$ is perhermitian.*

(10) *$\Theta(\langle \mathscr{T} \rangle)$ $(= \Psi(\langle \mathscr{T} \rangle^{\mathrm{tr}}))$ is perhermitian.*

(11) *The block matrix $(\mathscr{T}(E_{ij}))$ is perhermitian.*

(12) *$\mathscr{T}^*$ is perhermitian-preserving.*

Proof. Result 2.1(v) immediately gives (1) $\Leftrightarrow$ (2). For (1) $\Rightarrow$ (3), suppose that $\mathscr{T}$ is perhermitian-preserving and define the transformation $\mathscr{J}: \mathscr{M}_n \to \mathscr{M}_n$ by $\mathscr{J}(X) = JX$.

Since X is perhermitian if and only if JX is hermitian (cf. Result 2.1 (iii)), we have that $\mathscr{T}$ must equal the composite $\mathscr{J}\circ\mathscr{T}_{\mathscr{A},D}\circ\mathscr{J}$ for some hermitian-preserving transformation $\mathscr{T}_{\mathscr{A},D}$ where $A_1, \cdots, A_s \in \mathscr{M}_n$ and $D \in \mathscr{H}_s$. If we now let $t = 2s$, define $A_k = JA_{t-k+1}J$ $(k = s+1, \cdots, t)$ and define $G \in \mathscr{M}_t$ to have the 2×2 block structure

$$G = \begin{bmatrix} 0 & 0 \\ JD & 0 \end{bmatrix},$$

then we have that $G \in \mathscr{PH}_t$, $g_{t-k+1,j} = d_{kj}$ $(k = 1, \cdots, s)$ and

$$\begin{aligned} \mathscr{T}(X) &= J \sum_{i,j=1}^{s} d_{ij} A_i J X A_j^* \\ &= \sum_{i,j=1}^{s} g_{t-i+1,j} J A_i J X A_j^* \\ &= \sum_{i=s+1}^{t} \sum_{j=1}^{s} g_{ij} J A_{t-i+1} J X A_j^* \\ &= \sum_{i,j=1}^{t} g_{ij} A_i X A_j^* \end{aligned}$$

where the last equality makes use of the zero blocks in G and the fact that $A_k = JA_{t-k+1}J$ $(k = s+1, \cdots, t)$.

Conversely ((3) $\Rightarrow$ (1)), suppose that $\mathscr{T}$ has the form indicated in (3) and that P is perhermitian. We then have

$$\begin{aligned} J\left(\sum_{i,j=1}^{t} g_{ij} A_i P A_j^*\right)^* J &= \sum_{i,j=1}^{t} \bar{g}_{ij} (JA_jJ)(JP^*J)(JA_iJ)^* \\ &= \sum_{i,j=1}^{t} g_{t-j+1,t-i+1} A_{t-j+1} P A_{t-i+1}^* \\ &= \sum_{i,j=1}^{t} g_{ij} A_i P A_j^* \end{aligned}$$

so that $\mathscr{T}(P)$ is perhermitian whenever P is; thus, $\mathscr{T}$ is perhermitian-preserving.

The equivalences (1) $\Leftrightarrow$ (4) and (1) $\Leftrightarrow$ (5) can be established in a similar fashion.

Using a proof technique analogous to Theorem 1 of [12] (viz., by computing $\mathscr{T}(B)$ for each of the basis elements in Result 2.3, and forcing these to be perhermitian), we get (1) $\Leftrightarrow$ (6).

Noting that (t_{rs}^{ij}) is perhermitian if and only if $t_{rs}^{ij} = \bar{t}_{n-s+1,n-r+1}^{n-j+1,n-i+1}$, we obtain (6) $\Leftrightarrow$ (7) $\Leftrightarrow$ (8). By Lemma 1 of [20] and Theorem 1 of [18], we have that $\Gamma(T^{\mathrm{tr}}) = \Psi(T)^{\mathrm{tr}} = \Omega(T)$ and $\Psi(T^{\mathrm{tr}}) = \Gamma(T)^{\mathrm{tr}} = \Theta(T)$ for all $T \in \mathscr{M}_n(\mathscr{M}_n)$. Therefore, since T is perhermitian if and only if T^{tr} is perhermitian, we have (7) $\Leftrightarrow$ (10) and (8) $\Leftrightarrow$ (9). Also, by Lemma 2 of [20] we have that the block matrix $(\mathscr{T}(E_{ij})) = \Psi(\langle \mathscr{T} \rangle)$, which gives (8) $\Leftrightarrow$ (11).

Finally, since $\{E_{ij}\}$ is an orthonormal basis for $\mathscr{M}_n$, we have that the matrix representation of the Hilbert adjoint of $\mathscr{T}$ is $\langle \mathscr{T}^* \rangle = \langle \mathscr{T} \rangle^*$, thus yielding (1) $\Leftrightarrow$ (12). □

We note that $\langle \mathscr{T} \rangle$ being perhermitian is independent of $\mathscr{T}$ being perhermitian-preserving. While this is analogous to the hermitian-preserving transformations, it is in

stark contrast to the centrohermitian-preserving transformations (cf. Theorem 4.1 of [13]).

Acknowledgment. The authors wish to thank Professor E. E. Underwood, Utah State University, for suggesting the topics of persymmetric (perhermitian) and centrosymmetric (centrohermitian) matrices.

REFERENCES

[1] G. P. BARKER, R. D. HILL, AND R. D. HAERTEL, *On the completely positive and positive semidefinite preserving cones*, Linear Algebra Appl., 56 (1984), pp. 221–229.

[2] A. CANTONI AND P. BUTLER, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl., 13 (1976), pp. 275–288.

[3] D. H. CARLSON, *On real eigenvalues of complex matrices*, Pacific J. Math., 15 (1965), pp. 1119–1129.

[4] M. CHOI, *Completely positive linear maps on complex matrices*, Linear Algebra Appl., 10 (1975), pp. 285–290.

[5] G. CYBENKO, *On the eigenstructure of Toeplitz matrices*, IEEE Trans. Acoust. Speech Signal Process, 32 (1984), pp. 918–921.

[6] H. P. DECELL, JR., *An application of the Cayley–Hamilton theorem to generalized matrix inversion*, SIAM Rev., 7 (1965), pp. 526–528.

[7] Y. GENIN, *A survey of the eigenstructure properties of finite hermitian Toeplitz matrices*, Integral Equations Operator Theory, 10 (1987), pp. 621–639.

[8] M. J. GOLDSTEIN, *Reduction of the eigenproblem for hermitian persymmetric matrices*, Math. Comp., 28 (1974), pp. 237–238.

[9] ———, *Reduction of the pseudoinverse of a hermitian persymmetric matrix*, Math. Comp., 28 (1974), pp. 715–717.

[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[11] J. GRCAR AND A. SAMEH, *On certain parallel Toeplitz linear-system solvers*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 238–256.

[12] R. D. HILL, *Linear transformations which preserve hermitian matrices*, Linear Algebra Appl., 6 (1973), pp. 257–262.

[13] R. D. HILL, R. G. BATES, AND S. R. WATERS, *On centrohermitian matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 128–133.

[14] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, to appear.

[15] C. R. JOHNSON, *The inertia of a product of two hermitian matrices*, J. Math. Anal. Appl., 57 (1977), pp. 85–90.

[16] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, 2nd ed., Academic Press, Orlando, FL, 1985.

[17] A. LEE, *On S-symmetric, S-skewsymmetric, and S-orthogonal matrices*, Period Math. Hungar., 7 (1976), pp. 71–76.

[18] C. J. OXENRIDER AND R. D. HILL, *On the matrix reorderings* Γ *and* Ψ, Linear Algebra Appl., 69 (1985), pp. 205–212.

[19] G. PETERS AND J. H. WILKINSON, $Ax = \lambda Bx$ *and the generalized eigenproblem*, Siam J. Numer. Anal., 7 (1970), pp. 479–492.

[20] J. A. POLUIKIS AND R. D. HILL, *Completely positive and hermitian-preserving linear transformations*, Linear Algebra Appl., 35 (1981), pp. 1–10.

[21] W. D. RAY, *The inverse of a finite Toeplitz matrix*, Technometrics, 12 (1970), pp. 153–156.

[22] P. A. ROEBUCK AND S. BARNETT, *A survey of Toeplitz and related matrices*, Internat. J. Systems Sci., 9 (1978), pp. 921–934.

[23] W. R. TRENCH, *Numerical-solution of the eigenvalue problem for symmetric rationally generated Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 291–303.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 180–199, April 1990

A MATRIX EQUATION APPROACH TO THE DESIGN OF LOW-ORDER REGULATORS*

L. H. KEEL† AND S. P. BHATTACHARYYA‡

Abstract. This paper presents an algorithm for stabilizing a linear multivariable system with a controller of fixed dynamic order. This is an output feedback stabilization problem. An algorithm attempts to solve this via a sequence of approximate pole assignment problems. The approximation is driven by the optimization of a performance index consisting of a weighted sum of the condition number of the closed-loop eigenvectors and the norm of the difference between the computed and actual controls.

The algorithm can be used for generating low-order solutions to the regulator problem. The problem treated here is useful in design problems that involve parameter optimization and is also important in practical situations where stabilization is to be accomplished with a fixed number of available parameters.

Key words. regulator, eigenstructure, output feedback, state feedback, stability

AMS(MOS) subject classification. 93D15

1. Introduction.

The regulator or feedback stabilization problem is the basic problem that control theory attempts to solve. Many design procedures can only be initiated after a nominal stabilizing controller has been found. However, except for very special cases, there are no direct procedures available to solve this problem when the controller order is fixed. Existing solutions to the regulator problem can only generate controllers that are of high enough order that arbitrary pole placement becomes possible. This includes the LQG theory [1], observed state feedback [2], and arbitrary pole placement approaches [3], [4]. Controllers that are robust with respect to unstructured perturbations evidently suffer from the same difficulty of high order (see examples given in [5]). We also mention that adaptive control theory is notorious for producing high-order solutions.

It is certainly essential in practice, to have *low*-order solutions to the stabilization problem. This requirement arises because the controller must eventually carry out several functions such as tracking, disturbance rejection, desensitization against parameter variations, provide good transient response, small steady-state error, prevent various signals from saturating, etc., in addition to the basic task of stabilization. Many of these requirements are in conflict with each other in ways that cannot be handled analytically and the only recourse left to the designer is to iteratively redesign the controller using ad hoc methods and graphical displays until a satisfactory solution is obtained. This redesign must be carried out in the parameter space of the stabilizing controller. If the basic stabilizing controller order is unnecessarily high this parameter space is also of high dimension and the subsequent design process can become unwieldy. From this prospective, the high order of controllers produced by "modern" control theory is one of the severest limitations of this theory.

We attempt to alleviate this problem by presenting, in this paper, a direct algorithm in the state space domain, for designing low-order stabilizing controllers. This algorithm first attempts to stabilize the closed-loop system with a fixed-order controller. This corresponds to an extended output feedback stabilization problem for which no analytical

* Received by the editors March 21, 1988; accepted for publication (in revised form) March 2, 1989. This research was partially supported by National Science Foundation grant ECS-8309792 and National Aeronautics and Aerospace Administration grant NAG-1-863 (Bitnet%Bhatt@TAMVXEE).

† Department of Electrical Engineering and Center of Excellence in Information Systems Engineering and Management, Tennessee State University, Nashville, Tennessee 37203.

‡ Department of Electrical Engineering, Texas A&M University, College Station, Texas 77843.

solution is available. We attempt to solve this iteratively. At each iteration a state feedback matrix assigning a prescribed set of eigenvalues is found and this matrix is approximated by output feedback. This is done successively by readjusting the desired closed-loop pole locations in the left half of the complex plane to minimize a performance index that measures the deviation of the actual eigenvalues from the desired ones. A low-order solution is found by sequentially increasing the controller order until stabilization is achieved.

The algorithm that is given depends on the parameterization of the state feedback pole assignment problem derived in [6]. This is briefly described in the next section. In § 3, the fixed-order output feedback stabilization problem is formulated as an optimization problem and § 4 describes how the performance index can be decreased by increasing the controller order. Examples are given in § 5 and some of the gradient evaluations of § 4 are derived in the Appendix.

2. The Sylvester equation formulation. An algorithm was introduced in [6] for solving the pole assignment problem using state feedback. This algorithm consists of solving for X and then for F

$$AX - X\tilde{A} = -BG, \tag{2.1}$$

$$FX = G \tag{2.2}$$

for given $(A, B, \tilde{A})$ with an arbitrary choice of G. In (2.1) and (2.2) A, X, and $\tilde{A}$ are $n \times n$ matrices. From a result in [7] the solution X of (2.1) generically has full rank if (A, B) is controllable and $(G, \tilde{A})$ is observable. Let $\lambda_i(T)$ denote the ith eigenvalue of T and $\lambda(T)$, the spectrum or eigenvalue set of T. It follows that if X has full rank the solution F has the property:

$$\lambda(A + BF) = \lambda(\tilde{A}). \tag{2.3}$$

The advantages of this algorithm are:

(a) The algebraic variety $F(\Lambda)$ of matrices F that assign a prescribed set of eigenvalues Λ can be obtained by setting $\Lambda = \lambda(\tilde{A})$ for a fixed $\tilde{A}$, and letting the *free* parameter G run through the set of all possible real values.

(b) Efficient numerical procedures [8] are available for the solution of Sylvester's equation (2.1).

Based on this parameterization of $F(\Lambda)$, algorithms were given [9] and [10] for optimizing the conditioning of the closed-loop eigenvectors and [11] for minimizing the norm of the state feedback matrix F. Here, we extend these results by considering measurement rather than state feedback and by treating the problem of stabilization rather than arbitrary pole placement.

3. Output feedback controllers. Consider the linear time-invariant plant S cascaded with the pth order feedback compensator C:

$$S: \dot{x} = Ax + Bu, \qquad y_m = Cx, \tag{3.1}$$

$$C: \dot{x}_c = A_c x_c + B_c y_m, \qquad u = C_c x_c + D_c y_m. \tag{3.2}$$

The closed-loop system is

$$\begin{pmatrix} \dot{x} \\ \dot{x}_c \end{pmatrix} = \begin{pmatrix} A + BD_cC & BC_c \\ B_cC & A_c \end{pmatrix} \begin{pmatrix} x \\ x_c \end{pmatrix} \tag{3.3}$$

or

$$\underbrace{\begin{pmatrix}\dot{x}\\ x_c\end{pmatrix}}_{\dot{x}_p} = \left\{ \underbrace{\begin{pmatrix}A & 0\\ 0 & 0_p\end{pmatrix}}_{A_p} + \underbrace{\begin{pmatrix}B & 0\\ 0 & I_p\end{pmatrix}}_{B_p} \underbrace{\begin{pmatrix}D_c & C_c\\ B_c & A_c\end{pmatrix}}_{K_p} \underbrace{\begin{pmatrix}C & 0\\ 0 & I_p\end{pmatrix}}_{C_p} \right\} \underbrace{\begin{pmatrix}x\\ x_c\end{pmatrix}}_{x_p} \tag{3.4}$$

and the transfer function of the pth-order compensator is

$$C(s) := C_c(sI - A_c)^{-1}B_c + D_c. \tag{3.5}$$

Formula (3.4) shows that any fixed-order compensator design problem is equivalent to a static output feedback problem. In particular, the problem of stabilization with a fixed-order controller p is equivalent to that of stabilizing $A_p + B_pK_pC_p$ by choice of K_p. The general solution of this problem is unknown. The best available special results are those of Brasch and Pearson [3] and Kimura [4] that deal, respectively, with arbitrary eigenvalue assignment and "almost" arbitrary eigenvalue assignment.

Let Λ denote a symmetric set of $n + p$ complex numbers (i.e., complex numbers occur in complex conjugate pairs) and let

$$\underline{K}_p(\Lambda) := \{K_p \mid K_p \in R^{(m+p)\times(r+p)}, \lambda(A_p + B_pK_pC_p) \in \Lambda\} \tag{3.6}$$

where $A_p \in R^{(n+p)\times(n+p)}$, $B_p \in R^{(n+p)\times(m+p)}$, and $C_p \in R^{(r+p)\times(n+p)}$ are as in (3.4).

The result of Brasch and Pearson [3] states that if (A, B, C) is controllable and observable with controllability index ν_c and observability index ν_o, and $p \geqq \min\{\nu_c, \nu_o\}$, then $K_p(\Lambda) \neq \varnothing$ for every choice of Λ. The result of Kimura [4] states that if $p \geqq n - m - r + 1$ then $\lambda(A_p + B_pK_pC_p)$ can be made arbitrarily close to any set Λ of $n + p$ symmetric complex numbers.

The lower bound on the order of a stabilizing controller established by the above results is in general too conservative. This stems from the fact that both results essentially require arbitrary pole placement. In fact for specific choices of Λ, $K_p(\Lambda)$ will "almost always" be empty unless p, the compensator order, is high. To lower the compensator order we therefore relax the specification of Λ in (3.6) to a simply connected region $\Omega \subset C^-$ and consider the family

$$\underline{K}_p(\Omega) = \{K_p \mid K_p \in R^{(m+p)\times(r+p)}, \lambda(A_p + B_pK_pC_p) \subset \Omega \subset C^-\}. \tag{3.7}$$

It is reasonable to expect that $\underline{K}_p(\Omega)$ will in general be nonempty for values of p much less than the lower bounds given by the results of Brasch and Pearson or Kimura and numerical examples support this.

The effective characterization of the family $\underline{K}_p(\Omega)$ is an unsolved open problem. Our approach to this problem will be to consider the state feedback family:

$$\underline{F}_p(\Omega) = \{F_p \mid F_p \in R^{(m+p)\times(n+p)}, \lambda(A_p + B_pF_p) \subset \Omega \subset C^-\} \tag{3.8}$$

and determine an $F_p \in \underline{F}_p(\Omega)$ and then find K_p such that $\|F_p - K_pC_p\|$ is small in the hope that such a $K_p \in \underline{K}_p(\Omega)$. The advantage of this approach is that the family $\underline{F}_p(\Omega)$ can be characterized conveniently as shown later. For the remainder of this section we drop the subscript p for convenience.

In general, even if $\|F - KC\|$ is small it is not true that $\lambda(A + BF)$ and $\lambda(A + BKC)$ are close. The latter can be achieved by making the eigenstructure of $A + BF$ as orthonormal as possible. Let $\sigma_{\max}(T)$ and $\sigma_{\min}(T)$ denote the largest and smallest singular values of T. It is well known [8], [12] that the perturbation of the eigenvalues of the diagonalizable matrix $(A + BF)$ for changes in the entries is small if the condition number $k(X) := \|X\|_2\|X^{-1}\|_2$ of the eigenvector matrix X is small. Let

$F - KC := T$ so that $A + BKC = A + BF - BT$. Then using the formula in [12] we have

$$\begin{aligned} |\lambda_i(A+BKC)-\lambda_j(A+BF)| &\leqq \|BT\|_2 k(X) \\ &\leqq \|B\|_2\|T\|_2 k(X) \\ &\leqq \|B\|_2\|F-KC\|_F k(X), \end{aligned} \tag{3.9}$$

which shows that control over the eigenvalue locations of $A + BKC$ can be obtained only if both $\|F - KC\|$ and $k(X)$ are kept small. One way of doing this is to minimize

$$\begin{aligned} J &= \alpha_1 k(X) + \alpha_2 \|F-KC\|_F^2 \\ &= \alpha_1 \frac{\sigma_{\max}(X)}{\sigma_{\min}(X)} + \alpha_2 \operatorname{trace}\{(F-KC)^T(F-KC)\} \end{aligned} \tag{3.10}$$

by letting $\lambda(A + BF)$ range over the region $\Omega \subset C^-$. Similarly, by letting $A + F_D C = A + BK_D C$ a dual problem can be formulated as

$$J_D = \beta_1 \frac{\sigma_{\max}(X_D)}{\sigma_{\min}(X_D)} + \beta_2 \operatorname{trace}\{(F_D - BK_D)^T(F_D - BK_D)\}. \tag{3.11}$$

The idea of improving the conditioning of the eigenstructure and of minimizing the norm of $F - KC$ was first introduced in Keel and Bhattacharyya [13], [14]. Here an improved version of this algorithm is presented. In particular, we convert the constrained optimization problem to an unconstrained problem and extend the class of regions $\Omega \subset C^-$ to more general and useful regions. These details are given next.

4. Stabilization algorithm. In the Sylvester equation approach described in § 2,

$$AX - X\tilde{A} = -BG, \tag{4.1}$$

$$FX = G \tag{4.2}$$

and let $\lambda(\tilde{A}) \subset \Omega \subset C^-$. Under the assumption $\lambda(A) \cap \lambda(\tilde{A}) = \varnothing$ and (A, B) controllable, $(G, \tilde{A})$ observable, the unique solution X will "almost surely" be nonsingular by deSouza and Bhattacharyya [7] and then $\lambda(A + BF) = \lambda(\tilde{A})$ with $F = GX^{-1}$. By letting $\lambda(\tilde{A})$ range over Ω this algorithm generates the family of $\underline{F}(\Omega)$, by letting G be a free parameter run through *all* possible values this formula generates the family $\underline{F}(\Omega)$ defined in (3.8).

If $\tilde{A}$ is a complex diagonal matrix in (4.1), it is clear that X in (4.1) is the corresponding complex eigenvector matrix. However we want to treat these matrices as real for computational convenience. The following Lemma 4.5 shows that $\tilde{A}$ can be taken as a real matrix without loss of generality. Before we state Lemma 4.3 it is necessary to introduce some facts.

DEFINITION 4.1. A real square matrix A is called a pseudodiagonal matrix if it is of the form

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ -\beta_1 & \alpha_1 & & & \\ & & \alpha_2 & \beta_2 & \\ & & -\beta_2 & \alpha_2 & \\ & & & & \alpha_3 \\ & & & & & \ddots \end{pmatrix} \tag{4.3}$$

with α_i, β_i real.

DEFINITION 4.2. A complex square matrix is called normal if $A^*A = AA^*$.

LEMMA 4.3 [15]. *A complex square matrix is unitary similar to a diagonal complex matrix if and only if it is normal.*

LEMMA 4.4. *Any real pseudodiagonal matrix is normal.*

Proof. Taking the ith block from (4.3) such as

$$A_i = \begin{pmatrix} \alpha_i & \beta_i \\ -\beta_i & \alpha_i \end{pmatrix}, \tag{4.4}$$

we have

$$A_i A_i^* = \begin{pmatrix} \alpha_i^2 + \beta_i^2 & 0 \\ 0 & \alpha_i^2 + \beta_i^2 \end{pmatrix} = A_i^* A_i. \tag{4.5}$$

Thus, each block is normal. Now let

$$A = \operatorname{diag}(A_1 \quad A_2 \cdots \cdots A_n), \tag{4.6}$$

$$AA^* = \operatorname{diag}(A_1 A_1^* \quad A_2 A_2^* \cdots A_n A_n^*), \tag{4.7}$$

$$A^*A = \operatorname{diag}(A_1^* A_1 \quad A_2^* A_2 \cdots A_n^* A_n). \tag{4.8}$$

Since $AA^* = A^*A$, the statement is true.

LEMMA 4.5. *Let* $(A + BF)X = X\tilde{A}$ *and* $(A + BF)Y = Y\hat{A}$, *where*

(1) A, B, $\tilde{A}$, X *and* F *are real matrices with appropriate dimensions.*

(2) $\tilde{A}$ *is real pseudodiagonal,* $\hat{A}$ *is complex diagonal, and*

(3) X *and* Y *are nonsingular. Then,*

$$k(X) = k(Y). \tag{4.9}$$

Proof. From Definitions 4.1 and 4.2, $\tilde{A}$ is known to be normal and unitary similar to the complex diagonal matrix $\hat{A}$. Thus

$$\tilde{A} = U\hat{A}U^*. \tag{4.10}$$

Write

$$(A + BF)X = X\tilde{A} = XU\hat{A}U^* \tag{4.11}$$

so that

$$(A + BF)XU = XU\hat{A} \tag{4.12}$$

and

$$XU = Y. \tag{4.13}$$

Now,

$$YY^* = XUU^*X^* = XX^* = XX^T. \tag{4.14}$$

□

From this lemma, minimizing $\sigma_{\max}(X)/\sigma_{\min}(X)$ in (3.10) is equivalent to minimizing $\sigma_{\max}(Y)/\sigma_{\min}(Y)$. Therefore we can henceforth take $\tilde{A}$ as a real pseudodiagonal matrix without loss of generality. In fact, the condition numbers of X and Y are equal, i.e., $k(X) = k(XU) = k(Y)$. In order to use a gradient-based algorithm the closed-form expression of the gradient of the performance index (3.10) with respect to the variables G, K and the variable elements of $\tilde{A}$ denoted $\tilde{a}_i$ is evaluated. The details of this derivation are given in the Appendix.

THEOREM 4.6. *Given the performance index J in* (3.10), *and constraints* (4.1) *and* (4.2), *the gradients of J with respect to the independent variables G, K, and* $\tilde{A}$ are as follows:

(a)

$$\frac{\partial J}{\partial G} = 2\{\alpha_2(F-KC)X^{-T} + B^T U^T\} \tag{4.15}$$

where U satisfies

$$\tilde{A}U - UA = \frac{\alpha_1}{\sigma_{\min}^2(X)}\{\sigma_{\min}(X)v_a u_a^T - \sigma_{\max}(X)v_i u_i^T\} - 2\alpha_2 X^{-1}(F^T - (KC)^T)F \tag{4.16}$$

where v_a *and* u_a *are right and left singular vectors corresponding to* $\sigma_{\max}(X)$ *and* v_i *and* u_i *are for* $\sigma_{\min}(X)$, *respectively.*

(b) *Let* $\tilde{a}_i$ *denote a variable element of* $\tilde{A}$:

$$\frac{\partial J}{\partial \tilde{a}_i} = -\text{trace}\left\{UX\frac{\partial \tilde{A}}{\partial \tilde{a}_i}\right\} \tag{4.17}$$

where U satisfies (4.16).

(c)

$$\frac{\partial J}{\partial K} = -2\alpha_2(F-KC)C^T. \tag{4.18}$$

Equations (4.15)–(4.18) are used to devise a gradient algorithm that iterates on the free parameters G, K and the entries of $\tilde{A}$ to reduce J. At each iteration of the algorithm we get $\tilde{A}_i$, F_i, and K_i. Since $\lambda(\tilde{A}_i) \subset \Omega$ we have $\lambda(A + BF_i) \subset \Omega$ for each i. However, $\lambda(A + BK_iC)$ may or may not be in Ω for each i, and the algorithm is designed to make $\lambda(A + BK_iC)$ close to $\lambda(\tilde{A}_i) = \lambda(A + BF_i)$ after some iterations.

The following structure of the closed-loop eigenvalue matrix $\tilde{A}$ ensures stability without constraints during the iterations:

$$\tilde{A} = \begin{pmatrix} -\tilde{a}_1^2 & \tilde{a}_2 & & & & \\ -\tilde{a}_2 & -\tilde{a}_1^2 & & & & \\ & & -\tilde{a}_3^2 & \tilde{a}_4 & & \\ & & -\tilde{a}_4 & -\tilde{a}_3^2 & & \\ & & & & \ddots & \\ & & & & & \ddots \end{pmatrix}.$$

Note that $\tilde{a}_i$ in the matrix $\tilde{A}$ are the only nonzero parameters and furthermore the stability requirement $\lambda(\tilde{A}) \subset C^-$ can be automatically satisfied without constraints for *all* real values of $\tilde{a}_i$.

We can also parameterize $\tilde{A}$ in such a way that the desired closed-loop eigenvalue locations are automatically confined to some useful region Ω as in Figs. 4.1 and 4.2.

In choosing $\tilde{A}$, a maximal number of 2×2 blocks are included in the initial choice. As the algorithm evolves some of the off-diagonal terms may become very small. At that point we start to vary the corresponding diagonal terms independently. In the damping ratio region described in Fig. 4.2, θ is also a free parameter.

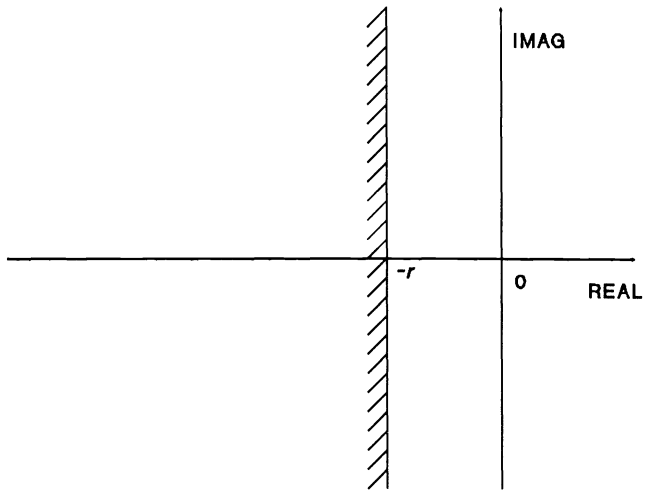


FIG. 4.1. *Marginal stability region.*

Marginal Stability Region. For this case we can simply modify the matrix $\tilde{A}$ to

$$\tilde{A} = \begin{pmatrix} -(\tilde{a}_1^2+\gamma) & \tilde{a}_2 & & & & \\ -\tilde{a}_2 & -(\tilde{a}_1^2+\gamma) & & & & \\ & & -(\tilde{a}_3^2+\gamma) & \tilde{a}_4 & & \\ & & -\tilde{a}_4 & -(\tilde{a}_3^2+\gamma) & & \\ & & & & \ddots & \\ & & & & & \ddots \\ & & & & & & \ddots \end{pmatrix}$$

with $\tilde{a}_i$ as the real variable parameters and γ is fixed. The eigenvalues of $\tilde{A}$ are all to the left of the line $\mathrm{Re}\,(s) = -\gamma$.

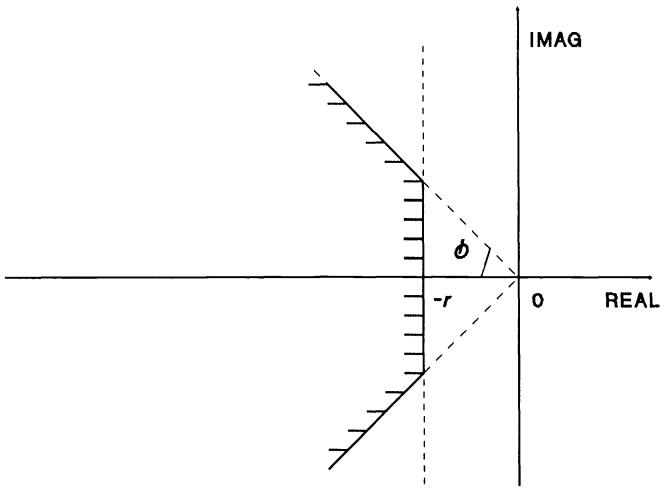


FIG. 4.2. *Damping ratio region.*

Damping Ratio Region.

$$\tilde{A} = \begin{pmatrix} -(\tilde{a}_1^2+\gamma) & (\tilde{a}_1^2+\gamma)\tan\phi\sin\theta_1 & & & \\ -(\tilde{a}_1^2+\gamma)\tan\phi\sin\theta_1 & -(\tilde{a}_1^2+\gamma) & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \ddots \end{pmatrix}.$$

Now we discuss what happens when the proposed algorithm fails to find a stabilizing controller of order i. In this case, we increase the controller order to $i+1$. It is then necessary to have a way to select the initial values of G_0, $\tilde{A}_0$, and K_0 for the controller of order $i+1$ to ensure that the performance index J keeps decreasing. The following theorem shows the way to select initial variables so that J always decreases with increasing controller order.

THEOREM 4.7. *Let J^* be the optimal performance index with optimal variables G^*, $\tilde{A}^*$, and K^* where*

$$J^* = \frac{\sigma_{\max}(X^*)}{\sigma_{\min}(X^*)} + \|F^* - K^*C\|_F^2 \tag{4.19}$$

and X^* and F^* satisfy

$$AX^* - X^*\tilde{A}^* = -BG^*, \qquad F^* = G^*(X^*)^{-1}.$$

Then for the extended system

$$A_e = \begin{pmatrix} A & 0 \\ 0 & I_i \end{pmatrix}, \quad B_e = \begin{pmatrix} B & 0 \\ 0 & I_i \end{pmatrix}, \quad C_e = \begin{pmatrix} C & 0 \\ 0 & I_i \end{pmatrix} \tag{4.20}$$

the value of its performance index J_e is equal to J^* if the set of initial variables are

$$G_e = \begin{pmatrix} G^* & 0 \\ 0 & X_3\tilde{A}_i \end{pmatrix}, \qquad K_e = \begin{pmatrix} K^* & 0 \\ 0 & X_3\tilde{A}_iX_3^{-1} \end{pmatrix} \tag{4.21}$$

where $\tilde{A}_i$ is an arbitrary pseudodiagonal matrix of an extended matrix

$$\tilde{A}_e = \begin{pmatrix} \tilde{A} & 0 \\ 0 & \tilde{A}_i \end{pmatrix}$$

and

$$X_3 = \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \sigma_i \end{pmatrix}$$

for $\sigma_1 \geqq \sigma_2 \geqq \cdots \geqq \sigma_i > 0$ with $\sigma_i \geqq \sigma_{\min}(X^*)$ and $\sigma_1 \leqq \sigma_{\max}(X^*)$.

Proof. Let the optimal values of J^* be obtained by G^* and K^*; then the extended system becomes

$$\underbrace{\begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}}_{A_e}\underbrace{\begin{pmatrix} X^* & X_1 \\ X_2 & X_3 \end{pmatrix}}_{X_e}-\underbrace{\begin{pmatrix} X^* & X_1 \\ X_2 & X_3 \end{pmatrix}}_{X_e}\underbrace{\begin{pmatrix} \tilde{A}^* & 0 \\ 0 & \tilde{A}_i \end{pmatrix}}_{\tilde{A}_e}=-\underbrace{\begin{pmatrix} B & 0 \\ 0 & I_i \end{pmatrix}}_{B_e}\underbrace{\begin{pmatrix} G^* & G_1 \\ G_2 & G_3 \end{pmatrix}}_{G_e}$$

(4.22)

$$=\begin{pmatrix} AX^*-X^*\tilde{A}^* & AX_1-X_1\tilde{A}_i \\ -X_2\tilde{A} & -X_3\tilde{A}_i \end{pmatrix}$$

$$=-\begin{pmatrix} BG^* & BG_1 \\ G_2 & G_3 \end{pmatrix}$$

if we pick $G_1 = 0$ and $G_2 = 0$, then $X_1 = 0$ and $X_2 = 0$ and $X_3\tilde{A}_i = G_3$. Here we choose

(4.23)
$$X_3=\begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_i \end{pmatrix}$$

for $\sigma_1 \geqq \sigma_2 \geqq \cdots \geqq \sigma_i > 0$ with $\sigma_i \geqq \sigma_{\min}(X^*)$ and $\sigma_1 \leqq \sigma_{\max}(X^*)$. Such a X_3 is guaranteed by the choice of $G_3 = X_3\tilde{A}_i$ and

(4.24)
$$\sigma_{\min}(X^*)=\sigma_{\min}(X_e),$$

(4.25)
$$\sigma_{\max}(X^*)=\sigma_{\max}(X_e).$$

Therefore,

(4.26)
$$\frac{\sigma_{\max}(X^*)}{\sigma_{\min}(X^*)}=\frac{\sigma_{\max}(X_e)}{\sigma_{\min}(X_e)}.$$

Now consider the term $\|F^* - K^*C\|_F^2$. Since

$$X_e=\begin{pmatrix} X^* & 0 \\ 0 & X_3 \end{pmatrix},$$

we have

(4.27)
$$X_e^{-1}=\begin{pmatrix} (X^*)^{-1} & 0 \\ 0 & X_3^{-1} \end{pmatrix}$$

where

$$X_3^{-1}=\begin{pmatrix} 1/\sigma_1 & & \\ & \ddots & \\ & & 1/\sigma_i \end{pmatrix}.$$

Now

$$F_e=G_eX_e^{-1}=\begin{pmatrix} G^* & 0 \\ 0 & X_3\tilde{A}_i \end{pmatrix}\begin{pmatrix} (X^*)^{-1} & 0 \\ 0 & X_3^{-1} \end{pmatrix}$$

(4.28)

$$=\begin{pmatrix} G^*(X^*)^{-1} & 0 \\ 0 & X_3\tilde{A}_iX_3^{-1} \end{pmatrix}$$

and let

$$K_e = \begin{pmatrix} K^* & K_1 \\ K_2 & K_3 \end{pmatrix}; \tag{4.29}$$

then

$$F_e - K_e C_e = \begin{pmatrix} G^*(X^*)^{-1} - K^*C & -K_1 \\ -K_2 C & X_3 \tilde{A}_i X_3^{-1} - K_3 \end{pmatrix}. \tag{4.30}$$

Here we choose $K_1 = 0$ and $K_2 = 0$. Also, we can choose

$$K_3 = X_3 \tilde{A}_i X_3^{-1} \tag{4.31}$$

because X_3 and $\tilde{A}_i$ are well defined. With such a K we have

$$F_e - K_e C_e = \begin{pmatrix} G^*(X^*)^{-1} - K^*C & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.32}$$

Thus,

$$\| F^* - K^*C \|_F^2 = \| F_e - K_e C_e \|_F^2. \tag{4.33}$$

Therefore, we conclude

$$\frac{\sigma_{\max}(X^*)}{\sigma_{\min}(X^*)} + \| F^* - K^*C \|_F^2 = \frac{\sigma_{\max}(X_e)}{\sigma_{\min}(X_e)} + \| F_e - K_e C_e \|_F^2 \tag{4.34}$$

with choices of

$$G_e = \begin{pmatrix} G^* & 0 \\ 0 & X_3 \tilde{A}_i \end{pmatrix} \quad \text{and} \quad K_e = \begin{pmatrix} K^* & 0 \\ 0 & X_3 \tilde{A}_i X_3^{-1} \end{pmatrix} \tag{4.35}$$

with X_3 as in (4.23). This concludes the proof. □

This theorem is useful for finding a low-order stabilizing controller because it shows how, by sequentially increasing the order of the controller, J can be guaranteed to decrease. Since a small enough value of each term of J confines the spectrum of $A + BKC$ to Ω (in accordance with (3.9)) the algorithm eventually stabilizes the system by sequentially increasing the order of controllers.

5. Examples. The algorithm developed in the last section is applied to several examples here. The gradient calculations of Theorem 4.6 are used along with the Harwell subroutine package.

Example 1. The first example is a simplified model of the NASA F-8 Digital Fly-By-Wire (DFBW) airplane [16] and its dynamic equation of lateral directional is as follows:

$$\frac{d}{dt}\begin{pmatrix} p \\ r \\ \beta \\ \phi \end{pmatrix} = \begin{pmatrix} -2.6 & 0.25 & -38.0 & 0 \\ -0.075 & -0.27 & 4.4 & 0 \\ 0.078 & -0.99 & -0.23 & 0.052 \\ 1.0 & 0.078 & 0 & 0 \end{pmatrix}\begin{pmatrix} p \\ r \\ \beta \\ \phi \end{pmatrix} + \begin{pmatrix} 17.0 & 7.0 \\ 0.82 & -3.2 \\ 0 & 0.046 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix},$$

$$\begin{pmatrix} r \\ \phi \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} p \\ r \\ \beta \\ \phi \end{pmatrix}.$$

The given design specifications [16] are that the closed-loop poles must be left of the line $s = -0.2$, i.e., $\gamma = 0.2$, and the damping factor is $\geqq 0.7$, i.e., $\phi = \pi/4$ in Fig. 4.2. Total equilibrium velocity $V_0 = 620$ ft/s (Mach = 0.6) and equilibrium angle for the optimization problem initial values are chosen to be

$$\tilde{A}_0 = \begin{pmatrix} -3 & 2 & & \\ -2 & -3 & & \\ & & -5 & 3 \\ & & -3 & -5 \end{pmatrix},$$

$$G_0 = \begin{pmatrix} 1 & 1.5 & 0.5 & -2 \\ 5 & 1 & -0.25 & 0.5 \end{pmatrix},$$

$$K_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

After 41 gradient iterations minimizing J in (3.10) the following zeroth-order stabilizing compensator is obtained:

$$K^* = \begin{pmatrix} 4.60357 & -1.75629 \\ 5.21515 & -1.85922 \end{pmatrix}.$$

Note that the order of pole placement compensators (both Brasch and Pearson [3] and Kimura [4]) is one. The corresponding data is shown in Tables 1.1, 1.2, and Fig. 5.1. For comparison, the same problem was run without including the condition number term in J (i.e., $\alpha_1 = 0$ in (3.10)). It is seen from the corresponding data, shown in Table

TABLE 1.1
Eigenvalues for Example 1.
($\alpha_1 = 1$, $\alpha_2 = 1$, $\phi = \pi/4$, $\zeta \geqq 0.7$, $\gamma = -0.1$)

A_0	$A_0 + B_0K_0^0C$	$A_0 + B_0F_0^*$	$A_0 + B_0K_0^*C_0$
$-2.39 \pm j0.00$	$-2.39 \pm j0.00$	$-9.45 \pm j3.70$	$-7.58 \pm j4.96$
$+0.00 \pm j0.00$	$+0.00 \pm j0.00$	$-0.34 \pm j0.29$	$-0.42 \pm j0.33$
$-0.34 \pm j2.62$	$-0.34 \pm j2.62$		

TABLE 1.2
Performance indices.

	J	$\|F_0 - K_0C_0\|_F^2$	$k(X_0)$
Initial	115.9021	61.3301	94.572
Optimal	47.03439	0.06839	46.966

TABLE 1.3
Eigenvalues for Example 1.
($\alpha_1 = 0$, $\alpha_2 = 1$, $\phi = \pi/4$, $\zeta \geqq 0.7$, $\gamma = 0.1$)

A_0	$A_0 + B_0K_0^0C_0$	$A_0 + B_0F_0^*$	$A_0 + B_0K_0^*C_0$
$-2.39 \pm j0.00$	$-2.39 \pm j0.00$	$-2.39 \pm j0.01$	$-1.44 \pm j2.54$
$+0.00 \pm j0.00$	$+0.00 \pm j0.00$	$-2.42 \pm j0.32$	$-3.44 \pm j0.00$
$-0.34 \pm j2.62$	$-0.34 \pm j2.62$		$-1.15 \pm j0.00$

TABLE 1.4
Performance indices.

	J	$\|F_0 - K_0C_0\|_F^2$	$k(X_0)$
Initial	155.9021	61.3301	94.572
Optimal	233089.02	0.01530	233089

1.3, 1.4, and Fig. 5.2, that the condition number increases significantly, and although stabilization is achieved, the closed-loop eigenvalues fail to be in Ω.

Example 2. Consider the symmetric vibration model of the standard Draper/RPL satellite shown in Fig. 5.3. The dynamic equations, taken from [17] are:

$$\frac{d}{dt}\begin{pmatrix}\theta\\ q_1\\ q_2\\ \dot\theta\\ \dot q_1\\ \dot q_2\end{pmatrix} = A\begin{pmatrix}\theta\\ q_1\\ q_2\\ \dot\theta\\ \dot q_1\\ \dot q_2\end{pmatrix} + B\begin{pmatrix}u_1\\ u_2\end{pmatrix},$$

$$\begin{pmatrix}y_1\\ y_2\end{pmatrix} = C\begin{pmatrix}\theta\\ q_1\\ q_2\\ \dot\theta\\ \dot q_1\\ \dot q_2\end{pmatrix}$$

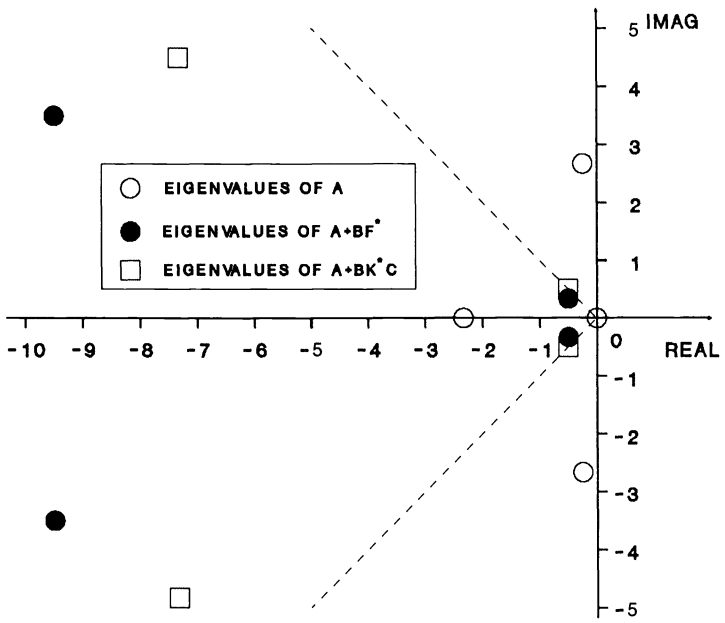


FIG. 5.1. *Eigenvalue locations corresponding to Table* 1.1.

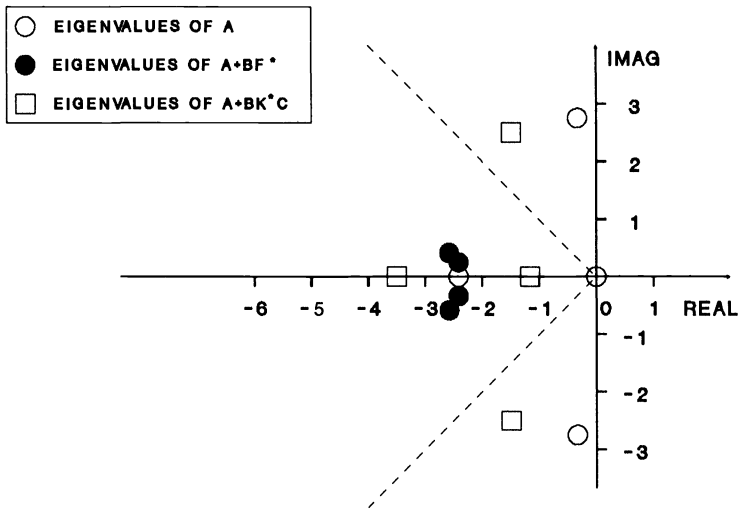


FIG. 5.2. *Eigenvalue locations corresponding to Table* 1.3.

where

$$A=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 14.8732 & 32.8086 & 0 & 0 & 0 \\ 0 & -146.702 & -7476.64 & 0 & 0 & 0 \\ 0 & -41.8468 & -2699.36 & 0 & 0 & 0 \end{pmatrix},$$

$$B=\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -0.04168 & 0.23623 \\ 10.38611 & -25.647 \\ 3.725120 & -9.1629 \end{pmatrix},$$

$$C=\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

From the design specifications in [17], it follows that the closed-loop system must have poles to the left of $s = -0.5$. For the minimization of J the initial values are chosen to be

$$\tilde{A}_0=\begin{pmatrix} -0.2 & 2 & & & & \\ -2 & -0.2 & & & & \\ & & -1 & 10 & & \\ & & -10 & -1 & & \\ & & & & -0.5 & 1 \\ & & & & -1 & -0.5 \end{pmatrix},$$

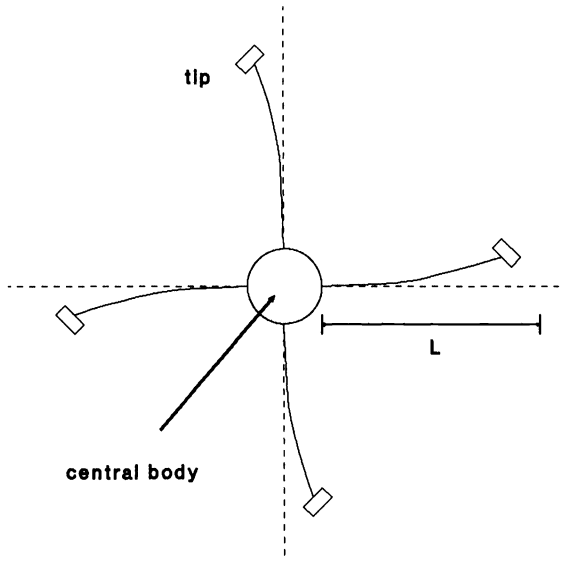


FIG. 5.3. *Draper/RPL symmetric vibrational model.*

$$G_0 = \begin{pmatrix} 1.125 & 1.5 & -0.5 & 3.5 & 1.5 & 2 \\ -1 & 2.5 & 1.6 & 4 & 0.5 & -1 \end{pmatrix},$$

$$K_0 \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

After 67 iterations, the following zeroth-order stabilizing controller is obtained:

$$K^* = \begin{pmatrix} -90.97491 & 20.62868 \\ -197.646 & 5.326668 \end{pmatrix}.$$

TABLE 2.1
Eigenvalues for Example 2.
($\alpha_1 = 1$, $\alpha_2 = 1$, $\gamma = 0.5$)

A_0	$A_0 + B_0K_0^0C$	$A_0 + B_0F_0^*$	$A_0 + B_0K_0^*C_0$
$+0.00 \pm j53.1$	$+0.00 \pm j53.1$	$-2.89 \pm j36.7$	$-3.58 \pm j30.7$
$+0.00 \pm j5.43$	$+0.00 \pm j5.43$	$-2.18 \pm j0.30$	$-2.41 \pm j0.67$
$+0.00 \pm j0.00$	$+0.00 \pm j0.00$	$-0.88 \pm j5.82$	$-1.45 \pm j6.08$
$+0.00 \pm j0.00$	$+0.00 \pm j0.00$		

TABLE 2.2
Performance indices.

	J	$\|F_0 - K_0C_0\|_F^2$	$k(X_0)$
Initial	11965915	11965506	409.2925
Optimal	587.2232	45.63520	541.5880

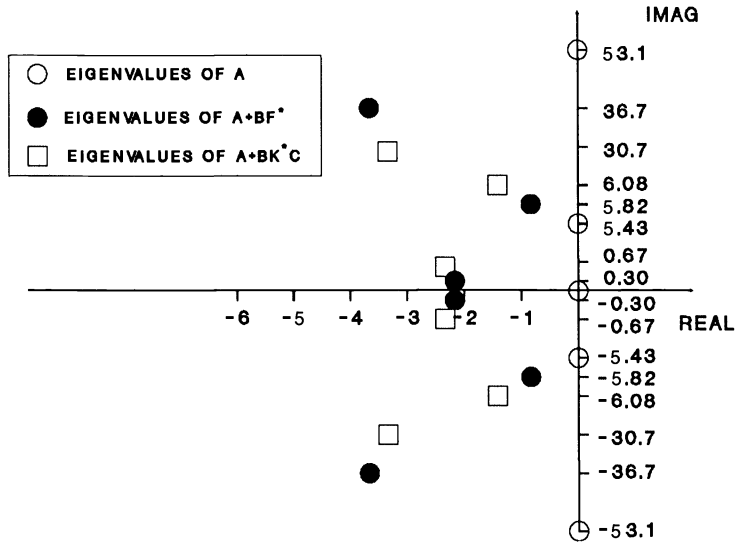


FIG. 5.4. *Eigenvalue locations corresponding to Table* 2.1.

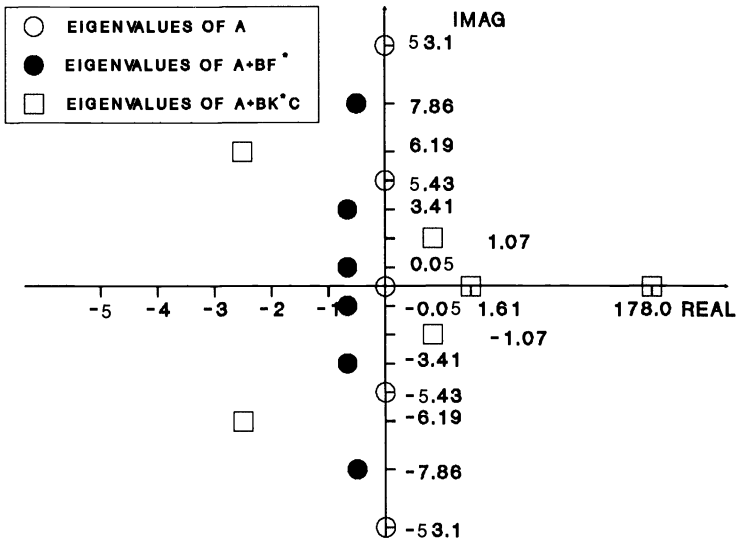


FIG. 5.5. *Eigenvalue locations corresponding to Table* 2.3.

Note that the order of pole-placement compensators (both Brasch and Pearson [3] and Kimua [4]) is three. Tables 2.1, 2.2, and Fig. 5.4 display the performance indices and the corresponding eigenvalue locations. For the purpose of comparison, the problem was also run with the condition number term left out of the performance index (i.e., $\alpha_1 = 0$). In this case the algorithm fails to stabilize the system as shown in Tables 2.3 and 2.4 and in Fig. 5.5. This example illustrates that both terms of the performance index need to be considered in the stabilization procedure.

TABLE 2.3
Eigenvalues for Example 2.
($\alpha_0 = 0$, $\alpha_2 = 1$, $\gamma = 0.5$)

A_0	$A_0 + B_0K_0^0C_0$	$A_0 + B_0F_0^*$	$A_0 + B_0K_0^*C_0$
$+0.00 \pm j53.1$	$+0.00 \pm j53.1$	$-0.63 \pm j0.05$	$+178.0 \pm j0.00$
$+0.00 \pm j5.43$	$+0.00 \pm j5.43$	$-0.66 \pm j3.41$	$-2.57 \pm j6.19$
$+0.00 \pm j0.00$	$+0.00 \pm j0.00$	$-0.59 \pm j7.86$	$+1.61 \pm j0.00$
$+0.00 \pm j0.00$	$+0.00 \pm j0.00$		$+0.83 \pm j1.07$

TABLE 2.4
Performance indices.

	J	$\|F_0 - K_0C_0\|_F^2$	$k(X_0)$
Initial	11965915	11965506	409.2925
Optimal	383859.66	490.5633	383369.1

6. Concluding remarks. The results given here are algorithmic in nature and can be improved on by developing constructive necessary and sufficient conditions for stabilizability with a fixed-order controller. This in turn will require effective ways of characterizing the Hurwitz region. These problems are difficult and have received very little attention in the literature. Finally, we mention that the algorithm neither guarantees a "global" minimum nor does it always find a stabilizing controller of a prescribed order whenever one exists. The existence of stabilizing controllers of a fixed order is still our unsolved problem.

Appendix.

Proof of Theorem 4.6.

(a)

$$J = \alpha_1 \frac{\sigma_{\max}(X)}{\sigma_{\min}(X)} + \alpha_2 \operatorname{trace}\{(F-KC)^T(F-KC)\}. \tag{A1}$$

Let

$$\begin{aligned} J_1 &:= \frac{\sigma_{\max}(X)}{\sigma_{\min}(X)} \\ &= \operatorname{trace}\left\{\frac{\sigma_{\max}(X)}{\sigma_{\min}(X)}\right\} \end{aligned} \tag{A2}$$

and

$$\Delta J_1 = \operatorname{trace}\left\{\frac{1}{\sigma_{\min}^2(X)}(\sigma_{\min}(X)\Delta\sigma_{\max}(X)) - \sigma_{\max}\,\Delta\sigma_{\min}(X)\right\}. \tag{A3}$$

Note that

$$\Delta\sigma_{\max}(X) = u_a^T \Delta X v_a, \tag{A4}$$

$$\Delta\sigma_{\min}(X) = u_i^T \Delta X v_i \tag{A5}$$

where v_i and u_i are left and right singular vectors corresponding to $\sigma_{\min}$ and v_a and v_a are for $\sigma_{\max}$. Thus,

$$\Delta J_1 = \frac{1}{\sigma_{\min}^2(X)} \operatorname{trace}\{\sigma_{\min}(X) v_i u_i^T - \sigma_{\max}(X) v_a u_a^T\} \Delta X. \tag{A6}$$

Now

$$\begin{aligned} J_2 &:= \operatorname{trace}\{(F-KC)^T(F-KC)\} \\ &= \operatorname{trace}\{F^TF-(KC)^TF-F^T(KC)+(KC)^T(KC)\} \\ &= \operatorname{trace}(F^TF) - 2\operatorname{trace}\{(KC)^TF\} + \operatorname{trace}\{(KC)^T(KC)\} \end{aligned} \tag{A7}$$

and

$$\begin{aligned} \Delta J_2 &= 2\operatorname{trace}(F^T\Delta F) - 2\operatorname{trace}\{(KC)^T\Delta F\} \\ &= 2\operatorname{trace}\{[F^T-(KC)^T]\Delta F\}. \end{aligned} \tag{A8}$$

Now we have

$$\begin{aligned} \Delta J = &\frac{\alpha_1}{\sigma_{\min}^2(X)} \operatorname{trace}\{\sigma_{\min}(X) v_i u_i^T - \sigma_{\max}(X) v_a u_a^T\} \Delta X \\ &+ 2\alpha_2 \operatorname{trace}\{(F^T-(KC)^T)\Delta F\}. \end{aligned} \tag{A9}$$

From $F = GX^{-1}$, the gradient of F with respect to G is given directly as

$$\begin{aligned} \Delta F &= \Delta G X^{-1} + G\Delta(X^{-1}) \\ &= \Delta G X^{-1} - GX^{-1}\Delta X X^{-1} \\ &= \Delta G X^{-1} - F\Delta X X^{-1} \\ &= (\Delta G - F\Delta X)X^{-1}. \end{aligned} \tag{A10}$$

Substituting (A10) into (A9), we have

$$\begin{aligned} \Delta J = &2\alpha_2 \operatorname{trace}\{(F^T-(KC)^T)\Delta G X^{-1}\} \\ &+ \operatorname{trace}\left\{\frac{\alpha_1}{\sigma_{\min}^2(X)}(\sigma_{\min}(X) v_i u_i^T - \sigma_{\max}(X) v_a u_a^T)\right. \\ &\left. - 2\alpha_2 X^{-1}(F^T-(KC)^T)F\right\}\Delta X. \end{aligned} \tag{A11}$$

Using [7], we have

$$A\Delta X - \Delta X\tilde{A} = -B\Delta G \tag{A12}$$

and

$$\Delta X = \sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_{ij} A^{i-1} B \Delta G \tilde{A}^{j-1}. \tag{A13}$$

Substituting (A13) into the second term of (A11), we have

(A14)

$$\begin{aligned}\Delta J := {} & 2\alpha_2 \operatorname{trace}\{X^{-1}(F^T-(KC)^T)\Delta G\} \\ & + \operatorname{trace}\Bigg\{\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}\tilde{A}^{j-1} \\ & \quad \cdot \underbrace{\left(\frac{\alpha_1}{\sigma_{\min}^2(X)}(\sigma_{\min}(X)v_iu_i^T-\sigma_{\max}(X)v_au_a^T)-2\alpha_2X^{-1}(F^T-(KC)^T)F\right)}_{X_f} \\ & \qquad\qquad A^{i-1}B\Delta G\Bigg\} \\ = {} & 2\alpha_2 \operatorname{trace}\{X^{-1}(F^T-(KC)^T)\Delta G\} \\ & + \operatorname{trace}\Bigg\{\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}\tilde{A}^{j-1}X_fA^{i-1}}_{U}B\Delta G\Bigg\} \\ = {} & \operatorname{trace}\{2\alpha_2X^{-1}(F^T-(KC)^T)+UB\}\Delta G.\end{aligned}$$

From (A12) and (A13) it follows that U is the unique solution of

$$\tilde{A}U-UA=X_f. \tag{A15}$$

Therefore

$$\frac{\partial J}{\partial G}=2\{\alpha_2(F-KC)X^{-T}+B^TU^T\} \tag{A16}$$

where U satisfies

$$\begin{aligned}\tilde{A}U-UA=\frac{\alpha_1}{\sigma_{\min}^2(X)}\{\sigma_{\min}(X)v_iu_i^T-\sigma_{\max}(X)v_au_a^T\} \\ -2\alpha_2X^{-1}(F^T-(KC)^T)F.\end{aligned} \tag{A17}$$

(b) Now we evaluate the gradients of (3.10) with respect to the variable elements of $\tilde{A}$. Recall the equation (A9)

$$\begin{aligned}\Delta J=\frac{\alpha_1}{\sigma_{\min}^2(X)}\operatorname{trace}\{\sigma_{\min}(X)v_i\,u_i^T-\sigma_{\max}(X)v_au_a^T\}\Delta X \\ +2\alpha_2\operatorname{trace}\{(F^T-(KC)^T)\Delta F\}.\end{aligned} \tag{A18}$$

From $F = GX^{-1}$, we compute (G is fixed)

$$\begin{aligned}\Delta F &= -GX^{-1}\Delta XX^{-1} \\ &= -F\Delta XX^{-1}.\end{aligned} \tag{A19}$$

Substituting ΔF into (A18), we have

(A20)

$$\Delta J = \text{trace}\underbrace{\left\{\frac{\alpha_1}{\sigma_{\min}^2(X)}(\sigma_{\min}(X)v_i u_i^T - \sigma_{\max}(X)v_a u_a^T) - 2\alpha_2 X^{-1}(F-(KC)^T)F\right\}}_{X_f}\Delta X$$

$$= \text{trace}\{X_f \Delta X\}.$$

Since

(A21)
$$A\Delta X - \Delta X\tilde{A} = X\Delta\tilde{A},$$

(A22)
$$\Delta X = \sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}A^{i-1}(-X\Delta\tilde{A})\tilde{A}^{j-1}.$$

Substituting (A22) into (A20), we have

(A23)
$$\Delta J = \text{trace}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}X_f A^{i-1}(-X\Delta\tilde{A})\tilde{A}^{j-1}\right\}$$
$$= -\text{trace}\underbrace{\left\{\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}\tilde{A}^{j-1}X_f A^{i-1}\right\}}_{U}X\Delta\tilde{A}\Big\}.$$

It is clear that U is the unique solution of

$$\tilde{A}U - UA = X_f$$

as in (A14):

(A24)
$$\Delta J = -\text{trace}\{UX\Delta\tilde{A}\}.$$

Therefore,

(A25)
$$\frac{\partial J}{\partial\tilde{a}_i} = -\text{trace}\left\{UX\frac{\partial\tilde{A}}{\partial\tilde{a}_i}\right\}.$$

As an example the following calculation is considered. Let

(A26)
$$U = \begin{pmatrix}u_{11} & u_{12}\\ u_{21} & u_{22}\end{pmatrix}\quad X = \begin{pmatrix}x_{11} & x_{12}\\ x_{21} & x_{22}\end{pmatrix}\quad \tilde{A} = \begin{pmatrix}\tilde{a}_1^2 & 0\\ 0 & \tilde{a}_2^2\end{pmatrix}.$$

Then

(A27)
$$\frac{\partial J}{\partial\tilde{a}_1} = 2\,\text{trace}\left\{\begin{pmatrix}u_{11} & u_{12}\\ u_{21} & u_{22}\end{pmatrix}\begin{pmatrix}x_{11} & x_{12}\\ x_{21} & x_{22}\end{pmatrix}\begin{pmatrix}2\tilde{a}_1 & 0\\ 0 & 0\end{pmatrix}\right\}$$
$$= 4\tilde{a}_1(u_{11}x_{11} + u_{12}x_{21}),$$

(A28)
$$\frac{\partial J}{\partial\tilde{a}_2} = 2\,\text{trace}\left\{\begin{pmatrix}u_{11} & u_{12}\\ u_{21} & u_{22}\end{pmatrix}\begin{pmatrix}x_{11} & x_{12}\\ x_{21} & x_{22}\end{pmatrix}\begin{pmatrix}0 & 0\\ 0 & 2\tilde{a}_2\end{pmatrix}\right\}$$
$$= 4\tilde{a}_2(u_{21}x_{12} + u_{22}x_{22}),$$

or

$$\text{(A29)} \qquad \begin{pmatrix} \dfrac{\partial J}{\partial \tilde{a}_1} \\ \dfrac{\partial J}{\partial \tilde{a}_2} \end{pmatrix} = 4 \begin{pmatrix} \tilde{a}_1 (u_{11}x_{11} + u_{12}x_{21}) \\ \tilde{a}_2 (u_{21}x_{12} + u_{22}x_{22}) \end{pmatrix}.$$

(c) Finally, the gradient of J with respect to K is easily derived:

$$\text{(A30)} \qquad \begin{aligned} \Delta J &= -2\alpha_2 \operatorname{trace} \{ CF^T \Delta K - C(KC)^T \Delta K \} \\ &= -2\alpha_2 \operatorname{trace} \{ (CF^T - C(KC)^T) \Delta K \}. \end{aligned}$$

Thus,

$$\text{(A31)} \qquad \frac{\partial J}{\partial K} = -2\alpha_2 [F - KC] C^T.$$

REFERENCES

[1] H. KWAKERNAAK AND R. SIVAN, *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.

[2] D. G. LUENBERGER, *An introduction to observers*, IEEE Trans. Automat. Control, 16 (1971), pp. 596–603.

[3] F. M. BRASCH AND J. B. PEARSON, *Pole placement using dynamic compensator*, IEEE Trans. Automat. Control, 15 (1970), pp. 34–43.

[4] H. KIMURA, *Pole assignment by gain output feedback*, IEEE Trans. Automat. Control, 20 (1975), pp. 509–516.

[5] ———, *Robust stability of a class of transfer functions*, IEEE Trans. Automat. Control, 29 (1984), pp. 788–793.

[6] S. P. BHATTACHARYYA AND E. DESOUZA, *Pole assignment via Sylvester's equations*, Systems Control Lett., 1 (1982), pp. 261–263.

[7] E. DESOUZA AND S. P. BHATTACHARYYA, *Controllability, observability and the solution of* $AX - XB = C$, Linear Algebra Appl., 39 (1981), pp. 167–188.

[8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[9] L. H. KEEL AND S. P. BHATTACHARYYA, *Low order robust stabilizer design using Hurwitz conditions*, in Proc. IEEE Conference on Decision and Control, December 1985, Ft. Lauderdale, FL.

[10] R. K. CAVIN III AND S. P. BHATTACHARYYA, *Robust and well conditioned eigenstructure assignment via Sylvester's equation*, Optimal Control Appl. Methods, 4 (1983), pp. 205–212.

[11] L. H. KEEL, J. A. FLEMING, AND S. P. BHATTACHARYYA, *Minimum norm pole assignment via Sylvester's equation*, American Mathematical Society Contemporary Mathematics Series, 47 (1985), pp. 265–272.

[12] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

[13] L. H. KEEL AND S. P. BHATTACHARYYA, *An algorithm for low order stabilizing compensator design via Sylvester's equation*, in Proc. 2nd IEEE Control Systems Soc. Symposium on CACSD, Santa Barbara, CA, March 1985.

[14] ———, *Compensator design for robust eigenstructure assignment*, in Proc. American Control Conference, Boston, MA, June 1985.

[15] P. LANCASTER, *The Theory of Matrices: With Applications*, Vol. 13, Academic Press, New York, 1970, pp. 317–322.

[16] J. R. ELLIOTT, NASA's *Advanced control law program for the* F-8 *digital-fly-by-wire aircraft*, IEEE Trans. Automat. Control, 22 (1977), pp. 753–757.

[17] D. S. BODDEN AND J. L. JUNKINS, *Eigenvalue optimization algorithms for structural/controller design iterations*, in Proc. American Control Conference, San Diego, CA, June 1984.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 200–206, April 1990

003

SOME 0–1 SOLUTIONS TO MATRIX EQUATION $A^m - A^n = lJ$—PART I*

CHI FAI HO†

Abstract. Connected graphs with adjacency matrices satisfying the matrix equation $A^m - A^n = lJ$ occur very often in the study of concurrent computation in computer science. It is shown that there are solutions even when A has the same row and column sum.

Key words. regular graph, 0–1 matrix, adjacency matrix, concurrent computation, satisfiable, network topology

AMS(MOS) subject classification. 05C

1. Introduction. Suppose $G = (V, E)$ is a connected regular directed graph with common indegree and outdegree d. Let $q_n(x, y)$ be the number of n-paths from vertex x to vertex y. Suppose there exists an integer $M > 0$ such that for any n, x, y, z,

$$|q_n(x,y) - q_n(x,z)| < M. \tag{1}$$

Then G is said to be M-satisfiable.

In the study of concurrent computation, network topologies are usually denoted by directed graphs. When such a network is put to work, a computation graph of an algorithm will be mapped onto the network such that adjacent tasks in the graph are mapped onto adjacent processors in the network. Moreover, each processor should not receive too many tasks while some others have a few assignments. When the computation graph is a complete d-ary tree, the described conditions can be formulated mathematically into (1). Various studies [1]–[4] and [6]–[10] have been conducted in search of such a network.

Let A be the adjacency matrix, or simply the matrix, of G. Since G is regular with the same indegree and outdegree, A has column and row sums all equal to d. It is well known that the (i, j) entry of A^n is exactly $q_n(i, j)$.

Subtract from A^n a suitable multiple of J, the matrix of all 1's, to obtain a nonnegative matrix A_n with at least one zero entry. If G is M-satisfiable, all entries of A_n are less than $2M$. So $\{A_n\}$ is a finite set and there are two matrices $A_m = A_n$, which means

$$A^m - A^n = lJ \tag{2}$$

for some l.

Suppose A satisfies (2). Then $\{A_n\}$ is a finite set, and there is an $M > 0$ such that the entries of each A_n are between 0 and $M - 1$ for any n. That is, G is M-satisfiable.

Define G *satisfiable* if it is M-satisfiable for some M.

THEOREM 1. *G is satisfiable if and only if $A(G)$ satisfies (2) for some l.*

Clearly any one-vertex graph or complete k-graph is satisfiable. Besides these trivial cases, is there another satisfiable graph?

A graph G has order p if there are p vertices.

The answer is affirmative for some orders. In this paper we derive the following theorems. In the coming paper [5], we will show that there are no satisfiable loop-free graphs for certain orders.

* Received by the editors December 7, 1987; accepted for publication (in revised form) March 17, 1989.

† Department of Mathematics and Computer Science, California State University, Hayward, California 94542 (lll-winken!csuh!ho).

THEOREM A. *When* $d = 2$, $p = 2^n(2^k - 1)$ *for* $n \geqq 0$, $k > 0$, *there exists a satisfiable graph of order* p.

THEOREM B. *When* $d = 2$, $p = 2^n(2^k + 1)$ *for* $n \geqq 0$, $k > 0$, k *odd, there exists a satisfiable graph of order* p.

THEOREM C. *When* $p = d^n(d^k - 1)$ *for* $n \geqq 0$, $k > 0$, *there exists a satisfiable graph of order* p.

2. Existence of satisfiable graphs. Any graph of only one vertex is always satisfiable; we are interested in nontrivial solutions. Figure 1 shows some satisfiable graphs for $d = 2$. A careful examination of these examples leads to the following general construction.

THEOREM 2. *If* $d = 2$, *there exists a satisfiable graph of order* $2^k - 1$ *for* $k > 0$.

Proof. We will construct the graphs.

Start with a complete binary tree of depth $k > 0$, and order $2^{k+1} - 1$. Let r be the root. There are two branches at r. Use a_i to denote the nodes of the left branch and b_i to denote those of the right branch. Nodes of the left branch of depth n are of depth $n + 1$ of the graph. For $2^n \leqq i < 2^{n+1}$, a_i is a node of the left branch of depth n. A similar convention applies to b_i.

We write $x \rightarrow (y, z)$ to mean that the outgoing edges of vertex x are directed to y and z. The following rules describe the graph. The graphs for $k = 1, 2, 3$ are shown in Fig. 1.

R1. $r \rightarrow (a_1, b_1)$,

R2. $a_{2^i+j} \rightarrow (a_{2^{i+1}+2j}, a_{2^{i+1}+2j+1}), 0 \leqq j < 2^i, 0 \leqq i < k-1$,

R3. $a_{2^{k-1}} \rightarrow (r, b_1)$,

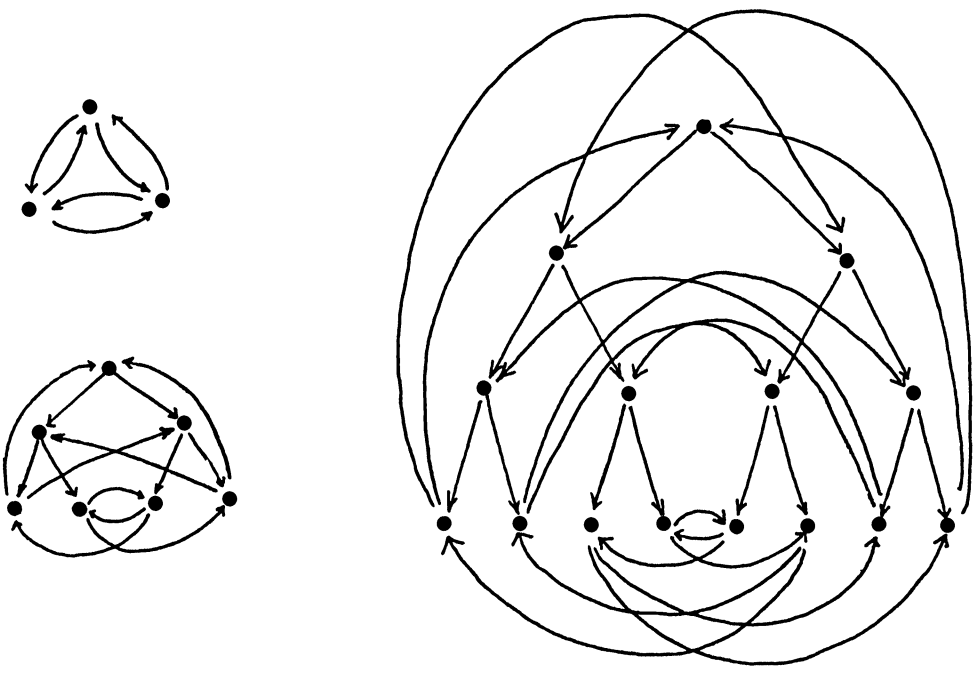


FIG. 1

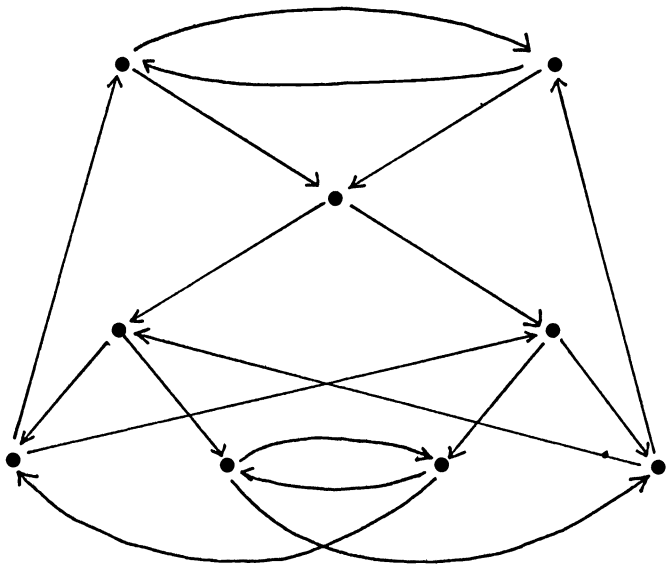


FIG. 2

R4. $a_{2^{k-1}+2^i+j} \rightarrow (b_{2^{i+1}+2j}, b_{2^{i+1}+2j+1}), 0 \leqq j < 2^i, 0 \leqq i < k-1,$

R5. Interchange roles of a and b in R2 to R4.

R1 and R2 describe the binary property of the graph. R3 through R5 describe how to wrap the outgoing edges of the leaves to the lower level nodes to complete a 4-regular directed graph with indegree and outdegree 2.

We defer the proof that the above graph is satisfiable to a later section. The matrix of the graph satisfies

$$A^{2(k+1)} - A^{k+1} = 2^{k+1}J. \qquad \square$$

THEOREM 3. *If $d = 2$ and k is an odd positive integer, there exists a satisfiable graph of order $2^k + 1$.*

Proof. Again we will construct the graphs. The construction is similar to that of Theorem 2, except R3 is changed to

R3a. $a_{2^{k-1}} \rightarrow (a, b_1),$

R3b. $a \rightarrow (r, b),$

where a and b are two extra nodes. An example is shown in Fig. 2.

The proof that these graphs are satisfiable is similar to that of Theorem 2 and is omitted. $\square$

The above construction gives satisfiable graphs only when k is odd. It fails when k is even. In particular, there is no satisfiable loop-free graph of order 5. This fact will be proved in [5].

3. Line graphs of satisfiable graphs. Suppose G is any directed graph. The line graph G' of G is defined as follows: $V(G')$, the vertices of G', is the set $E(G)$ of the edges of G. If (x, y) and (y, z) are two edges in G, then there is an edge in G' from vertex (x, y) to (y, z).

THEOREM 4. *Let A be the matrix of G. Suppose A satisfies some equation $f(A) = lJ$, where $f(x)$ is a polynomial with rational coefficients. Then the matrix A' of G' satisfies $A'f(A') = lJ$, where J is of corresponding order. The converse is also true.*

Proof. Recall that $q_n(i, j)$ is the (i, j) entry of A^n. Denote by $g_A(i, j)$ the number obtained by replacing each x^n in $f(x)$ by $q_n(i, j)$. Then, since $f(x)$ is a polynomial in x, $g_A(i, j) = l$ for all i, j.

Suppose v_i and v_j are two vertices of G. There are two incoming edges for v_i, e_{i1}, and e_{i2}; and two outgoing edges for v_j, e_{j1}, and e_{j2}. Each n-path from v_i to v_j is a sequence of edges from an outgoing edge of v_i to an incoming edge of v_j. By attaching e_{i1} and e_{j1} to the sequence, we obtain an $(n + 1)$-path from e_{i1} to e_{j1} in G'. Similarly, each $(n + 1)$-path in G' from e_{i1} to e_{j1} corresponds to an n-path in G from v_i to v_j. Hence the (i, j) entry of A^n is identical to the (e_{i1}, e_{j1}) entry of A'^{n+1}. If x^{n+1} in $xf(x)$ is replaced by $q_{n+1}(e_{i1}, e_{j1})$ and $g_{A'}(e_{i1}, e_{j1})$ is the result, then $g_{A'}(e_{i1}, e_{j1}) = g_A(i, j) = l$. That is, $A'f(A') = lJ$.

The converse follows similarly. □

COROLLARY 5. *G is satisfiable if G' is satisfiable.*

Proof. It follows from Theorem 4 by choosing $f(x) = x^m - x^n$.

If G has the same indegree and outdegree d, so does G' and the order of G' is d times that of G.

Combining Theorems 2, 3, Corollary 5, and the above fact, we prove Theorems A and B stated in the Introduction. □

4. Generalization. With a similar construction as in the proof of Theorem 2, we obtain satisfiable graphs of orders $(d^k - 1)/(d - 1)$ for any $d \geqq 2$. We illustrate the idea for $d = 3$ and leave the general construction as an exercise.

First, construct a complete ternary tree of depth $k > 0$ with edges directed from parents to children. Each child of the root is the root of the subsequent ternary branch. Let X, Y, and Z be the three branches. Leaves of X are divided into three groups, each of which descends from a child of the root of X. Choose two such groups. Spread all outgoing edges of each group to the leaves of Y or Z. They cover exactly all leaves of Y and Z.

For the remaining group, direct all outgoing edges to the remaining nodes in Y and Z. However, there is an edge left without a target. It points to the root of the original tree.

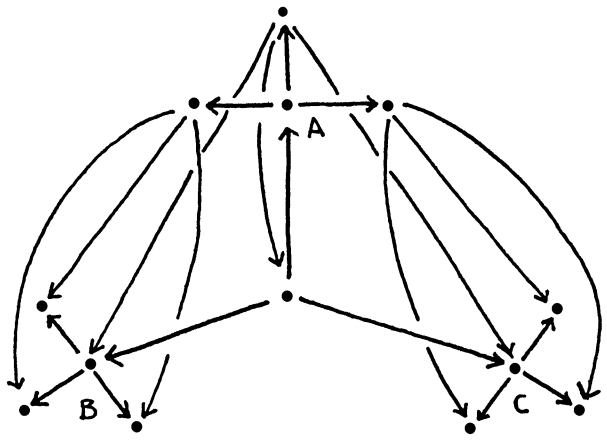


FIG. 3

Repeat the same procedure to Y and Z.

An illustration of the construction is shown in Fig. 3. Readers should convince themselves that the graphs are satisfiable.

5. Proof of Theorem 2. Denote A_i^n the ith row of A^n and represent the row pictorially by assigning to each vertex j the number $q_n(i, j)$. Since we can subtract any multiple of J from A^n, we can subtract a constant from all components. Any row of the matrix is identified with a $(2^k - 1)$-vector. In this section, vector equality is defined up to a multiple of $v = (1, \cdots, 1)$.

At this point, we modify our previous notations as follows: Replace $A_{a_i}^n$ by $A(n, i)$, $A_{b_i}^n$ by $B(n, i)$, $a(i)$ by a_i, and $b(i)$ by b_i, and $F(n, i)$ by either $A(n, i)$ or $B(n, i)$.

LEMMA 6. *For each i, there exists $m_i < k + 2$ such that the vector $F(m_i, i)$ has a* 1 *at positions r, $a(1), \cdots, a(2^{k_i} - 1)$, or r, $b(1), \cdots, b(2^{k_i} - 1)$ and* 0, *elsewhere. That is, we have the following situation in Fig.* 4.

Proof. In view of the symmetry along a_i's and b_i's, we consider only a_i's.

Let L be the left branch of r, so $a(2^i + j)$ is a node on the ith level of L. As $i \leqq k$, $A(k - i, 2^i + j)$ has 1 at positions $a(2^k + j2^{k-1} + n)$, $0 \leqq n < 2^{k-i}$ and 0, elsewhere. If $j = 0$, $A(k - i + 1, 2^i + j)$ has the desired property.

If $j > 0$, let l, m be such that $j = 2^l + m$, $0 \leqq m < l$, $A(k - i + 1, 2^l + m)$ has 1 at positions $b(2^{(k-i)+(l-1)} + m2^{k-i-1} + n)$, $0 \leqq n < 2^{k-i+1}$ and 0 elsewhere. But $b(k - i + 1, 2^l + m)$ has the same children as $a(k - i + 1, 2^i + j)$. As $l < i$, by an induction argument on i, there exists an $m_i \leqq k + 1$ such that $B(m_i, 2^l + m)$ has the desired property. Certainly $m_i > k - i + k$ for any positive i. Thus, $A(k - i + 1, 2^i + j)$ has the desired property. □

Suppose $A(m, l)$ is the row described in Lemma 8. The rows of $a(l)$ of succeeding powers of A are as given in Fig. 5.

$$A(m+k+1, l) = A(m, l).$$

Therefore, $A(m + k + 1, l) - A(m, l) = m'v$. Hence, for each i there exists an m_i and l_i such that

$$F(m_i+k+1, i) - F(m_i, i) = l_i^v, \qquad F = A, B.$$

As each $m_i < n + 1$, we have

$$F(2(k+1), i) - F(k+1, i) = 2^{k+1-m_i} l_i v, \qquad F = A, B.$$

As $A_r^{k+1} - A_r^0 = v$ implying $A_r^{2(k+1)} - A_r^{k+1} = 2_v^{k+1}$, we deduce $A^{2(k+1)} - A^{k+1} = 2^{k+1}J$.

This completes the proof of Theorem 2. □

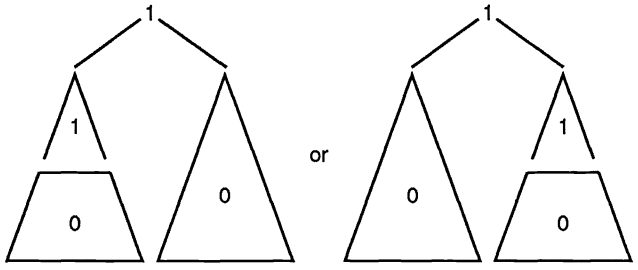


FIG. 4

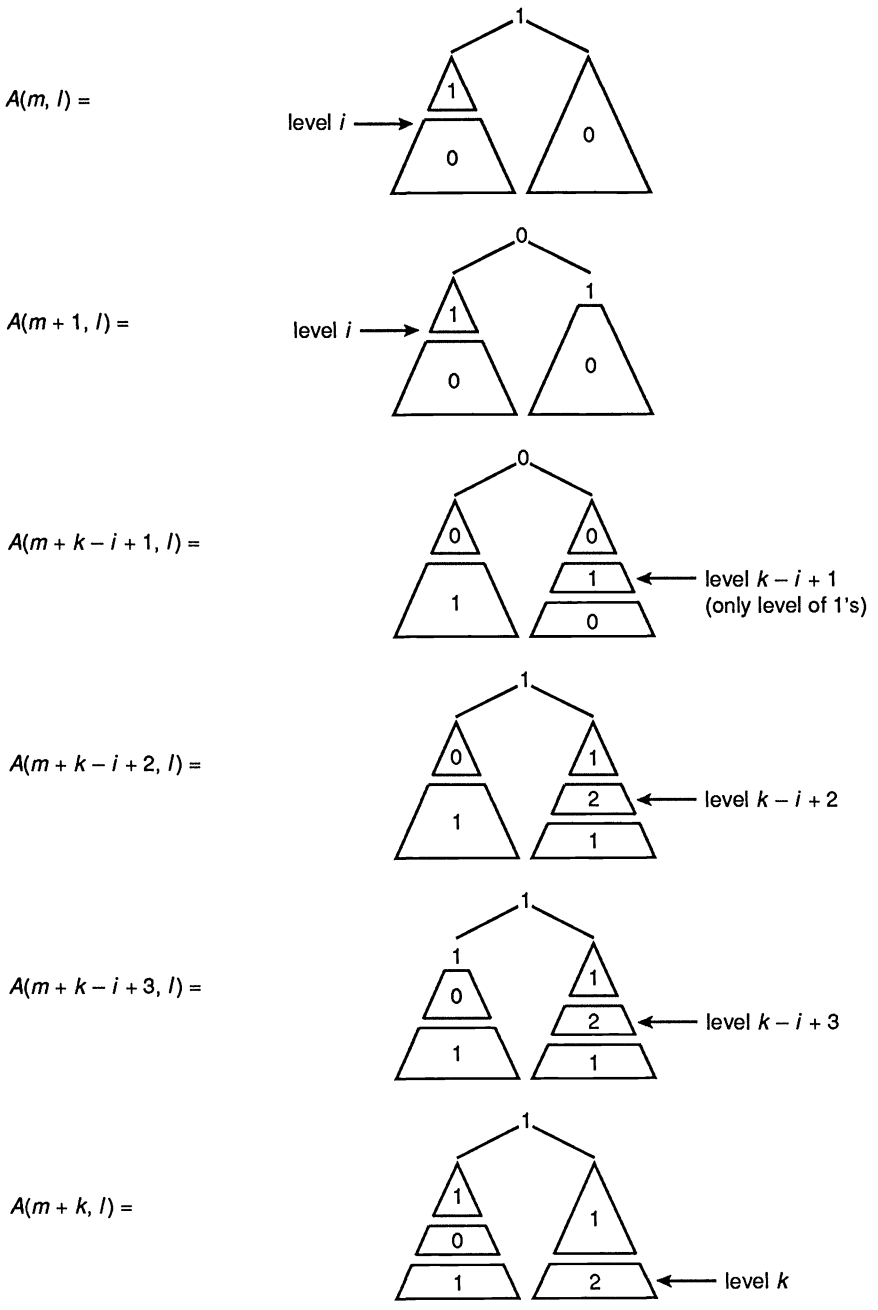


FIG. 5

REFERENCES

[1] J. L. BENTLEY AND H. T. KUNG, *A tree machine for searching problems*, Proc. International Conference on Parallel Processing, 1979, pp. 257–266.

[2] S. A. BROWNING, *The tree machine: A highly concurrent computing environment*, Ph.D. thesis, Science Dept., California Institute of Technology, Pasadena, CA, 1980.

[3] A. M. DESPAIN AND D. A. PATTERSON, *X-tree: A tree structured multiprocessor computer architecture*, Conference Proc. of the 5th Symposium on Computer Architecture, 1978, pp. 144–151.

[4] J. R. GOODMAN AND C. H. SEQUIN, *Hypertree: A multiprocessor interconnection topology*, Computer Science Tech. Report 4227, April, 1981.
[5] C. F. HO, *Some solutions to matrix polynomial* $A^m - A^n = lJ$, II, to be submitted.
[6] P. LI, *The tree machine operation system*, Tech. Report 4618, Computer Science Dept., California Institute of Technology, Pasadena, CA, July, 1981.
[7] P. LI AND A. L. MARTIN, *The sneptree—A versatile interconnection network*, Tech. Report 5194, Computer Science Dept., California Institute of Technology, Pasadena, CA, 1985.
[8] M. L. SCHLUMBERGER, *De Bruijn communications networks*, Ph.D. thesis, Computer Science Dept., Stanford University, Stanford, CA, 1974.
[9] C. L. SEITZ, *The cosmic cube*, Comm. ACM, 28 (1985), pp. 22–33.
[10] J. L. A. VAN DE SNEPSCHEUT, *Mapping a dynamic tree on a fixed graph*, unpublished article, February, 1981.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 207–212, April 1990

004

SETS OF POSITIVE OPERATORS WITH SUPREMA*

W. N. ANDERSON, JR.†, T. D. MORLEY‡, AND G. E. TRAPP§

Abstract. Let K be an n-by-n self-adjoint matrix and let $K \circledast X$ be the Kronecker product of the matrix K and the linear operator X. Thus if $X: H \to H$, where H is a Hilbert space, then $K \circledast X: H^n \to H^n$. Given a positive operator A, operators of the form $A + K \circledast X$, where X is a positive operator, are studied. The signs of the eigenvalues of K and the rank of K play crucial roles in characterizing suprema of the following set: $\{X \geqq 0 \mid A + K \circledast X \geqq 0\}$. It is shown that $\{X \geqq 0 \mid A + K \circledast X \geqq 0\}$ has a supremum for all $A \geqq 0$ if and only if K has exactly one negative eigenvalue. For the cases rank $K = 1$ and rank $K = 2$, the existence of the supremum is already known under the names "shorted operator" and "cascade limit," respectively. The suprema in the case that rank $K > 2$ are new nonlinear operations.

Key words. operator inequality, Schur completement, shorted operator

AMS(MOS) subject classifications. 15A45, 47D15

1. Introduction and the shorted operator. Let H be a Hilbert space. A linear operator X on H is termed *positive* if $(Xx, x) \geqq 0$ for all $x \in H$, and $X = X^*$. We write $X \leqq Y$ to mean $Y - X$ is positive. If K is an n-by-n matrix, we define the *Kronecker product* $K \circledast X$ as

$$K \circledast X = \begin{pmatrix} k_{11}X & k_{12}X & & \\ k_{21}X & k_{22}X & & \\ & & \ddots & \\ & & & k_{nn}X \end{pmatrix}$$

and thus $K \circledast X$ is an operator on H^n, the space of all n-tuples, $\tilde{x} = (x_1, \cdots, x_n)$ with inner product $(\tilde{x}, \tilde{y}) = \sum_{i=1}^{n} (x_i, y_i)$. In the sequel we will consider only self-adjoint matrices K. Now, given a positive operator A, partitioned with respect to H^n, as follows:

$$A = \begin{pmatrix} A_{11} & A_{12} & & \\ A_{21} & & & \\ & \ddots & & \\ & & & A_{nn} \end{pmatrix}$$

we define the *shorted operator* of A as follows:

$$S(A) = \sup \{X \geqq 0 \mid A \geqq (X \oplus 0)\} \text{ where}$$

$$X \oplus 0 = \begin{pmatrix} X & 0 \cdots 0 \\ 0 & \\ \vdots & \vdots \\ 0 & \cdots 0 \end{pmatrix}.$$

(See [6], [12].)

We note that the supremum in the definition of $S(A)$ is with respect to the partial order defined above. There is no a priori reason for the supremum to exist. In fact Pos (H), the cone of positive operators on H, is a lattice if and only if H is one-dimensional.

* Received by the editors June 6, 1988; accepted for publication (in revised form) May 3, 1989.

† Department of Mathematics and Computer Science, Fairleigh Dickinson University, Teaneck, New Jersey 07666.

‡ School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332 (MA201TM@GITVM1).

§ Department of Statistics and Computer Science, West Virginia University, Morgantown, West Virginia 26506 (UN106020@WVNVAXB).

However, the supremum does exist and the following theorem summarizes important properties of the shorted operator; proofs of these results may be found in [6].

THEOREM 1. *Suppose A is positive. Then*

(i) *$S(A)$ exists and is positive;*

(ii) *If $A \leqq B$, then $S(A) \leqq S(B)$;*

(iii) $S(A) \leqq A_{11}$,

(iv) *If A_n is a monotone decreasing sequence of positive operators, then* $\lim_{n\to\infty} S(A_n) = S(\lim_{n\to\infty} A_n)$; *these are strong operator limits.*

We remark (see [1]) that if A is invertible then $S(A)$ has the following representation:

$$S(A) = A_{11} - (A_{12}A_{13}\cdots A_{1n}) \begin{pmatrix} A_{22} & A_{23} & & \\ A_{32} & & \ddots & \\ & & & A_{nn} \end{pmatrix}^{-1} \begin{pmatrix} A_{21} \\ A_{31} \\ \cdot \\ \cdot \\ A_{n1} \end{pmatrix}.$$

Thus the shorted operator is a generalization of the classic *Schur complement.* The terminology "shorted operator" comes from applications in network theory (see [1], [6]).

2. Preliminaries and examples of suprema problems. Our primary goal in this paper is to generalize results that pertain to the existence of suprema of sets of positive operators. Our results include, as special cases, Propositions 2 to 6, which are given below. We will see that each of these propositions may be reformulated in terms of the supremum of an $A + K \circledast X$ problem for the appropriate choice of A and K.

PROPOSITION 2 (Krein [12], Anderson and Trapp [6]). *Let*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

be positive; then there is a unique $X_\infty \geqq 0$ such that

$$X_\infty = \sup \left\{ X \geqq 0 \,\middle|\, \begin{pmatrix} X & 0 \\ 0 & 0 \end{pmatrix} \leqq A \right\}.$$

Proposition 2 is, of course, merely a restatement of Theorem 1(i) and guarantees the existence of the shorted operator.

PROPOSITION 3 (Ando [7]; see also Fujii [11] and Anderson, Morley, and Trapp [2]). *Let B and C be positive operators; then there is a unique $X_\infty \geqq 0$ such that*

$$X_\infty = \sup \left\{ X \geqq 0 \,\middle|\, \begin{pmatrix} B & X \\ X & C \end{pmatrix} \geqq 0 \right\}.$$

The operator X_∞ of Proposition 3 is called the geometric mean of B and C and is denoted $B\#C$.

PROPOSITION 4 (Anderson and Trapp [6]). *Let B and C be positive. Then there is a unique X_∞ such that*

$$X_\infty = \sup \left\{ X \geqq 0 \,\middle|\, \begin{pmatrix} B - X & B \\ B & B + C \end{pmatrix} \geqq 0 \right\}.$$

The operator X_∞ of Proposition 4 is called the *parallel sum* of B and C and denoted by $B{:}C$. We note that Proposition 4 follows as a corollary of Proposition 2; we have included it as a separate statement because parallel addition is very important in its own right.

PROPOSITION 5 (Ando [8]). *Let*

$$A=\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

be positive. Then there exists a unique $X_\infty \geqq 0$ *such that*

$$X_\infty = \sup\left\{X \geqq 0 \,\middle|\, \begin{pmatrix} A_{11}-X & A_{12} \\ A_{21} & A_{22}+X \end{pmatrix} \geqq 0\right\}.$$

The operator X_∞ of Proposition 5 is called the *cascade limit* of the operator A. (See Ando [8], Anderson, Reynolds, and Trapp [5], and Anderson, Morley, and Trapp [4] for additional information about the cascade operation and the cascade limit.)

PROPOSITION 6. *Let C and D be positive with D invertible and let E be a linear operator. Then there is a unique* X_∞ *such that*

$$X_\infty = \sup\left\{X \geqq 0 \,\middle|\, \begin{pmatrix} C+EDE^* & E+X \\ E^*+X & D^{-1} \end{pmatrix} \geqq 0\right\}.$$

The operator X_∞ of Proposition 6 solves the Riccati equation

$$XDX+XDE^*+EDX=C.$$

This proposition was first conjectured by Trapp [13] and independently proven by Ando and Bunce (see [9]). Setting $E = 0$ in Proposition 6 yields Proposition 3 for the case C invertible.

3. The existence of a supremum of $\{X \geqq D \mid A + K \circledast X \geqq 0\}$. In this section we study the existence of

$$\sup\{X \geqq 0 \mid A+K \circledast X \geqq 0\}$$

and prove that the above supremum exists for all $A \geqq 0$ if and only if K has exactly one negative eigenvalue. We count eigenvalues with multiplicity and thus

$$K=\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = \operatorname{diag}(-1,-1)$$

has *two* negative eigenvalues. In Theorem 7 we present a basic result; we consider the case of a diagonal K having exactly one negative eigenvalue that we assume is in the (1, 1) position of K.

THEOREM 7. *Let* $A \geqq 0$, *and let* $K = \operatorname{diag}(-1, 1, 1, \cdots, 1, 0, 0, \cdots)$; *then there is an* X_∞ *such that* $X_\infty = \sup\{X \geqq 0 \mid A + K \circledast X \geqq 0\}$.

Proof. Let $L = \operatorname{diag}(0, 1, 1, \cdots, 1, 0, 0, \cdots)$. We show the existence of an operator $X_\infty \leqq A_{11}$ such that $S(A + L \circledast X_\infty) = X_\infty$. Consider a sequence of X's defined as follows: $X_0 = A_{11}$ and $X_{n+1} = S(A + L \circledast X_n)$. First we note that $X_1 = S(A + L \circledast X_0) \leqq A_{11} = X_0$ and by induction

$$0 \leqq X_{n+1} = S(A+L \circledast X_n) \leqq S(A+L \circledast X_{n-1}) = X_n.$$

Thus the X_n are a monotonically decreasing sequence and converge to a limit X_∞, and since $S(\ )$ is continuous under monotone decreasing limits (Theorem 1(iv)), we have $X_\infty = S(A + L \circledast X_\infty)$.

The supremum definition of the shorted operator implies that

$$A+L \circledast X_\infty \geqq \begin{pmatrix} S(A+L \circledast X_\infty) & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} X_\infty & 0 \\ 0 & 0 \end{pmatrix}.$$

And since $L \circledast X_\infty - X_\infty \oplus 0 = K \circledast X_\infty$, we have $A + K \circledast X_\infty \geqq 0$. Therefore the X_∞ we have found is in the set.

Next we must show that it is an extremum. Take $X \geqq 0$ with $A + K \circledast X \geqq 0$. We have to prove that $X_\infty \geqq X$. For this it suffices to prove by induction that $X_n \geqq X$ for each n. First $X_0 = A_{11} \geqq X$ is obvious because $A + L \circledast X \geqq K \oplus 0$. And now assume that $X_n \geqq X$; then we have the following string of inequalities:

$$A + L \circledast X_n \geqq A + L \circledast X \geqq X \oplus 0.$$

Then by shorting the left and right inequalities we have the following:

$$X_{n+1} = S(A + L \circledast X_n) \geqq S(X \oplus 0) = X.$$

And the result follows; see [3] for a similar proof.

In the proof of the next theorem, we use the fact that we may choose two operators C and D such that the following set does not have a supremum:

$$\{X \mid X \geqq 0, X \leqq C, X \leqq D\}.$$

For 2×2 matrices the following C and D provide a counterexample.

$$C = \begin{pmatrix} 14 & 0 \\ 0 & 14 \end{pmatrix}. \quad D = \begin{pmatrix} 28 & 0 \\ 0 & 7 \end{pmatrix}.$$

We have that

$$Y = \begin{pmatrix} 14 & 0 \\ 0 & 7 \end{pmatrix} \quad \text{and} \quad Z = \begin{pmatrix} 12 & 4 \\ 4 & 6 \end{pmatrix}$$

are both maximal points in the set $\{X \mid X \geqq 0, X \leqq C, X \leqq D\}$, but Y and Z are not comparable, and therefore there is no maximum. □

THEOREM 8. *Let $K = K^*$. Then $M = \{X \geqq 0 \mid A + K \circledast X \geqq 0\}$ has a supremum for all $A \geqq 0$ if and only if K has exactly one negative eigenvalue.*

Proof. Since K is Hermitian, we find an invertible Q such that $Q^*KQ = \operatorname{diag}(\varepsilon_1, \cdots, \varepsilon_n)$, where each ε_i is $+1$, -1, or 0. The number of $+1$'s is the number of positive eigenvalues of K, the number of -1's is the number of negative eigenvalues of K, and the number of 0's is the nullity of K. Since Q is invertible, so is $(Q \circledast I)$ and therefore $A + K \circledast X$ is positive if and only if $[Q \circledast I]^* \{A + K \circledast X\} [Q \circledast I]$ is positive. But this expression may be rewritten to obtain $A' + K' \circledast X \geqq 0$, where $A' = (Q \circledast I)^* A (Q \circledast I)$ and $K' = Q^*KQ$. Here we have used the facts that $(E \circledast F)^* = E^* \circledast F^*$ and $(E \circledast F)(G \circledast H) = EG \circledast FH$. (See, for example, Bellman [10].)

Thus we have reduced the theorem to the case where K is a diagonal matrix whose diagonal elements are $+1$, -1, or 0. Rather than using K' and A' below, we revert to the nonprime notation. The proof involves three cases.

Case I. If K has no diagonal entries equal to -1, then $\{X: A + K \circledast X \geqq 0\}$ is unbounded, so the supremum does not exist.

Case II. If K has exactly one negative diagonal element, then by reordering we may assume $k_{11} = -1$. If $k_{ii} = 0$ for $i > 1$, then $\sup \{X \geqq 0 \mid A + K \circledast X \geqq 0\}$ is the shorted operator, and therefore the supremum exists. If some $k_{ii} = 1$, $i > 1$, then Theorem 7 applies and the supremum exists.

Case III. If more than one diagonal entry of K is negative, we may renumber and assume $K = \operatorname{diag}(-1, -1, \cdots, -1, 1, 1, \cdots, 1, 0, \cdots, 0)$ so that

$$k_{ii} = -1 \qquad i = 1, \cdots, s$$

$$k_{ii} = +1 \qquad i = s+1, \cdots, t$$

$$k_{ii} = 0 \qquad i = t+1, \cdots, n.$$

Let C and D be two positive operators, as discussed above, such that

$$\sup \{X \mid X \geqq 0, X \leqq C, X \leqq D\}$$

does not exist. Then set $A = \operatorname{diag}(A_1, A_2, \cdots, A_n)$, where $A_1 = C$; $A_2 = D$; $A_i = C + D$, $i = 3, \cdots, s$; and $A_i = 0$, $i > s$. The $\sup \{X \geqq 0 \mid A + K \circledast X \geqq 0\}$ cannot exist, since it would be $\sup \{X: X \geqq 0, X \leqq C, X \leqq D\}$. This completes the proof. □

Theorem 8 shows that the only interesting case occurs when K has exactly one negative eigenvalue. In this case we may define a function ϕ as follows:

$$\phi(A; K) = \sup \{X \geqq 0 A + K \circledast X \geqq 0\}.$$

THEOREM 9. *Let $A \geqq 0$, $K = K^*$ with exactly one negative eigenvalue, and let ϕ be defined as above. Then the following hold*:

(a) *ϕ is monotone on A and K*, i.e., *if $A \leqq B$ and $K \leqq L$ with L having exactly one negative eigenvalue, then $\phi(A; K) \leqq \phi(B; L)$.*

(b) *Let A_n be a monotonically decreasing sequence converging (strongly) to A. Then*

$$\lim_{n \to \infty} \phi(A_n; K) = \phi(A; K).$$

Proof. Part (a) follows from the fact that if $A \leqq B$ and $K \leqq L$, then

$$A + K \circledast X \leqq B + L \circledast X$$

and thus

$$\{X \geqq 0 \mid A + K \circledast X \geqq 0\} \subseteq \{X \geqq 0 \mid B + L \circledast X \geqq 0\}.$$

The proof follows because when the set of allowable X's increases, the supremum cannot decrease. Part (b) follows from the proof of Theorem 7 as the approximations X_n are monotonically decreasing and continuous under monotone decreasing limits.

We are only interested in the case of exactly one negative eigenvalue of K, in this case the maximum X falls into one of the following categories:

Case A: A short of A when the rank of $K = 1$;

Case B: A cascade limit of A when the rank of $K = 2$;

Case C: The X_∞ of Theorem 8 when the rank of $K \geqq 3$.

Special cases of Cases A and B include Propositions 2–6. In the following we let K_n denote the matrix K that shows that Proposition n follows from Theorem 8.

$$K_2 = K_4 = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \text{the eigenvalues are } -1, 0 \text{ and the rank is 1.}$$

$$K_3 = K_6 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \text{the eigenvalues are } +1, -1 \text{ and the rank is 2.}$$

$$K_5 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{the eigenvalues are } +1, -1 \text{ and the rank is 2.}$$

As an example of operation included in Case C, we consider the following scalar problem. Assume a, b, c, d, e, and f are real and that

$$\begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \text{ is positive.}$$

One can show that the

$$\sup\left\{x \geqq 0 \,\middle|\, \begin{pmatrix} a-x & b & c \\ b & d+x & e \\ c & e & f+x \end{pmatrix} \geqq 0\right\}$$

is a root of the following cubic equation, which is obtained by setting the determinant of the constraint matrix equal to zero.

$$x^3+(d+f-a)x^2+(df-e^2+b^2+c^2-ad-af)x$$
$$+b^2f-2bce+c^2f-adf+ae^2=0.$$

The K matrix associated with this suprema problem has eigenvalues of -1, 1, and 1 and is of rank 3. So if we consider the operator version of this problem, we will obtain a new operator function. Possibly the analysis in [9] could be applied to this new situation, and it could be shown that the solution of the suprema problem satisfies an operator version of the cubic equation. We remind the reader that in [9] the solution of a suprema problem is shown to satisfy an operator Riccati equation.

REFERENCES

[1] W. N. ANDERSON, *Shorted operators*, SIAM J. Appl. Math., 20, (1970), pp. 520–525.

[2] W. N. ANDERSON, T. D. MORLEY, AND G. E. TRAPP, *A characterization of parallel subtraction*, Proc. Nat. Acad. Sci. USA, 76 (1979), pp. 3599–3601.

[3] ———, *Positive solutions of* $X = A - B^*X^{-1}B$, Linear Algebra Appl., to appear.

[4] ———, *Cascade addition and subtraction of matrices*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 609–629.

[5] W. N. ANDERSON, D. F. REYNOLDS, AND G. E. TRAPP, *Cascade addition of matrices*, Proc. West Virginia Acad. Sci., 46 (1974), pp. 185–192.

[6] W. N. ANDERSON AND G. E. TRAPP, *Shorted operators* II, SIAM J. Appl. Math., 28 (1975), pp. 60–71.

[7] T. ANDO, *Topics in operator inequalities*, Lecture Notes, Sapporo Japan, 1978.

[8] ———, *Limit of cascade iteration of matrices*, Numer. Funct. Anal. Optim., 2 (1980), pp. 579–589.

[9] T. ANDO, J. BUNCE, AND G. E. TRAPP, *An alternate variational characterization of matrix Riccati equation solutions*, Preprint.

[10] R. BELLMAN, *An Introduction to Matrix Analysis*, Second edition, Academic Press, New York, 1970.

[11] J. I. Fujii, *On geometric and harmonic means of positive operators*, Math. Japan., 24 (1979), pp. 203–207.

[12] M. G. Krein, *The theory of self-adjoint extensions of semi-bounded Hermitian transformations and its applications*, Mat. Sb., 20 (1947), pp. 431–495; 21 (1947), pp. 365–404.

[13] G. E. TRAPP, *The Riccati equation and the geometric mean*, Contemp. Math., 47 (1985), pp. 437–445.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 213–217, April 1990

005

ALGEBRAIC POLAR DECOMPOSITION*

IRVING KAPLANSKY†

Abstract. Choudhury and Horn made a conjecture concerning conditions for a complex matrix to admit a decomposition as a product of an orthogonal matrix and a symmetric matrix. This conjecture, in a stronger form, is confirmed.

Key words. complex orthogonal, complex symmetric, polar decomposition

AMS(MOS) subject classifications. 15A23, 15A57

1. Introduction. Let A be a complex square matrix. It is classical that A can be written as the product of a unitary matrix and a positive semidefinite Hermitian matrix; the Hermitian part is always unique and the unitary part is unique if A is invertible. In [1] Choudhury and Horn studied an algebraic variant. In this variant one seeks to write $A = QS$, with Q complex orthogonal (rather than unitary) and S complex symmetric (rather than Hermitian). There appears to be no reasonable way to restrict Q or S or both so as to make the decomposition unique, and so we forget about uniqueness. However, existence of the decomposition merits scrutiny. In fact, the decomposition is not always possible, as the matrix

$$\begin{pmatrix} 1 & i \\ 0 & 0 \end{pmatrix}$$

shows. So it is natural to impose conditions. If $A = QS$, then $A'A = SQ'.QS = S^2$ and $AA' = QS.SQ' = QS^2Q^{-1}$. Thus two necessary conditions are visible: similarity of $A'A$ and AA' and the possession by $A'A$ of a square root. On page 225 of [1] it is conjectured that these two conditions are sufficient. The conjecture is true, and, moreover, the square root condition is redundant.

In view of the algebraic nature of the investigation, it is to be expected that any algebraically closed field of characteristic $\neq 2$ is acceptable. The case of characteristic 2 is indeed different and will not be examined in this paper.

There is one more note before the formal statement of the theorem. The hypothesis that $A'A$ is similar to AA' will be weakened to the hypothesis that $(A'A)^m$ and $(AA')^m$ have the same rank for all m. This is not being done for the sake of generalization, but rather because the rank condition is trivially inherited by the direct summands that we shall encounter below. In any event the generalization is nominal, for in the nilpotent case (the only case that matters) one can see a priori that the rank condition implies similarity.

THEOREM. *Let A be a square matrix over an algebraically closed field of characteristic $\neq 2$. Then A can be written QS, with Q orthogonal and S symmetric, if and only if $(A'A)^m$ and $(AA')^m$ have the same rank for every m.*

2. Three lemmas. The lemmas in this section will facilitate the proof of the theorem.

As far as possible, we shall operate in a basis-free fashion. This calls for the following setup. We assume given an algebraically closed field k of characteristic $\neq 2$ and two linear spaces V and W of the same dimension over k, each carrying a nonsingular symmetric

* Received by the editors November 14, 1988; accepted for publication (in revised form) March 15, 1989.
† Mathematical Sciences Research Institute, 1000 Centennial Drive, Berkeley, California 94720. This work was supported by National Science Foundation grant DMS 8505550.

bilinear form. For both V and W we use the notation $(\ ,\ )$ for the form. We are given a linear transformation A from V to W. It has a transpose A' that maps W into V. We have $(Av, w) = (v, A'w)$ for all v in V, w in W. Our objective is to write $A = QS$, where S is symmetric on V and Q maps V orthogonally into W, that is, $(Qv_1, Qv_2) = (v_1, v_2)$ for all v_1, v_2 in V. When this is so, we have $A' = SQ'$, $A'A = S^2$; furthermore, the equation $A = QS$ implies that A and S have the same null space. So a necessary condition for $A = QS$ to be achievable is that $A'A$ has a symmetric square root with the same null space as A. This condition is also sufficient. This essentially appears on p. 220 of [1], but for completeness a proof is included.

LEMMA 1. *Let A, V, and W be as above. Suppose that $A'A$ has a symmetric square root S with the same null space as A. Then there exists an orthogonal linear transformation Q from V to W satisfying $A = QS$.*

Proof. We define Q first from the range of S to the range of A by setting $QSx = Ax$. From the equality of the null spaces of S and A we first see that this is a valid definition and then that the mapping is one-to-one. We have

$$(Sx, Sx) = (S^2x, x) = (A'Ax, x) = (Ax, Ax).$$

So Q preserves the bilinear form as far as Q is thus far defined. By Witt's theorem, Q can be extended to an orthogonal mapping from all of V onto all of W. □

LEMMA 2. *Over an algebraically closed field of characteristic $\neq 2$, let V be a $(2m+1)$-dimensional linear space with a nonsingular symmetric bilinear form. Let T be a nilpotent symmetric linear transformation on V with elementary divisors of degrees m and $m+1$. Then T has a symmetric square root with a one-dimensional null space equal to T^mV.*

Proof. It is evident that T has a square root. By [1, Thm. 4] T has a symmetric square root S (while in [1] the ground field is the field of complex numbers, the proof there is valid in this more general context). Necessarily, S has a single elementary divisor of degree $2m+1$ and so it has a one-dimensional null space equal to $S^{2m}V = T^mV$. □

LEMMA 3. *Over an algebraically closed field of characteristic $\neq 2$ let V be a $2m$-dimensional linear space with a nonsingular symmetric bilinear form. Let T be a nilpotent symmetric linear transformation on V with elementary divisors of degrees m and m. Let u be a vector in V with $T^{m-1}u \neq 0$, $(T^{m-1}u, u) = 0$. Then T has a symmetric square root with a one-dimensional null space spanned by $T^{m-1}u$.*

Proof. Take a basis $x_1, \cdots, x_m, y_1, \cdots, y_m$ of V with each $(x_i, y_i) = 1$ and the form vanishing on all other pairs of basis elements. Let U be the linear transformation given by

$$Ux_i = x_{i+1} (i = 1, \cdots, m-1), \qquad Ux_m = 0,$$

$$Uy_i = y_{i-1} (i = 2, \cdots, m), \qquad Uy_1 = 0.$$

Note that U is nilpotent and that it has the same elementary divisors as T: thus U and T are similar. One readily checks that U is symmetric. From this, one knows that U and T are orthogonally similar. So V admits a basis of the same kind relative to T, and we shall use the same notation x_i, y_i for this basis. $T^{m-1}V$ is two-dimensional, spanned by x_m and y_1. So $T^{m-1}u$ has the form $ax_m + by_1$. In computing $(T^{m-1}u, u)$, the only portions of u that make a contribution are the terms in x_1 and y_m, and this part of u must have the form $ax_1 + by_m$. Since $(ax_m + by_1, ax_1 + by_m) = 2ab$, we must have $a = 0$ or $b = 0$ in order for $(T^{m-1}u, u)$ to vanish. Thus $T^{m-1}u$ is a scalar multiple of x_m

or y_1. By symmetry we can suppose that $T^{m-1}u$ is a scalar multiple of x_m. We now define S as having the single Jordan block

$$y_m, x_1, y_{m-1}, x_2, \cdots, y_2, x_{m-1}, y_1, x_m,$$

that is, S sends each of these vectors to the next and sends x_m to 0. It is routine to check that S is symmetric. We have $S^2 = T$, and the null space of S is spanned by x_m, and hence by $T^{m-1}u$. □

3. Reduction to the case where $A'A$ is nilpotent. We return to A, V, and W as in § 2. In this section we shall reduce the problem of achieving an algebraic polar decomposition of A to the case where $A'A$ is nilpotent.

It is standard that V has a unique direct sum decomposition $V = V_1 + V_2$ with each V_i invariant under $A'A$, $A'A$ invertible on V_1, and $A'A$ nilpotent on V_2. Write $W = W_1 + W_2$ for the analogous decomposition of W relative to AA'. There are three things to be checked:

(a) A maps V_2 into W_2 and A' maps W_2 into V_2.

(b) A maps V_1 one-to-one onto W_1 and A' maps W_1 one-to-one onto V_1.

(c) The two decompositions are orthogonal relative to the bilinear forms.

By symmetry only half of each of these statements needs proof. The following characterization of the summands V_1 and V_2 will be used: V_2 consists of the elements annihilated by some power of $A'A$, and V_1 consists of the elements lying in the range of large powers of $A'A$ (see p. 113 of [2] for details concerning this).

(a) If $x \in V_2$, then $(A'A)^n x = 0$ for some n, $(AA')^n Ax = 0$, $Ax \in W_2$.

(b) Suppose that $x \in V_1$ and $Ax = 0$. Then $A'Ax = 0$ and $x = 0$, since $A'A$ is invertible on V_1. Thus A induces a one-to-one map of V_1 into W_1. In particular, $\dim W_1 \geqq \dim V_1$. By symmetry the opposite inequality also holds. Thus $\dim V_1 = \dim W_1$ and it follows that the map A of V_1 into W_1 is onto as well as one-to-one.

(c) To establish the orthogonality of v_1 and v_2 for $v_i \in V_i$, we take n large enough so that $(A'A)^n v_2 = 0$ and v_1 is in the range of $(A'A)^n$, say $v_1 = (A'A)^n v'$. Then $(v_1, v_2) = ((A'A)^n v', v_2) = (v', (A'A)^n v_2) = 0$.

We begin the proof of the theorem. For an invertible linear transformation the QS decomposition is known. Hence our business is finished regarding V_1 and W_1. It remains to treat the restrictions of A and A' to V_2 and W_2, but first we have to observe that the hypothesis of the theorem is inherited. Since rank is additive on direct sums, we see that, for every m, $(A'A)^m$ on V_2 and $(AA')^m$ on W_2 have the same rank. We change notation, replacing V_2 and W_2 by V and W. Henceforth $A'A$ and AA' are nilpotent.

4. The case where $A'A$ is nilpotent. The procedure will be to detach a well-behaved direct summand. The idea is not new; for instance, it appears in essence in [3]. In the present context appropriate modifications are needed to cope with two vector spaces.

We form the longest product

$$\cdots A'AA'A$$

that is nonzero and call it B. Let r be the number of terms in this product. The parity of r makes a difference.

***r* even.** $B = (A'A)\cdots(A'A)$. It cannot be the case that (Bx, x) vanishes for all x in V, for then $B = 0$ by polarization (this uses characteristic $\neq 2$). Choose x in V with $(Bx, x) \neq 0$. If $C = (AA')^{r/2}$, then B and C have the same rank by hypothesis and in

particular $C \neq 0$. Thus we have y in W with $(Cy, y) \neq 0$. We line up the following $r+1$ elements of V:

$$x_0 = x, \qquad x_1 = A'y, \qquad x_2 = A'Ax, \qquad x_3 = A'AA'y, \cdots$$

$$x_{r-1} = A'(AA')^{\frac{r}{2}-1}y, \qquad x_r = (A'A)^{\frac{r}{2}}x = Bx.$$

For $i + j = r$ we have $(x_i, x_j) = (Bx, x)$ for i even and (Cy, y) for i odd. Thus (x_i, x_j) is nonzero for $i + j = r$. Because of the maximal property of r, (x_i, x_j) vanishes for $i + j > r$. So the matrix of elements (x_i, x_j) has nonzero elements on the antidiagonal (the diagonal running from the upper right corner to the lower left corner) and 0's to the right; it is an invertible matrix. As is well known, this implies that the elements $x_0, \cdots, x_r$ are linearly independent. Furthermore, the bilinear form is nonsingular on the subspace X, which they span [4, Thm. 1 on p. 4], and X is an orthogonal direct summand of V [4, Thm. 2 on p. 6]. The construction is now repeated on the other side of the ledger, producing elements $y_0 = y, y_1 = Ax, y_2 = AA'y, \cdots, y_r = Cy$, which form a basis of an orthogonal direct summand Y of W. A sends x_i into y_{i+1} for $i = 0, \cdots, r-1$ and annihilates x_r. The null space of A is one-dimensional, spanned by x_r. $A'A$ is a symmetric linear transformation on X admitting the two Jordan blocks $x_0, x_2, x_4, \cdots, x_r$ and $x_1, x_3, x_5, \cdots, x_{r-1}$. The range of $(A'A)^{r/2}$ on X is spanned by x_r. Lemma 2 applies with $T = A'A$ and $m = r/2$. The symmetric square root of $A'A$ thus obtained has $(A'A)^m X$ as its null space, that is, its null space is spanned by x_r and thus coincides with the null space of A. This gets us ready to apply Lemma 1 to A and A', restricted to X and Y, so that the QS decomposition is achieved there. Let X' and Y' be the orthogonal complements of X and Y in V and W. Then A sends X' into Y', A' sends Y' into X', and A and A' remain transposes when restricted to X' and Y'. The proof of the theorem is finished by induction as soon as we observe that the hypothesis of the theorem is inherited; this is seen by additivity of the rank just as at the end of the preceding section.

r odd. $B = A(A'A)\cdots(A'A)$. The procedure is similar but has to be changed a little, since B now sends V into W. We select x in V and y in W simultaneously to satisfy $(Bx, y) \neq 0$. The elements x_i and y_i are picked in essentially the same way as before:

$$x_0 = x, \quad x_1 = A'y, \quad x_2 = A'Ax, \cdots, \quad x_{r-1} = (A'A)^{(r-1)/2}x,$$

$$x_r = A'(AA')^{\frac{(r-1)}{2}}y = By,$$

$$y_0 = y, \quad y_1 = Ax, \quad y_2 = AA'y, \cdots, \quad y_{r-1} = (AA')^{(r-1)/2}y,$$

$$y_r = A(A'A)^{\frac{(r-1)}{2}}x = B'x.$$

The passages to the subspaces X and Y go as before. In achieving the polar decomposition of A restricted to X, we use Lemma 3 in place of Lemma 2. The details are as follows. The null space of A is spanned by x_r. $T = A'A$ admits the two Jordan blocks $x_0, x_2, \cdots, x_{r-1}$ and $x_1, x_3, \cdots, x_r$. With the choices $u = x_1$ and $m = (r+1)/2$ the hypotheses of Lemma 3 are in place. This makes Lemma 1 applicable and achieves the polar decomposition of A restricted to X.

Summary. The desired polar decomposition of the original linear transformation A has been achieved in a succession of steps. First, the portion where $A'A$ is invertible

was detached in § 3. Then the summand where $A'A$ is nilpotent was decomposed into a number of pieces, and on each of these, special circumstances made $A = QS$ possible, with the aid of suitable lemmas from § 2. In this way the proof of the theorem is complete.

REFERENCES

[1] D. CHOUDHURY AND R. A. HORN, *A complex orthogonal-symmetric analog of the polar decomposition*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 218–225.

[2] P. R. HALMOS, *Finite-Dimensional Vector Spaces*, Springer-Verlag, Berlin, New York, 1974.

[3] I. KAPLANSKY, *Orthogonal similarity in infinite-dimensional spaces*, Proc. Amer. Math. Soc., 3 (1952), pp. 16–25.

[4] ———, *Linear Algebra and Geometry—A Second Course*, Chelsea, New York, 1974.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 218–238, April 1990

006

THE LAPLACIAN SPECTRUM OF A GRAPH*

ROBERT GRONE†, RUSSELL MERRIS‡, AND V. S. SUNDER§

Abstract. Let G be a graph. The Laplacian matrix $L(G) = D(G) - A(G)$ is the difference of the diagonal matrix of vertex degrees and the 0-1 adjacency matrix. Various aspects of the spectrum of $L(G)$ are investigated. Particular attention is given to multiplicities of integer eigenvalues and to the effect on the spectrum of various modifications of G.

Key words. tree(s), eigenvalue(s), spectra, graph(s)

AMS(MOS) subject classifications. primary 05C05, 05C50; secondary 05C10, 05C25, 05C40

1. Introduction. Let $G = (V, E)$ be a graph with vertex set $V = \{v_1, v_2, \cdots, v_n\}$ and edge set $E = \{e_1, e_2, \cdots, e_m\}$. For each edge $e_j = \{v_i, v_k\}$, choose one of v_i, v_k to be the positive end of e_j and the other to be the negative end. We refer to this procedure by saying G has been given an *orientation*. The vertex-edge *incidence matrix* afforded by an orientation of G is the n-by-m matrix $Q = Q(G) = (q_{ij})$, where

$$q_{ij} = \begin{cases} +1, & \text{if } v_i \text{ is the positive end of } e_j, \\ -1, & \text{if it is the negative end,} \\ 0, & \text{otherwise.} \end{cases}$$

It turns out that $L(G) = QQ^t$ is independent of the orientation. In fact, $L(G) = D(G) - A(G)$, where $D(G)$ is the diagonal matrix of vertex degrees and $A(G)$ is the (symmetric) 0-1 adjacency matrix. Forsman [9] and Gutman [11] have shown how the connection between $L(G)$ and $K(G) = Q^tQ$ simultaneously explain the statistical and the dynamic properties of flexible branched polymer molecules. Indeed, since $L(G)$ and $K(G)$ share the same nonzero eigenvalues, it follows that for bipartite graphs the smallest eigenvalue of $A(G^*) \geqq -2$, where G^* is the line graph of G. This observation, first made by Hoffman, has led to a connection with the theory of root systems [2], [3]. Eichinger [5] has shown how the spectrum of $L(G)$ may be used to calculate the radius of gyration of a Gaussian molecule. Mohar [13] argues that, because of its importance in various physical and chemical theories, the spectrum of $L(G)$ is more natural and important than the more widely studied adjacency spectrum. In [1], Bien uses the smallest positive eigenvalue of $L(G)$ to estimate the "magnifying coefficient" of G.

It seems that $L(G)$ first occurred in the celebrated **Matrix-Tree Theorem:** If L_{ij} is the submatrix of $L(G)$ obtained by deleting its ith row and jth column, then $(-1)^{i+j} \det (L_{ij})$ is the number of different spanning trees in G. Since this result is attributed to G. Kirchhoff, $L(G)$ is sometimes called a *Kirchhoff matrix*. It is also known as a *matrix of admittance* (admittance = conductivity). Following [7], we will refer to $L(G)$ as a *Laplacian* matrix because it is a discrete analogue of the Laplace differential operator.

* Received by the editors November 23, 1987; accepted for publication (in revised form) May 22, 1989. The work of all three authors was supported by Office of Naval Research contract 85-K-0335.

† Department of Mathematical Sciences, San Diego State University, San Diego, California 92182.

‡ Department of Mathematics and Computer Science, California State University, Hayward, California 94542 (CSUH!MERRIS@LLL-WINKEN.LLNL.GOV).

§ Indian Statistical Institute, Bangalore 560059, India.

We have suppressed the dependence of $L(G)$ on the ordering of V because our primary interest is with the characteristic polynomial $c_G(x) = \det(xI - L(G))$.

Example 1.1. Let $G = C_6$, the simple circuit on six vertices. Then

$$c_G(x) = x(x-1)^2(x-3)^2(x-4)$$
$$= x^6 - 12x^5 + \cdots - 36x.$$

Of course, 12 is the sum of the vertex degrees and 36 is the sum of the six principal minors of $L(G)$ of order five. We can state each of these facts in another way. The sum of the vertex degrees is twice the number of edges. The sum of the minors is ($n =$) six times the number of spanning trees. Similar statements are available for the other coefficients [4, p. 38].

Example 1.2. Let $G = *_n$, the "star," i.e., $G = K_{1,n-1}$, the complete bipartite graph with $n - 1$ pendant (degree 1) vertices and one vertex of degree $n - 1$. Then $c_G(x) = x(x-n)(x-1)^{n-2}$. If the central vertex is listed last, then $(-1, -1, \cdots, n-1)$ is an eigenvector of $L(G)$ corresponding to n, while

$$\{(1,-1,0,\cdots,0),(0,1,-1,0,\cdots,0),\cdots,(0,0,\cdots,0,1,-1,0)\}$$

is a set of $n - 2$ linearly independent eigenvectors corresponding to one.

Denote the eigenvalues of $L(G)$ by $\lambda_1 \geqq \cdots \geqq \lambda_{n-1} \geqq 0 = \lambda_n$. From the Matrix-Tree Theorem (for example) we may deduce that $\lambda_{n-1} > 0$ if and only if G is connected. (In particular, $K(G)$ is nonsingular if and only if G is a tree.) Fiedler has called λ_{n-1} the *algebraic connectivity* of G [7], denoting it by $a(G)$.

2. Preliminary results. A vertex of degree one is called a pendant vertex. Denote by $p(G)$ the number of pendant vertices of G. A vertex is *quasipendant* if it is adjacent to a pendant vertex. Denote by $q(G)$ the number of quasipendant vertices of G. If T is a tree, it is known [4, p. 258] that

$$p(T) - q(T) \leqq \eta \leqq p(T) - 1, \tag{1}$$

where η is the multiplicity of zero as an eigenvalue of $A(T)$.

Denote by $m_G(\lambda)$ the multiplicity of λ as an eigenvalue of $L(G)$. Incidental to her work on permanental polynomials, Faria observed that

$$p(G) - q(G) \leqq m_G(1), \tag{2}$$

for any graph G [6].

THEOREM 2.1. *Suppose T is a tree on n vertices. If $\lambda > 1$ is an integer eigenvalue of $L(T)$ with corresponding eigenvector u, then*

(i) $\lambda | n$ (*i.e.*, λ *exactly divides* n),

(ii) $m_T(\lambda) = 1$,

(iii) *no coordinate of u is zero.*

Theorem 2.1 can fail totally for graphs that are not trees. (See Example 1.1.)

Proof. The characteristic polynomial of $L(T)$ is $xf(x)$, where $f(x)$ is an integer polynomial. Since T is a tree, $f(0) = n$ (as in Example 1.1). This proves (i). If $L(T)$ had two linearly independent eigenvectors corresponding to λ, we could produce a third eigenvector with zero in any prescribed coordinate. Hence, (iii) implies (ii).

Suppose u is an eigenvector of $L(T)$ afforded by λ. If some coordinate of u is zero, we may assume it is the last one, corresponding to vertex v_n. With $d = d_n$, the degree of v_n, $L(T)$ takes the form

$$L(T) = \begin{pmatrix} B_1 & 0 \cdots 0 & * \\ 0 & B_2 \cdots 0 & * \\ & \cdots & \\ 0 & 0 \cdots B_d & * \\ * & * \cdots * & d \end{pmatrix}, \tag{3}$$

where $B_1, B_2, \cdots, B_d$ are the principal submatrices of $L(T)$ corresponding, respectively, to the branches $T_1, T_2, \cdots, T_d$ of T at v_n. If u is partitioned conformally as $u = (u_1, u_2, \cdots, u_d, 0)$, then $uL(T) = \lambda u$ implies $u_i B_i = \lambda u_i$, $1 \leqq i \leqq d$. Since at least one of these u_i's must be nonzero, λ is an eigenvalue of some B_i. We may assume it is B_1. Note that B_1 is not quite $L(T_1)$. One of its main diagonal entries is too large, the one corresponding to the vertex of T_1 that is adjacent (in T) to v_n. If we assume this vertex is v_1, then $B_1 = L(T_1) + E_{11}$, where E_{11} is the matrix whose only nonzero entry is a one in position (1, 1). But then $\det B_1 = \det L(T_1) + \det L_{11}$, where L_{11} is the submatrix of $L(T_1)$ obtained by eliminating its first row and column. Now, $\det L(T_1) = 0$ while, by the Matrix-Tree Theorem, $\det L_{11} = 1$. Thus, λ is an eigenvalue of the unimodular matrix B_1, a contradiction. □

It seems surprising that for trees, $m_T(1)$ can be arbitrarily large while $m_T(2)$ can be at most one. It turns out that, integer or not, the largest eigenvalue of any bipartite graph is simple. This is a consequence of the following elementary observation.

PROPOSITION 2.2. *Let G be a bipartite graph. Then $B(G) = D(G) + A(G)$ and $D(G) - A(G) = L(G)$ are unitarily similar; in particular, the maximum eigenvalue of $L(G)$ is simple provided G is connected.*

If $G = K_n$, the complete graph, then $\lambda_1 = n$ and $m_G(\lambda_1) = n - 1$, i.e., the result can fail if G is not bipartite.

Proof. Since G is bipartite, the vertex set can be partitioned into two subsets V_1 and V_2 so that no two vertices in V_i are adjacent, for $i = 1, 2$.

Let $U = (u_{ij})$ be the diagonal matrix with

$$u_{ii} = \begin{cases} 1, & \text{if } v_i \in V_1, \\ -1, & \text{if } v_i \in V_2. \end{cases}$$

It is simple to verify that $UA(G)U^{-1} = -A(G)$ and that U commutes with $D(G)$. In case G is connected, the matrix $D(G) + A(G)$ is a nonnegative irreducible matrix and the second assertion is a consequence of the Perron–Frobenius theory. □

We now show that the upper-bound in (1) is a uniform upper bound on the multiplicity of any eigenvalue of $L(T)$.

THEOREM 2.3. *Let λ be an eigenvalue of $L(T)$ for some tree T on $n \geqq 2$ vertices. Then $m_T(\lambda) \leqq p(T) - 1$.*

As we saw in Example 1.2, equality can occur. On the other hand, if G is not a tree, it may have no pendant vertices.

Proof. Suppose v_n is a pendant vertex of T. We may assume v_{n-1} is the quasipendant of T adjacent to v_n. Let $u = (u_1, \cdots, u_n)$ be an eigenvector of $L(T)$ corresponding to λ. Then $(1 - \lambda)u_n = u_{n-1}$. Consider the possibility that $u_n = 0$. In this case, $u_{n-1} = 0$ and, moreover, $u' = (u_1, \cdots, u_{n-1})$ is an eigenvector of $L(T')$ corresponding to λ, where T' is the tree obtained from T by deleting v_n from V and $\{v_{n-1}, v_n\}$ from E. It follows by induction that u cannot be zero in all coordinates corresponding to pendant

vertices, or even in all but one of them! On the other hand, if the eigenspace W of $L(T)$ corresponding to λ were to have dimension greater than $p(T) - 1$, it would be possible to find a nonzero vector $w \in W$ that is zero on all but (at most) one of its coordinates corresponding to pendant vertices. □

In order to discuss the next result, the following notation will be convenient. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs with $V_1 \cap V_2 = \varnothing$. A *connected sum* of G_1 and G_2 is any graph $G = (V, E)$ where $V = V_1 \cup V_2$, and E differs from $E_1 \cup E_2$ by the addition of a single edge joining some (arbitrary) vertex of V_1 to some vertex of V_2. It will be useful to write $G = G_1 \# G_2$. Note that "#" is not a binary operation on graphs because it is not well defined. If $n_1 = o(V_1)$, the cardinality of V_1, and $n_2 = o(V_2)$, then $G_1 \# G_2$ may represent any of $n_1 n_2$ different graphs. In general, of course, some of these graphs will be isomorphic as the following example shows.

Example 2.4. Denote by P_n the path on n vertices (of length $n - 1$). If $G_1 = P_2$ and $G_3 = P_3$, then $G_1 \# G_2$ is isomorphic either to P_5 or the graph in Fig. 1.

THEOREM 2.5. *Let G be a (nonempty) graph on n vertices. Let $H = G \# *_k$ be a connected sum of G with the star on $k > 1$ vertices. Then $m_G(k) = m_H(k)$.*

Proof. Assume the vertices have been numbered so that $G \# *_k$ is obtained by joining the last vertex of G to the first vertex of $*_k$. If $L = L(G)$, $L_* = L(*_k)$, and $L_\# = L(H)$, then, with respect to the obvious ordering of vertices,

$$L_\# = (L \dotplus L_*) + A, \tag{4}$$

where $A = (a_{ij})$ is the $(n + k)$-by-$(n + k)$ matrix with

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \{(n,n),(n+1,n+1)\}, \\ -1 & \text{if } (i,j) \in \{(n,n+1),(n+1,n)\}, \\ 0 & \text{otherwise.} \end{cases}$$

From Example 1.2, we may choose an eigenvector w for L_*, corresponding to k, whose first component is $w_1 = 1$. We will use w to produce a linear bijection $u \to u_\#$ from $\ker(L - kI_n)$ onto $\ker(L_\# - kI_{n+k})$. For $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^k$, denote their juxtaposition by $x \oplus y \in \mathbb{R}^{n+k}$. Then for any $u \in \mathbb{R}^n$, define $u_\# = u \oplus u_n w$. Clearly, $u \to u_\#$ is linear and one-to-one. Since the nth and $(n + 1)$st coordinates of $u_\#$ are equal, $Au_\# = 0$ and, hence, $L_\# u_\# = Lu \oplus u_n L_* w$. But, then $u_\#$ is an eigenvector of $L_\#$ corresponding to k whenever u is an eigenvector of L corresponding to k. It remains to prove that every eigenvector of $L_\#$ corresponding to k is of the form $u_\#$ for some $u \in \ker(L - kI_n)$. Suppose $x \in \mathbb{R}^n$, $y \in \mathbb{R}^k$ and $x \oplus y$ is an eigenvector for $L_\#$ corresponding to k. We first assert that y is a multiple of w. This is seen by considering two cases.

Case i. The first vertex of $*_k$ (the one being connected to G by the new edge) is a pendant vertex. In this case, we may assume the vertices of $*_k$ so ordered that the kth

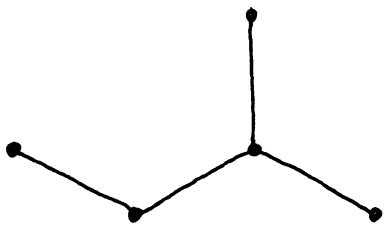


FIG. 1

vertex has degree $k-1$. Hence, $w=(1, 1, \cdots, 1, 1-k)$. (See Example 1.2.) We proceed to show that $y=y_1 w$. For $1<i<k$ (when $k>2$), we may conclude from the $(n+i)$th row of $L_{\#}(x \oplus y)=k(x \oplus y)$ that $y_k=(1-k)y_i$. Then, from the $(n+k)$th row, namely,

$$-y_1-y_2-\cdots-y_{k-1}+(k-1)y_k=ky_k,$$

it follows that $(1-k)y_1$ is also equal to y_k. In other words (since $k>1$), $y=y_1 w$.

Case ii. The first vertex of $*_k$ has degree $k-1$. In this case, $w_1=1$ while $w_2=\cdots=w_k=1/(1-k)$. (When $k=2$, the two cases coincide.) We use a similar argument to deduce that $(1-k)y_i=y_1$ for $i=2, \cdots, k$. Thus, $y=y_2 w$.

We have shown that the typical vector in $\ker(L_{\#}-kI_{n+k})$ is of the form $x \oplus cw$. It remains only to show that $x_n=c$ and that $x \in \ker(L-kI_n)$. Now, by comparing $(n+1)$st rows of

$$\begin{aligned} k(x \oplus cw) &= L_{\#}(x \oplus cw) \\ &= (Lx \oplus kcw)+(0, \cdots, 0, x_n-c) \oplus (-x_n+c, 0, \cdots, 0), \end{aligned}$$

we see that $kc=kc-x_n+c$. Finally, compare the first n rows to deduce that $Lx=kx$. □

Theorem 2.5 is useful as a reduction device. Suppose, for example, that T is a tree and we want to know whether or not two is an eigenvalue. Then we may *prune* off P_2's without changing the answer to our question.

Example 2.6. Let G be the graph in Fig. 2. Then we may write $G=G' \# *_2$ in a variety of ways. For any of these, $m_G(2)=m_{G'}(2)$. But then G' can be written as $G'' \# *_2$, also in several ways. Indeed, we may eventually prune off six copies of $*_2$. (See Fig. 3.) The result is that $m_G(2)=m_{C_4}(2)$. The characteristic polynomial for the square is $x(x-2)^2(x-4)$, so $m_G(2)=2$.

Example 2.7. As nice as the pruning process of Example 2.6 is, we eventually come to a "core" graph from which no P_2's may be pruned and for which it still may not be clear, even for trees, whether or not two is an eigenvalue. For the tree T in Fig. 4, $c_T(x)=x(x-1)^3(x-2)(x-5)(x^2-4x+1)^2$.

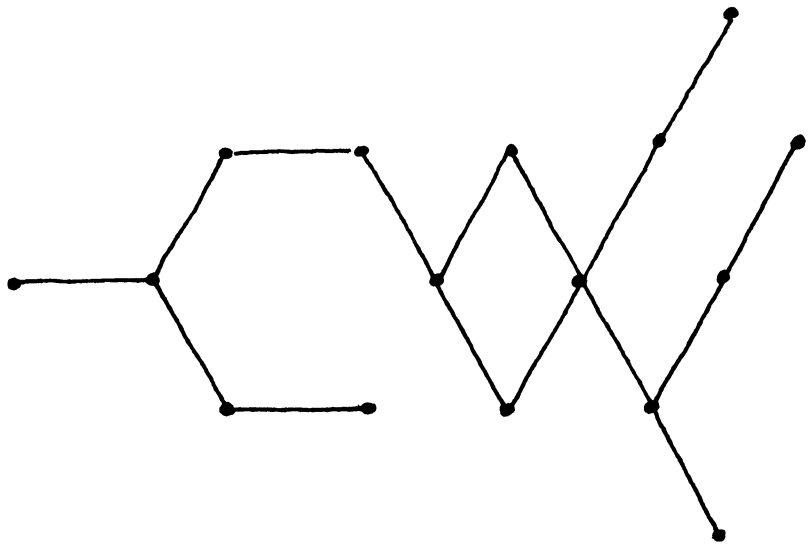


FIG. 2

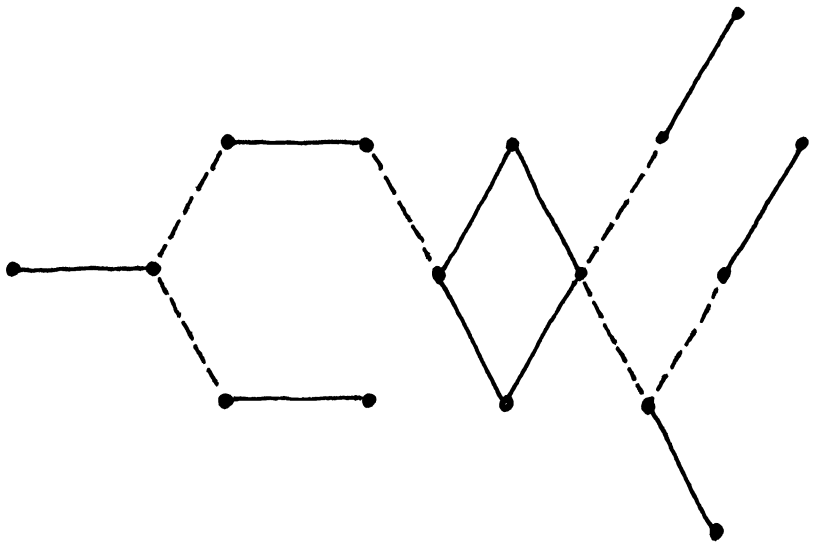


FIG. 3

COROLLARY 2.8. *Let $T = P_n$, the path on n vertices. Then*

(i) $$m_T(2) = \begin{cases} 1 & \text{if } 2 \mid n, \\ 0 & \text{otherwise.} \end{cases}$$

(ii) $$m_T(3) = \begin{cases} 1 & \text{if } 3 \mid n, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. We know from Theorem 2.1 that $m_T(k)$ is at most one when $k > 1$ is an integer and T is a tree. Since $P_2 = *_2$ and $P_3 = *_3$, the result follows from Theorem 2.5 and the pruning process of Example 2.6. □

Example 2.9. The graph in Fig. 4 is just one of a class of examples. If $k \geqq 2$ is an integer, we define a tree Z_k on $(2k - 1)(k + 1) + 1$ vertices as follows. Start with $k + 1$

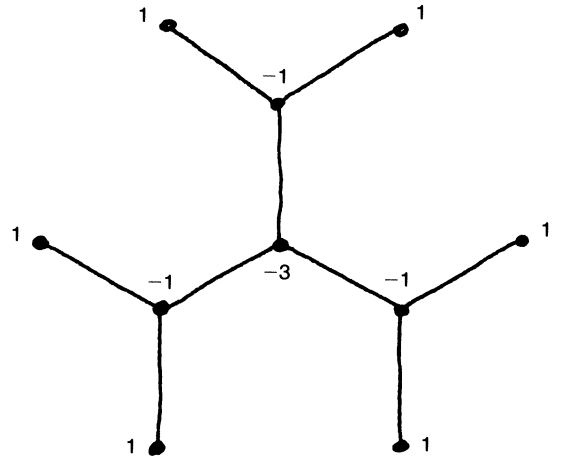


FIG. 4

copies of $*_{2k-1}$ and an additional vertex v. Then join the "center" of each star to v by an edge. It turns out that $m_{Z_k}(k) = 1$: that the multiplicity cannot be greater than one is assured by Theorem 2.1(ii). To see that the multiplicity is greater than zero, we may simply describe an eigenvector. The components of this eigenvector will be one in each of the $(2k-2)(k+1)$ coordinates corresponding to pendant vertices, $1-k$ in each of the $k+1$ coordinates corresponding to star centers, and $1-k^2$ in the coordinate corresponding to v. This explains the numbers in Fig. 4.

3. The multiplicity of $\lambda = 1$. We begin this section with an analogue of Theorem 2.5. We will be concerned with a slightly restricted version of the connected sum idea. By $G \vee P_3$ we mean (any) one of the graphs obtained from G and P_3 by joining some (arbitrary) vertex of G to a *pendant* vertex of P_3. (Of course, $P_3 = *_3$. We are using "$\vee$" here rather than "#" to indicate that it is now forbidden to join a vertex of G to the middle vertex of P_3. We will deal separately with this latter case in Proposition 3.14 below.)

THEOREM 3.1. *Let G be a (nonempty) graph on n vertices and suppose $H = G \vee P_3$. Then $m_G(1) = m_H(1)$.*

Proof. Let the second vertex of P_3 be the one of degree two. Then $w = (1, 0, -1)$ is an eigenvector of $L(P_3)$ corresponding to one. Number the vertices of G so that it is the last vertex that is joined to vertex one of P_3. Let L, L', and $\tilde{L}$ denote the Laplacian matrices of G, P_3, and H, respectively. Then, as in the proof of Theorem 2.5, $\tilde{L} = (L \dotplus L') + A$, where

$$A = 0_{n-1} \dotplus \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \dotplus 0_2.$$

Argue exactly as in Theorem 2.5 to show the map $u \to u \oplus u_n w$ is a linear injection of $\ker(L - I_n)$ into $\ker(\tilde{L} - I_{n+3})$. Conversely, if $x \oplus y$ is an eigenvector of $\tilde{L}$ corresponding to one, deduce that $y_2 = 0$ and that $y_3 = -y_1$. Proceed as in the proof of Theorem 2.5 to conclude that $y = x_n w$, so that $x \oplus y = x \oplus x_n w$ where $x \in \ker(L - I_n)$, as desired. □

Example 3.2. Let T be the tree in Fig. 5, with $k \geqq 2$. Then we may express T as $T' \vee P_3$ in a variety of ways; for any of these, $m_T(1) = m_{T'}(1)$. But, then $T' = T'' \vee P_3$, etc. Eventually, we see that $m_T(1) = m_S(1) = k - 1$, where $S = *_{k+1}$. It is instructive to compare this value with the Faria lower bound in (2), namely, $p(T) - q(T) = 0$. At the other extreme, the upper bound of Theorem 2.3 is $p(T) - 1 = k - 1$.

COROLLARY 3.3. *Let $T = P_n$. Then*

$$m_T(1) = \begin{cases} 1 & \text{if } 3 \mid n, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. By Theorem 3.1, it suffices to consider $n = 1$, 2, or 3. The characteristic polynomials for P_1, P_2, and P_3 are, respectively, x, $x(x-2)$, and $x(x-1)(x-3)$. □

Example 3.4. Since $P_3 = *_3$, pruning of a path of length three affects neither $m_G(3)$ (Theorem 2.5) nor $m_G(1)$ (Theorem 3.1). If G is the graph in Fig. 6, we may prune off 5 P_3's and obtain the hexagon of Example 1.1. Thus, $m_G(3) = m_G(1) = 2$.

Example 3.5. As in Example 2.7, one may prune off only so many P_3's, even for trees. If T is the tree in Fig. 7, then

$$c_T(x) = x(x-1)(x^2-3x+1)^2(x^2-7x+11)(x^3-6x^2+8x-1).$$

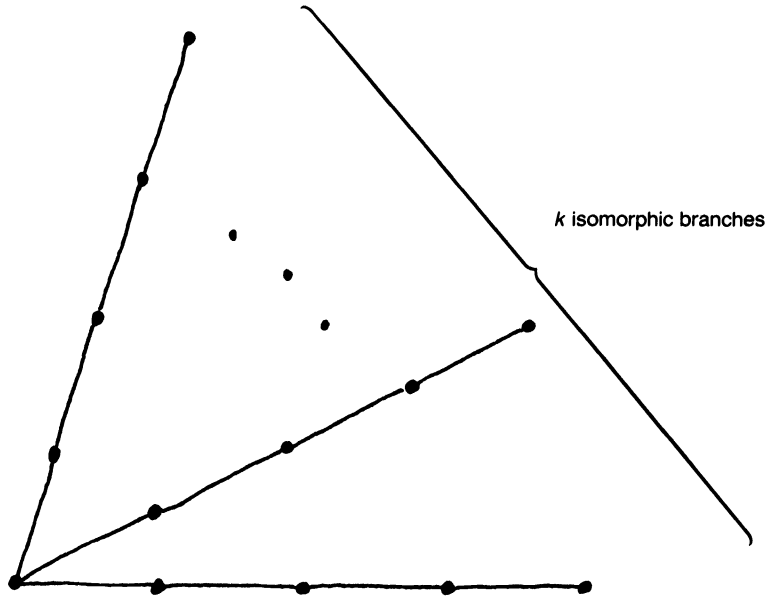


FIG. 5

We proceed now with a closer scrutiny of $m_G(1)$. Suppose G is a fixed but arbitrary graph on n vertices. Note that $u = (u_1, \cdots, u_n)$ is an eigenvector of $L(G)$ corresponding to $\lambda = 1$, if and only if

$$\sum_{\{v_i,v_j\} \in E} u_j = (d_i - 1)u_i, \qquad 1 \leqq i \leqq n. \tag{5}$$

(In particular, if $uL(G) = u$, then $u_t = 0$ for all quasipendant vertices v_t.)

In terms of eigenvectors, it is easy to explain why $m_G(1) \geqq p(G) - q(G)$, a difference that Faria refers to as the "Star Degree" of G. Suppose $v_1, \cdots, v_t$ are the pendant vertices adjacent to the quasipendant v_{t+1}. Then it is easily seen that

$$u_i = (1, 0, \cdots, 0, -1, 0, \cdots, 0),$$

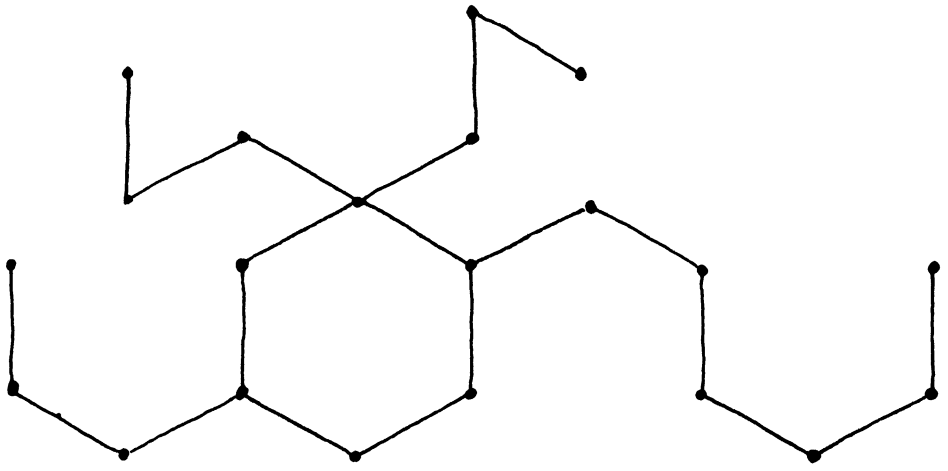


FIG. 6

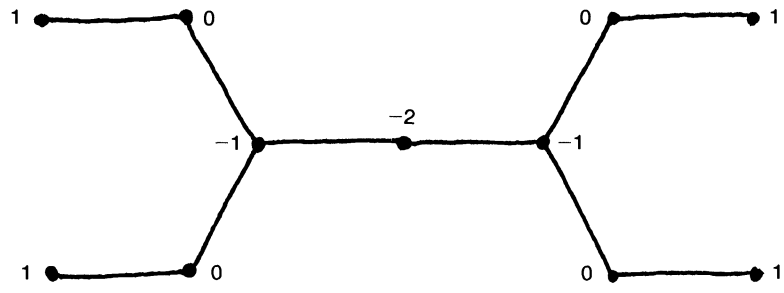


FIG. 7

with -1 in the ith coordinate, $2 \leqq i \leqq t$, is a set of $t-1$ linearly independent eigenvectors for $L(G)$ corresponding to $\lambda = 1$. We will call eigenvectors of this type *Faria vectors*. If the Faria vectors arising at each of the $q(G)$ quasipendant vertices are collected, the resulting $p(G) - q(G)$ set is a basis of what we call the *Faria space*. Thus, the Faria space accounts for the lower bound in (2). Our attention is naturally drawn to the excess or "spurious" multiplicity of one given by

$$s(G) = m_G(1) - p(G) + q(G), \tag{6}$$

i.e., the dimension of the space spanned by eigenvectors of $L(G)$ corresponding to one that are orthogonal to all the Faria vectors.

Let $p = p(G)$, $q = q(G)$, and $r = r(G)$, where $r(G) = n - p - q$ is the number of vertices of G that remain after the pendants and quasipendants have been accounted for. We will refer to these remaining vertices as *inner* vertices.

Assume the vertex set of G is ordered as $V = \{v_1, \cdots, v_n\}$, where $v_1, \cdots, v_r$ are the inner vertices, $v_{r+1}, \cdots, v_{r+q}$ are the quasipendants, and $v_{n-p+1}, \cdots, v_n$ are the pendant vertices. Assume further that $\{v_{r+i}, v_{n-p+i}\} \in E$, $1 \leqq i \leqq q$. It follows that $L(G)$ has the form

$$L(G) = \begin{pmatrix} A & X & 0 \\ X^t & Q & C \\ 0 & C^t & I_p \end{pmatrix},$$

where A is r-by-r and Q is q-by-q. Moreover, the submatrix of C occupying its first q columns is $-I_q$. Using this I_q submatrix (and its transpose in C^t) in elementary row and column operations, we may transform $L(G) - I_n$ to

$$\begin{pmatrix} L_R(G) & 0 & 0 \\ 0 & 0 & B \\ 0 & B^t & 0 \end{pmatrix}, \tag{7}$$

where $B = (-I_q 0)$, and $L_R(G) = A - I_r$ is the leading r-by-r principal submatrix of $L(G) - I_n$. Hence, from (7),

$$\begin{aligned} m_G(1) &= \text{nullity } [L(G) - I_n] \\ &= n - 2q - \text{rank } L_R(G) \\ &= p - q + \text{nullity } L_R(G). \end{aligned} \tag{8a}$$

(Note that the "$p - q$" in (8a) is the same "$p - q$" that arises as the dimension of the Faria space. The nullity of $L_R(G)$ corresponds to the eigenvectors for one that are orthogonal to the Faria vectors.) If we let $m_A(1)$ denote the multiplicity of one as an eigenvalue of A, then we may rewrite (8a) as

$$s(G) = \text{nullity } L_R(G) = m_A(1). \tag{8b}$$

We now proceed to estimate the nullity of $L_R(G)$ in two different ways, giving rise to two upper bounds for $s(G)$. Our first estimate involves the "point independence number" (PIN—also known as the "interior stability number" [2]) of a graph. A subset of vertices is *independent* if no two of them are adjacent. The PIN of G, $\alpha(G)$, is the maximum size of any independent set of vertices. Thus, e.g., $\alpha(K_n) = 1$ and $\alpha(K_{s,t}) = \max\{s, t\}$. If G has (exactly) k connected components $C_1, \cdots, C_k$, then $\alpha(G) = \sum \alpha(C_i)$. If R is the subgraph of G induced on the inner vertices, we will write $e(G) = \alpha(R)$.

THEOREM 3.6. *Suppose $G = (V, E)$ is a graph on n vertices. Then*

$$s(G) \leqq r(G) - e(G). \tag{9}$$

(The quantity $r(G) - e(G)$ is the *covering number* of R.)

Example 3.7. Let G be the graph in Fig. 8. Then $p = q = 2$, and the inner vertex graph R is the graph on $r = 4$ vertices having two components each consisting of a single edge. The matrix "A" is the direct sum of two copies of $3I_2 - J_2$, where J_2 is the 2-by-2 matrix each of whose entries is equal to one. Alternatively, $L_R(G)$ is the direct sum of $2I_2 - J_2$ with itself. In any case, $m_G(1) = s(G) = m_A(1) = \text{nullity } L_R(G) = 2$. On the other hand, $e(G) = 2$ and the upper bound in (9) is sharp. In fact,

$$c_G(x) = x(x-1)^2(x-2)(x-3)(x-4)(x^2-5x+2).$$

Proof of Theorem 3.6. Returning to (7)–(8), it suffices to show that

$$\text{rank } L_R(G) \geqq e = e(G).$$

By definition of e, $L_R(G)$ has a principal e-by-e diagonal submatrix. Since the degree (in G) of every vertex in R is at least two, this diagonal submatrix has full rank. □

The upper bound $r(G) - e(G) \geqq s(G)$ tends to be best when vertex degrees in the induced subgraph R are small. Our next result is a bound that tends to be best when vertex degrees are relatively large. We will say that a graph $G = (V, E)$ on n vertices is *rich* if $G = K_n$ or if $d_i + d_j \geqq n$ whenever $\{v_i, v_j\} \notin E$. (In particular, the "closure" of a rich graph is K_n.)

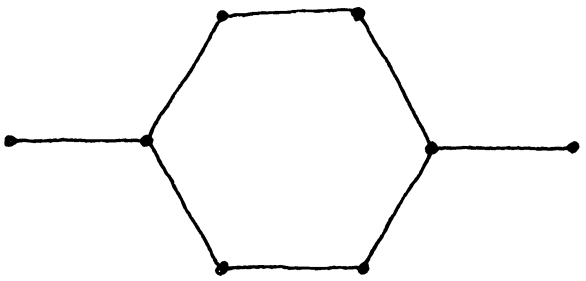


FIG. 8

THEOREM 3.8. *Suppose $G = (V, E)$ is a graph. Denote by R the subgraph of G induced on its inner vertices. If each of the k components of R is a rich graph, then $s(G) \leqq k$.*

In Example 3.7, R has two components, each of which is isomorphic to K_2. Since K_2 is rich, $s(G) \leqq 2$, and the upper bounds of Theorems 3.6 and 3.8 coincide. If, however, G were to be K_n, then $r(G) - e(G) = n - 1$, while the new upper bound is one. (In fact, of course, $m_{K_n}(1) = 0$.)

Proof. Observe that $L(G) + L(\bar{G}) = L(K_n) = nI_n - J_n$, where $\bar{G}$ is the complement of G and J_n is the n-by-n matrix each of whose entries is one. It follows that $\bar{\lambda}_{n-i} = n - \lambda_i$, $1 \leqq i < n$, where $\bar{\lambda}_1 \geqq \cdots \geqq \bar{\lambda}_n = 0$ are the eigenvalues of $L(\bar{G})$ and (as usual) $\lambda_1 \geqq \cdots \geqq \lambda_n = 0$ are the eigenvalues of $L(G)$. We next observe that

$$\lambda_1 \leqq \max_{\{v_i v_j\} \in E} (d_i + d_j).$$

This follows immediately from the Geršgorin Circle Theorem applied to the edge version $K(G)$. (The circles are all centered at two and their radii are $d_i + d_j - 2$, $\{v_i, v_j\} \in E$.)

Now, the matrix $L_R(G)$ is a direct sum, over the components C of R, of matrices $M(C) = L(C) - I + F(C)$, where $F(C)$ is a diagonal matrix with nonnegative integer entries, and I is an appropriately sized identity matrix. Let $\mu_1 \geqq \cdots \geqq \mu_t = 0$ be the eigenvalues of $L(C)$, and $\bar{\mu}_1 \geqq \cdots \geqq \bar{\mu}_t = 0$ the Laplacian eigenvalues of its complement. Denote by δ_i the degree, in C, of the ith vertex of C. Then, because C is rich,

$$\begin{aligned} \bar{\delta}_i + \bar{\delta}_j &= 2(t-1) - (\delta_i + \delta_j) \\ &\leqq t - 2, \end{aligned}$$

for each pair (i, j) corresponding to an edge of $\bar{C}$. Consequently, by what we have just seen, $\bar{\mu}_1 \leqq t - 2$ so $\mu_{t-1} \geqq 2$. We deduce that one eigenvalue of $L(C) - I$ is -1 and the rest are not less than $+1$. Since $M(C) \geqq L(G) - I$, in the positive semidefinite sense, the contribution of $M(C)$ to the rank of $L_R(G)$ is at least $t - 1$. □

It should be remarked that Theorems 3.6 and 3.8 are most effective after paths of length three have been pruned off (see, e.g., Example 3.4). Moreover, it is possible to mix the techniques among the components of R.

In the subsequent discussion, it will be useful to describe eigenvectors of $L(G)$ by labeling the vertices of G with the corresponding components of the eigenvectors. If, e.g., G is the tree in Example 3.5, then $p(G) - q(G) = 0$ and $s(G) = 1$. An eigenvector affording $\lambda = 1$ is exhibited in Fig. 7. It is clear that this vector is something new. It differs from the Faria vectors, e.g., in being constant on the orbits of the automorphism group $\Gamma(G)$. Evidently, (1, 2, 1) is a null vector of $L_R(G)$.

We define the *symmetric part* of the spectrum of $L(G)$ to be those eigenvalues, including appropriate multiplicities, that can be accounted for by eigenvectors that are constant on the orbits of $\Gamma(G)$. If, for example, $\Gamma(G)$ is trivial, then every eigenvalue is "symmetric." If, on the other hand, $\Gamma(G)$ acts transitively on the vertices, then $\lambda = 0$ is the only symmetric eigenvalue. In general, the number of symmetric eigenvalues of $L(G)$, multiplicities included, is equal to the number of orbits of $\Gamma(G)$.

We will say an eigenvalue is "alternating" or that it belongs to the *alternating part* of the spectrum if it is afforded by an eigenvector that (such as each of the Faria vectors) is orthogonal to the characteristic functions of the orbits. If T is the tree in Example 2.7, then $\lambda = 2$ is in the symmetric part and $\lambda = 1$ is in the alternating part. (Every eigenvector afforded by $\lambda = 1$ is in the Faria space.)

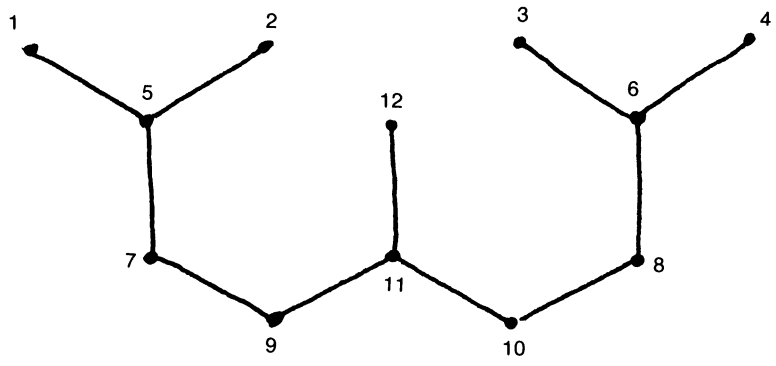


FIG. 9

Example 3.9. Let T be the tree in Fig. 9 with the vertices numbered as shown. Then, using (5), we may easily confirm that

$$u^{(1)} = (1,1,1,1,0,0,-2,-2,-2,-2,0,4),$$

$$u^{(2)} = (1,1,-1,-1,0,0,-2,2,-2,2,0,0),$$

$$u^{(3)} = (1,-1,0,0,0,0,0,0,0,0,0,0),$$

and

$$u^{(4)} = (0,0,1,-1,0,0,0,0,0,0,0,0),$$

are orthogonal eigenvectors corresponding to $\lambda = 1$. On the other hand (in the notation of Theorems 3.6 and 3.8), R consists of the subgraph induced on $\{v_i : 7 \leqq i \leqq 10\}$, both components of which are rich. (Alternatively, $\alpha(R) = 2$.) Thus,

$$\begin{aligned} m_T(1) &= P(T) - q(T) + s(T) \\ &= 4 - 2 + s(T) \\ &\leqq 2 + 2 = 4. \end{aligned}$$

Observe that $u^{(1)}$ is symmetric, $u^{(3)}$ and $u^{(4)}$ are Faria vectors, whereas $u^{(2)}$ is a yet to be explained eigenvector that is alternating but not in the Faria space. (Note that both $u^{(1)}$ and $u^{(2)}$ arise from the null space of the matrix $L_R(G)$.)

It is a straightforward procedure to determine the symmetric part of the spectrum [10]. Any symmetric eigenvector must be in the space spanned by the characteristic functions of the orbits. The graph T in Fig. 9, for example, has six orbits. Hence, $c_T(x) = f(x)g(x)$, where $f(x)$ has degree six (accounting for the symmetric part of the spectrum) and $g(x)$ has degree $12 - 6 = 6$, accounting for the alternating part. Note that $f(x) = xf_1(x)$ since the eigenvector corresponding to $\lambda = 0$ is constant on *all* vertices. To obtain $f(x)$, we perform a similarity transformation of the following type. Suppose the m orbits of $\Gamma(G)$ have sizes $k_1, k_2, \cdots, k_m$ and characteristic functions $w_1, \cdots, w_m$. Then the vectors $u_j = k_j^{-1/2} w_j$, $1 \leqq j \leqq m$, are orthonormal. Let U be any orthogonal matrix having u_j in column j, $1 \leqq j \leqq m$. Then $U^t L(G) U$ is the direct sum of an m-by-m matrix A (affording the symmetric part of the spectrum) and an $(n-m)$-by-$(n-m)$ matrix B. We note that A can easily be obtained as follows. Order the vertices of G by orbits and partition $L(G)$ into m^2 blocks of sizes k_i-by-k_j. Then the (i,j)-element of A is obtained by summing the elements in the (i,j)-block of $L(G)$ and dividing by $(k_i k_j)^{1/2}$.

Example 3.10. Let T be the tree in Fig. 9. Then

$$L(T) = \left[\begin{array}{cccc|cc|cc|cc|cc} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & -1 & 0 & 0 & 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 3 & 0 & -1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -1 & 0 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 0 & -1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{array}\right].$$

Hence,

$$A = \begin{pmatrix} 1 & -\sqrt{2} & 0 & 0 & 0 & 0 \\ -\sqrt{2} & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -\sqrt{2} & 0 \\ 0 & 0 & 0 & -\sqrt{2} & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix},$$

and $f(x) = \det(xI - A) = x(x-1)(x-4)(x^3 - 7x^2 + 12x - 3)$. It turns out that $g(x) = c_T(x)/f(x) = (x-1)^3(x^3 - 7x^2 + 12x - 1)$.

Returning to (6), we observe (since $r(G) \geqq s(G)$) that $n - m_G(1) \geqq 2q(G)$. We claim, in fact, that $m_G[0, 1) \geqq q(G)$ and $m_G(1, \infty) \geqq q(G)$, where $m_G(I)$ denotes the number of eigenvalues of $L(G)$, multiplicities included, belonging to the interval I.

THEOREM 3.11. *Let G be a graph. Then $m_G[0, 1) \geqq q(G)$ and $m_G(1, \infty) \geqq q(G)$.*

It follows from (2) and Theorem 3.11 that $m_G[0, 1] \geqq p(G) \leqq m_G[1, \infty)$. (This fact may also be proved by observing that I_p, $p = p(G)$, is a principal submatrix of $L(G)$, and using the Cauchy interlacing inequalities.) A result similar to Theorem 3.11 for $m_G(0, 2)$ and $m_G(2, \infty)$, when G is a tree, can be found in Corollary 4.3 below.

Before attempting a proof of Theorem 3.11 we require some background concerning the relationship of the sequence of leading principal subdeterminants of a symmetric matrix to the number of positive and negative eigenvalues of the matrix. Suppose that A is n-by-n, symmetric and nonsingular. Let $\alpha_0 = 1$ and let $\alpha_k = \det(A_k)$ where A_k is the leading principal k-by-k submatrix of A. It is well known (or easily proven by induction on n) that the number of negative eigenvalues of A is equal to the number of sign changes in the sequence $(\alpha_0, \alpha_1, \cdots, \alpha_n)$. We note that this sequence may contain intermediate zeros, in which case we can shorten the sequence by deleting the zeros and the theorem will still hold. As an immediate consequence we have the following lemma.

LEMMA 3.12. *Suppose that $A = A^t$ is $2q$-by-$2q$ and that $\det(A_{2k}) = (-1)^k$, $k = 1, \cdots, q$. Then A has q positive and q negative eigenvalues.*

Another well-known fact we require relates the spectrum of a principal submatrix of symmetric A to the spectrum of A.

LEMMA 3.13. *Suppose that B is a principal submatrix of the symmetric matrix A and that α is real. Then the number of eigenvalues of B that are greater than (respectively, greater than or equal to, less than, less than or equal to) α is a lower bound for the*

number of eigenvalues of A that are greater than (respectively, greater than or equal to, less than, less than or equal to) α.

Proof of Theorem 3.11. We may assume without loss of generality that the quasi-pendant vertices of G are numbered $1, 3, \cdots, 2q-1$, and that vertex $2k$ is a pendant vertex adjacent to vertex $2k-1$, for each $k = 1, \cdots, q$. Let B be the leading principal $2q$-by-$2q$ submatrix of $L(G)$. In view of Lemma 3.12, it will suffice to show that B has q eigenvalues greater than one and q eigenvalues less than one. To do this it will suffice to show that $A = B - I_{2q}$ satisfies the hypotheses of Lemma 3.12. Note that A has the form

$$\begin{pmatrix} (d_1-1) & -1 & * & 0 & \cdots & * & 0 \\ -1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ * & 0 & (d_3-1) & -1 & \cdots & * & 0 \\ 0 & 0 & -1 & 0 & \cdots & 0 & 0 \\ & \cdots & & & & & \\ * & 0 & * & 0 & \cdots & (d_{2q}-1) & -1 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 \end{pmatrix}$$

and that the even numbered rows (corresponding to pendants) of A have a single nonzero entry. We assume an inductive hypothesis on q, and hence it will suffice to prove that $\det(A) = (-1)^q$. If we use elementary row and column operations corresponding to adding multiples of even numbered rows and columns to other rows and columns, then A can be transformed into a direct sum of m copies of $-P$, where P is the 2-by-2 permutation matrix corresponding to a transposition. Hence $\det(A) = [\det(-P)]^q = (-1)^q$ and the proof is finished. □

In Theorem 3.1, we modified the connected sum idea from Theorem 2.5 and showed that $m_G(1) = m_H(1)$ when $H = G \vee P_3$, some graph obtained from G and P_3 by joining any vertex of G to a *pendant* vertex of P_3. Denote by $G \multimap P_3$ some graph obtained from G and P_3 by joining any vertex of G to the middle (quasipendant) vertex of P_3.

PROPOSITION 3.14. *Let G be a graph on n vertices and suppose $H = G \multimap P_3$. Then $m_G(1) \leqq m_H(1) \leqq m_G(1) + 2$, and each of the three possibilities for $m_H(1)$ can occur.*

Proof. Write $m_G(1) = m$. Assume the numbering of vertices to be such that the last vertex of G and the first vertex of P_3 have been joined to form H. Let M be an $(m-1)$-dimensional subspace of $\ker(L(G) - I_n)$ such that $v_n = 0$ for all $v \in M$. Then $\{v \oplus (0, a, -a) : v \in M, a \in R\}$ is an m-dimensional subspace of $\ker(L(H) - I_{n+3})$. Hence, $m \leqq m_H(1)$.

As in the proofs of Theorems 2.5 and 3.1, let $L = L(G)$, $L' = L(P_3)$, and $\tilde{L} = L(H)$. Then $\tilde{L} = (L \dotplus L') + A$, where A is the rank one matrix whose only nonzero entries amount to

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

in rows and columns n and $n+1$. Then

$$\begin{aligned} \operatorname{rank}(\tilde{L} - I_{n+3}) &= \operatorname{rank}([(L - I_n) \dotplus (L' - I_3)] + A) \\ &\geqq \operatorname{rank}[(L - I_n) \dotplus (L' - I_3)] - 1 \\ &= \operatorname{rank}(L - I_n) + 1, \end{aligned}$$

so

$$\begin{aligned} m_H(1) &= \text{nullity}\,(\tilde{L} - I_{n+3}) \\ &\leqq n+2-\text{rank}\,(L-I_n) \\ &= m+2. \end{aligned}$$

The last assertion is demonstrated by the following examples: (1) Let $G = P_3$. Form $G \multimap P_3$ by joining a pendant vertex of G to the "middle" vertex of P_3. Then $P_3 = G$ can be pruned off as in Example 3.2 and $m_H(1) = m_G(1)$. (2) Let $G = P_2$. Then $m_G(1) = 0$ while the characteristic polynomial of $G \multimap P_3$ (pictured in Fig. 1) is $x(x-1)(x^3 - 7x^2 + 13x - 5)$. (3) Let G be the tree in Fig. 10. Form $G \multimap P_3$ by joining the open vertex to the middle of P_3. Then $G \multimap P_3$ is the tree in Fig. 9 (Example 3.9). In this case, $m_G(1) = 2$ and $m_H(1) = 4$.

We now return to the "spurious multiplicity" $s(G)$ in (6). We know that the multiplicity of $\lambda = 1$ in the symmetric part of the spectrum of $L(G)$ accounts for part but not (in general) all of $s(G)$. (See Example 3.9). In Theorems 3.6 and 3.8 we found upper bounds for $s(G)$. We conclude this section with a discussion of possible lower bounds when G is a tree. We begin by defining an equivalence relation on the set Q of quasipendants of a tree T. If $v_1, v_2 \in Q$, we say $v_1 \equiv v_2$ if the distance $d(v_1, v_2)$ from v_1 to v_2 is an (integer) multiple of three, and if the degree d_t of vertex v_t is two whenever v_t is on the unique path from v_1 to v_2 and $d(v_1, v_t) \equiv 0 \pmod 3$.

PROPOSITION 3.15. *Let Q be the set of quasipendants of a tree T. Suppose $C_1, \cdots, C_t$ are the equivalence classes of Q and that their respective cardinalities are $q_1, \cdots, q_t$. Then $s(T) \geqq (\sum q_i) - t$.*

The somewhat laborious proof of this result involves finding a principal submatrix of $L_R(G)$ (see (7)) of sufficiently large nullity. This submatrix turns out to be a direct sum of $2I_2 - J_2$ with itself several times. We omit the computational details.

Example 3.16. Let T be the tree in Fig. 9. Then Q consists of the vertices numbered 5, 6, and 11. In this case, Q consists of a single equivalence class of size $q_1 = q(G) = 3$, and Proposition 3.15 asserts that $s(T) \geqq 2$. Since $p(T) - q(T) = 5 - 3 = 2$, and $m_T(1) = 4$ (Example 3.9), we know that $s(T) = 2$. In other words, Proposition 3.15 is strong enough to capture the existence (but not the nature) of eigenvectors $u^{(1)}$ and $u^{(2)}$ in Example 3.9.

Example 3.17. The tree T in Fig. 11 is exhibited with an eigenvector affording $\lambda = 1$. Indeed, for this tree, $m_T(1) = 1 = s(T)$, $p(T) = q(T)$, the lower bound given by Proposition 3.15 is zero (no two quasipendants are equivalent), the upper bound given in (9) is two, and Theorem 3.8 does not apply. It is an abundance of such examples

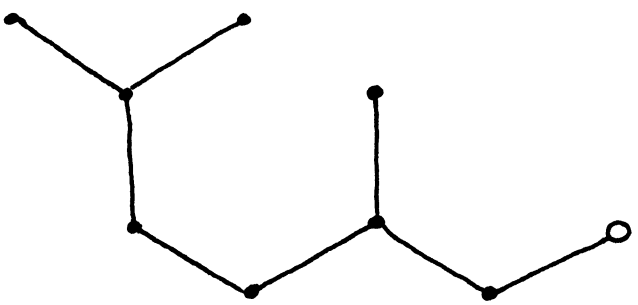


FIG. 10

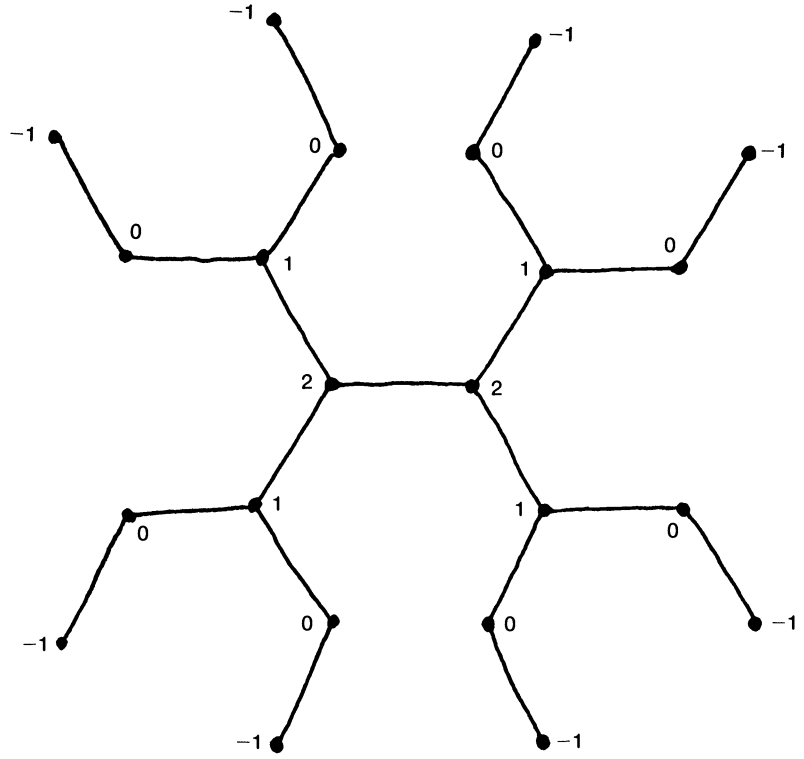


FIG. 11

that leads the authors to believe there can be no simple graph theoretic interpretation for $m_T(1)$.

4. Surgery on graphs. Techniques that allow one graph to be transformed into another with predictable effects on the eigenvalues have already proved useful. (See, e.g., Examples 2.6 and 3.2.) The main purpose of this section is to examine the influence of "moving an edge" in the geometric senses of (i) removing it without affecting its endpoints, (ii) removing it and identifying its endpoints, (iii) disconnecting one pair of vertices and joining some other pair. (Note that we have already addressed, to some small extent, the removal of a pendant edge and *both* of its endpoints. See Example 2.6.) Our first result is part of the "Laplacian Folklore."

THEOREM 4.1. *Let $\tilde{G}$ be a graph on n vertices. Suppose G is a (spanning) edge subgraph of $\tilde{G}$ obtained by removing just one of its edges. Then the $(n-1)$ largest eigenvalues of $L(G)$ interlace the eigenvalues of $L(\tilde{G})$.*

Proof. It suffices to prove that the nonzero eigenvalues interlace and for this we may consider the edge version $K(G)$. The result follows from the Cauchy interlacing inequalities because $K(G)$ is a principal submatrix of $K(\tilde{G})$. □

Note that if a pendant edge of $\tilde{G}$ is removed in Theorem 4.1, then $L(G)$ is a direct sum of $L(G')$ and (0), where G' is obtained from $\tilde{G}$ by removing both the pendant edge and vertex. Thus, the nonzero eigenvalues of $L(G)$ and $L(G')$ are the same. We have proved the following corollary.

COROLLARY 4.2. *Suppose v is a pendant vertex of the graph $\tilde{G}$. Let G be the graph obtained from $\tilde{G}$ by removing v (and its edge). Then the eigenvalues of $L(G)$ interlace the eigenvalues of $L(\tilde{G})$.*

We can use Corollary 4.2 to obtain a result similar to Theorem 3.11. Reviving the notation there, recall that $m_G(I)$ denotes the number of eigenvalues of $L(G)$, multiplicities included, belonging to the interval I.

COROLLARY 4.3. *If T is a tree with diameter d, then*

$$m_T(0,2) \geqq [d/2] \leqq m_T(2,\infty),$$

where square brackets indicate the greatest integer function.

Proof. First consider the case that $T = P_{d+1}$. We can easily show that $K(T) = 2I_d + A(T^*)$, where $T^* = P_d$ is the line graph of T. Since the spectrum of $A(T^*)$ is symmetric about the origin, the spectrum of $K(T)$ is symmetric about two, i.e., the nonzero spectrum of $L(T)$ is symmetric about two. (Together with Theorem 2.1(ii), this gives another proof of Corollary 2.8(i).) Since $m_T(2) \leqq 1$, the result is established in this case. Now, any tree T with diameter d contains P_{d+1} as a subtree. Thus, T can be reduced to P_{d+1} by a sequential removal of pendant vertices. The result follows from the interlacing established in Corollary 4.2. (See Lemma 3.13.) □

This seems an appropriate place to recall a striking result of Fiedler [8, p. 612]: Suppose two is an eigenvalue of $L(T)$ for some tree $T = (V, E)$. Let u be an eigenvector of $L(T)$ corresponding to two. Then the number of eigenvalues of $L(T)$ greater than two is equal to the number of edges $\{v_i, v_j\} \in E$ such that $u_i u_j > 0$, whereas the number of eigenvalues of $L(T)$ less than two is equal to the number of edges such that $u_i u_j < 0$. (Note that Theorem 2.1(iii) guarantees $u_i \neq 0$ for all i.) If, for example, T is the tree in Fig. 4 (with u exhibited), the six pendant edges are all of the type $u_i u_j < 0$, while the remaining three edges all yield $u_i u_j > 0$. Thus, exactly six eigenvalues of $L(T)$ are less than two, whereas exactly three are greater than two. In this case, $d = 4$ and $[d/2] = 2$.

In another pioneering paper [7], Fiedler introduced the algebraic connectivity $a(G) = \lambda_{n-1}$ of G. He proved that the algebraic connectivity of a path,

$$a(P_n) = 2(1 - \cos(\pi/n)),$$

is a lower bound for $a(G)$ for any connected graph G on n vertices. As another application of Corollary 4.2, we recover a related upper bound stated in the context of $A(G^*)$ by Doob [4, p. 187].

COROLLARY 4.4. *Let T be a tree on n vertices with diameter d. Then*

$$a(T) \leqq 2(1 - \cos(\pi/(d+1))).$$

Proof. Observe that T can be built up from P_{d+1} by attaching pendant vertices. It is seen from Corollary 4.2 that this building process does not increase the algebraic connectivity. □

COROLLARY 4.5. *Let T be a tree on $n \geqq 6$ vertices. If $T \neq *_n$, then $a(T) < 0.49$.*

Proof. As in the proof of Corollary 4.4, we may build T on the foundation of P_4. After attaching two pendant vertices, we arrive at a (possibly intermediate) stage of a tree T_2 with six vertices. Moreover, $T_2 \neq *_6$. There are only five possibilities for T_2. The one with maximum algebraic connectivity is the "near star" in Fig. 12(b) with algebraic connectivity $= 0.485\cdots$. Repeated applications of Corollary 4.2, as more pendant vertices are attached, proves that $a(T) \leqq a(T_2)$. □

Our next result is reminiscent of popular newspaper accounts of the recombinant techniques of molecular genetics.

THEOREM 4.6. *Let $G_1 = (V, E_1)$ be a graph and $G_2 = (V, E_2)$ a graph obtained from G_1 by removing an edge and adding a new edge that was not there before. Suppose $\alpha_1 \geqq \cdots \geqq \alpha_n$ are the eigenvalues of $L(G_1)$ and $\beta_1 \geqq \cdots \geqq \beta_n$ are the eigenvalues of $L(G_2)$. Then $\alpha_i \geqq \beta_{i+1}$ and $\beta_i \geqq \alpha_{i+1}$, $1 \leqq i < n$.*

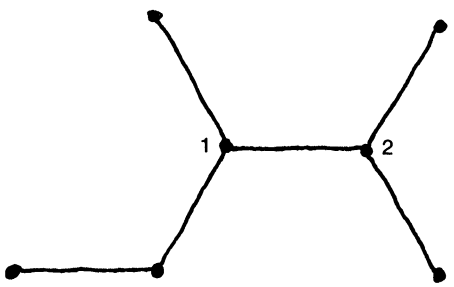


FIG. 12(a)

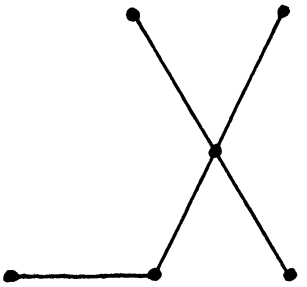


FIG. 12(b)

Proof. From the perspective of the edge version, $K(G_1)$ and $K(G_2)$ share an $(m-1)$-by-$(m-1)$ principal submatrix, where $m = o(E_1) = o(E_2)$ is the common cardinality of the two edge sets. Once again, the result is immediate from Cauchy interlacing. □

We now come to a less trivial situation in which the vertices at the ends of an edge are identified, in the process of which the edge is "collapsed" (or "contracted") and disappears (without producing a loop). In fact, Corollary 4.2 can be redrafted as a special case of this procedure, the case in which a pendant edge is collapsed. Our next result is a consequence of the Monotonicity Theorem [12].

LEMMA 4.7. *Let A, B, and C be n-by-n Hermitian matrices satisfying $A = B + C$. Denote the eigenvalues of A and B by $\alpha_1 \geqq \cdots \geqq \alpha_n$ and $\beta_1 \geqq \cdots \geqq \beta_n$, respectively. If C has exactly t positive eigenvalues, then $\beta_k \geqq \alpha_{k+t}$, $1 \leqq k \leqq n-t$.*

COROLLARY 4.8. *Let A, B, and C be n-by-n Hermitian matrices satisfying $A = B + C$. Denote the eigenvalues of A and B by $\alpha_1 \geqq \cdots \geqq \alpha_n$ and $\beta_1 \geqq \cdots \geqq \beta_n$, respectively. If C has exactly one positive eigenvalue and exactly one negative eigenvalue (so that rank $C = 2$), then $\alpha_k \geqq \beta_{k+1}$ and $\beta_k \geqq \alpha_{k+1}$, $1 \leqq k < n$.*

THEOREM 4.9. *Let $\tilde{G} = (\tilde{V}, \tilde{E})$ be a graph with $\tilde{e} = \{\tilde{v}_1, \tilde{v}_2\} \in \tilde{E}$. Suppose $\tilde{e}$ does not lie on a circuit of length three. Let $G = (V, E)$ be the graph obtained from $\tilde{G}$ by deleting (i.e., "collapsing") $\tilde{e}$ and identifying vertices $\tilde{v}_1$ and $\tilde{v}_2$. If $\tilde{\lambda}_1 \geqq \cdots \geqq \tilde{\lambda}_{n+1} = 0$ are the eigenvalues of $\tilde{L} = L(\tilde{G})$ and $\lambda_1 \geqq \cdots \geqq \lambda_n = 0$ are the eigenvalues of $L = L(G)$, then*

(i) $\lambda_i \geqq \tilde{\lambda}_{i+1}$, $1 \leqq i \leqq n$, *and*

(ii) $\tilde{\lambda}_i \geqq \lambda_{i+1}$, $1 \leqq i < n$.

Example 4.10. Let $\tilde{G}$ be the graph in Fig. 12(a) with spectrum (approximately)

$$4.63 > 3.23 > 2.14 > 1.00 > 0.68 > 0.32 > 0.00.$$

If the 1-2 edge is collapsed, the result is the graph G in Figure 12(b) with spectrum

$$5.09 > 2.43 > 1.00 = 1.00 > 0.49 > 0.00.$$

Proof of Theorem 4.9. Let $L_0 = (0) \dotplus L$. Then $\tilde{L} = L_0 + A$, where $A = (A_{ij})$ is a 3-by-3 block partitioned matrix:

$$A_{11} = \begin{pmatrix} k+1 & -1 \\ -1 & 1-k \end{pmatrix},$$

where $k+1$ is the degree (in $\tilde{G}$) of $\tilde{v}_1$; A_{12} is the 2-by-k matrix whose first row consists entirely of -1's and whose second row is all $+1$'s; $A_{21} = A_{12}^t$, and the other blocks are appropriately sized zero matrices. (In particular, A_{22} and A_{33} are square blocks, whereas

A_{23} is k-by-$(n-k-1)$.) Suppose first that $k > 0$. Then it suffices to prove that the inertia of A is $(1, 1, n-1)$, and invoke Corollary 4.8. Let X be the $(n+1)$-square matrix

$$X = \begin{pmatrix} -k-1 & 0 & 1 & 1 & 1 & \cdots & 1 \\ 1 & -k & 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & k & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & k & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & k & \cdots & 0 \\ & & \cdot & \cdot & \cdot & & \\ 1 & 1 & 0 & 0 & 0 & \cdots & k \end{pmatrix}.$$

Then we may confirm that each column of X is an eigenvector for A. More particularly, X^1, the first column, corresponds to the eigenvalue $\lambda = k + 2$; X^2 corresponds to $\lambda = -k$; and $X^3, \cdots, X^{n+1}$ all correspond to $\lambda = 0$.

The degenerate case $k = 0$ has already been established in Corollary 4.2. In fact, we can recover that stronger result here too since, in this case, $A \geqq 0$ (in the positive semi-definite sense) and rank $A = 1$. In this case, (i) is proved by appealing to Lemma 4.7, and $\tilde{\lambda}_i \geqq \lambda_i$, $1 \leqq i \leqq n$, because $\tilde{L} \geqq L_0$. □

It is clear from Example 4.10 that the strong inequalities $\tilde{\lambda}_i \geqq \lambda_i$, $1 \leqq i \leqq n$, may not hold for a general edge collapse, even for trees. Indeed, as the next result shows, Example 4.10 is not an isolated example.

THEOREM 4.11. *Let $\tilde{T}$ be a tree. Suppose $\tilde{e}$ is an edge of $\tilde{T}$ each of whose endpoints has degree at least three. If T is obtained from $\tilde{T}$ by collapsing $\tilde{e}$, then $\lambda_1 > \tilde{\lambda}_1$.*

Proof. It is somewhat more convenient to deal with the matrix $B(T) = D(T) + A(T)$ that, in view of Proposition 2.2 affords the same spectrum as $L(T)$. Let $\tilde{u}$ be the positive Perron eigenvector of $B(\tilde{T})$, normalized so that $\|\tilde{u}\| = 1$. The theorem will be proved by producing a unit vector u of size n such that $(B(T)u, u) > \tilde{\lambda}_1$.

Assume $\tilde{e} = \{\tilde{v}_n, \tilde{v}_{n+1}\}$ and write $a = \tilde{u}_n$ and $b = \tilde{u}_{n+1}$. Then $\tilde{e}$ and its immediate neighbors are pictured in Fig. 13, where the labels represent corresponding coordinates of $\tilde{u}$. Define $u \in \mathbb{R}^n$ by $u_i = \tilde{u}_i$, $1 \leqq i < n$ and $u_n = \alpha$, where $\alpha = (a^2 + b^2)^{1/2}$. Note that u is a unit vector. Observe also that

$$(B(T)u, u) = \sum (u_i + u_j)^2,$$

where the sum extends over those pairs (i, j) such that $\{v_i, v_j\}$ is an edge of T. Note that many of the terms in $\tilde{\lambda}_1 = (B(\tilde{T})\tilde{u}, \tilde{u})$ and $(B(T)u, u)$ are the same. Denote the sum of these common terms by c. Then

$$\begin{aligned} \tilde{\lambda}_1 &= (B(\tilde{T})\tilde{u}, \tilde{u}) \\ &= c + \sum_{i=1}^{j} (p_i + a)^2 + (a+b)^2 + \sum_{i=1}^{k} (b + x_i)^2 \end{aligned}$$

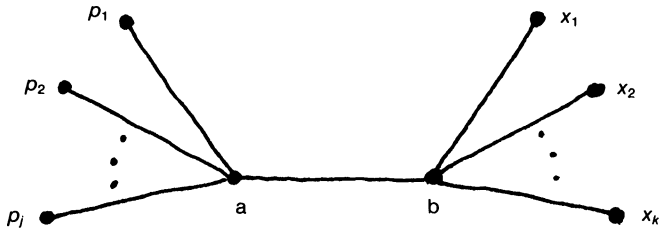


FIG. 13

while

$$(B(T)u,u)=c+\sum_{i=1}^{j}(p_i+\alpha)^2+\sum_{i=1}^{k}(\alpha+x_i)^2.$$

To show $(B(T)u, u) > \tilde{\lambda}_1$, we take their difference

$$2\sum_{i=1}^{j}p_i(\alpha-a)+2\sum_{i=1}^{k}x_i(\alpha-b)+(a-b)^2+(k-2)a^2+(j-2)b^2$$

and observe that $j + 1 = \tilde{d}_n$, the degree of $\tilde{v}_n$, while $k + 1 = \tilde{d}_{n+1} \geqq 3$. Hence, every term is nonnegative, and the first $j + k$ terms are all positive. □

It should be noted that the same reasoning will prove a slightly more general assertion: Let $\tilde{T}$ be a tree. Suppose $\tilde{v}_1$ and $\tilde{v}_2$ are vertices each of degree at least three. Suppose the unique path from $\tilde{v}_1$ to $\tilde{v}_2$ is homeomorphic to an edge (i.e., apart from the endpoints, each vertex on the path has degree two). Let T be the tree obtained by collapsing the entire path. Then $\lambda_1 > \tilde{\lambda}_1$. (Of course, if $\tilde{v}_1$ were a pendant vertex, we could have deduced $\lambda_1 \leqq \tilde{\lambda}_1$.)

Conjecture 4.12. Let $\tilde{T}$ be a tree with (Laplacian) spectrum $\tilde{\lambda}_1 \geqq \cdots \geqq \tilde{\lambda}_n > \tilde{\lambda}_{n+1} = 0$. Let T be a tree obtained from $\tilde{T}$ by collapsing an edge. Then $\tilde{\lambda}_{n-1} \geqq \lambda_{n-1} = a(T)$, the algebraic connectivity of T.

Example 4.13. Let $\tilde{G} = C_4$ with spectrum (4, 2, 2, 0). If an edge of $\tilde{G}$ is collapsed, the result is $G = C_3$ with spectrum (3, 3, 0). In this case, $n = 3$ and $\tilde{\lambda}_2 = 2 < \lambda_2 = 3$. Thus, Conjecture 4.12 fails, even for a bipartite $\tilde{G}$ with a circuit.

For general edge collapsing in trees, empirical evidence suggests that departure from interlacing occurs "near the top." We conclude by showing that $\lambda_1 \in [\tilde{\lambda}_2, 2\tilde{\lambda}_1]$.

PROPOSITION 4.14. *Let T be the tree obtained from $\tilde{T}$ by collapsing an edge $\tilde{e} = \{\tilde{v}_1, \tilde{v}_2\}$. Let $\tilde{d}$ be the minimum of the degrees $\tilde{d}_1$ and $\tilde{d}_2$. Then $|\lambda_1 - \tilde{\lambda}_1| \leqq \tilde{d} + 1 \leqq \tilde{\lambda}_1$. Consequently, $\lambda_1 \leqq 2\tilde{\lambda}_1$.*

Proof. We revive the notation used in the proof of Theorem 4.9, with $\tilde{d} = k + 1$. Then

$$\begin{aligned}|\tilde{\lambda}_1-\lambda_1| &= |\,\|\tilde{L}\|-\|L_0\|\,|\\ &\leqq \|\tilde{L}-L_0\|\\ &= \|A\| = k+2.\end{aligned}$$

Now, $\tilde{\lambda}_1$ is bounded below by the largest main diagonal entry of $L(\tilde{T})$. This is at least $\tilde{d} + 1$ unless $\tilde{d}$ is the largest degree of any vertex of $\tilde{T}$. If it is, then $\tilde{d}_1 = \tilde{d}_2 = k + 1$, and

$$B=\begin{pmatrix}k+1 & -1\\ -1 & k+1\end{pmatrix}$$

is a principal submatrix of $L(\tilde{T})$. But, the eigenvalues of B are k and $k + 2$. Thus, by Cauchy interlacing, $\tilde{\lambda}_1 \geqq k + 2 = \tilde{d} + 1$. □

Acknowledgments. The authors are grateful for useful conversations with Stephen Pierce and William Watkins. Many of the examples were worked out using C. Moler's MATLAB, and many ideas occurred to us while examining tables of tree eigenvectors prepared by David Powers [14].

REFERENCES

[1] F. BIEN, *Constructions of telephone networks by group representations*, Notices Amer. Math. Soc., 36 (1989), pp. 5–22.
[2] D. CVETKOVIĆ AND M. DOOB, *Developments in the theory of graph spectra*, Linear and Multilinear Algebra, 18 (1985), pp. 153–181.
[3] D. CVETKOVIĆ, M. DOOB, I. GUTMAN, AND A. TORĜASEV, *Recent Results in the Theory of Graph Spectra*, North-Holland, Amsterdam, 1988.
[4] D. CVETKOVIĆ, M. DOOB, AND H. SACHS, *Spectra of Graphs*, Academic Press, New York, 1979.
[5] B. EICHINGER, *Configuration statistics of Gaussian molecules*, Macromolecules, 13 (1980), pp. 1–11.
[6] I. FARIA, *Permanental roots and the star degree of a graph*, Linear Algebra Appl., 64 (1985), pp. 255–265.
[7] M. FIEDLER, *Algebraic connectivity of graphs*, Czech. Math. J., 23 (1973), pp. 298–305.
[8] ———, *Eigenvectors of acyclic matrices*, Czech. Math. J., 25 (1975), pp. 607–618.
[9] W. FORSMAN, *Graph theory and the statistics of polymer chains*, J. Chemical Phys., 65 (1976), pp. 4111–4115.
[10] R. GRONE AND R. MERRIS, *Algebraic connectivity of trees*, Czech. Math. J., 37 (1987), pp. 160–170.
[11] I. GUTMAN, *Graph-theoretical formulation of Forsman's equations*, J. Chem. Phys., 68 (1978), pp. 1321–1322.
[12] Y. IKEBE, T. INAGAKI, AND S. MIYAMOTO, *The monotonicity theorem, Cauchy's interlace theorem, and the Courant–Fischer theorem*, Amer. Math. Monthly, 95 (1988), pp. 352–354.
[13] B. MOHAR, *The Laplacian spectrum of graphs*, Preprint Series, Department of Mathematics, University E. K. Ljubljana, Yugoslavia, 26, 1988, pp. 353–384.
[14] D. L. POWERS, *Tree eigenvectors*, unpublished tables, 1986.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 239–271, April 1990

007

ROBUST STABILITY AND PERFORMANCE ANALYSIS FOR STATE-SPACE SYSTEMS VIA QUADRATIC LYAPUNOV BOUNDS*

DENNIS S. BERNSTEIN† AND WASSIM M. HADDAD‡

Abstract. For a given asymptotically stable linear dynamic system it is often of interest to determine whether stability is preserved as the system varies within a specified class of uncertainties. If, in addition, there also exist associated performance measures (such as the steady-state variances of selected state variables), it is desirable to assess the worst-case performance over a class of plant variations. These are problems of robust stability and performance analysis. In the present paper, quadratic Lyapunov bounds used to obtain a simultaneous treatment of both robust stability and performance are considered. The approach is based on the construction of modified Lyapunov equations, which provide sufficient conditions for robust stability along with robust performance bounds. In this paper, a wide variety of quadratic Lyapunov bounds are systematically developed and a unified treatment of several bounds developed previously for feedback control design is provided.

Key words. robust analysis, stability, performance, Lyapunov equations, structured uncertainty

AMS(MOS) subject classifications. 15A24, 15A45, 93D05

1. Introduction.

Unavoidable discrepancies between mathematical models and real-world systems can result in degradation of control-system performance including instability [1], [2]. Ideally, feedback control systems should be designed to be *robust* with respect to uncertainties, or perturbations, in the plant characteristics. Such uncertainties may arise either due to limitations in performing system identification prior to control-system implementation or because of unpredictable plant changes that occur during operation. Thus robustness *analysis* must play a key role in control-system design. That is, given an existing or proposed control system, determine the performance degradation due to variations in the plant.

In performing robustness analysis there are two principal concerns, namely, stability robustness and performance robustness. Stability robustness addresses the qualitative question as to whether or not the system remains stable for all plant perturbations within a specified class of uncertainties. A related problem involves determining the largest class of plant perturbations under which stability is preserved. Once robust stability has been ascertained, it is of interest to investigate *quantitatively* the performance degradation within a given robust stability range. In practice it is often desirable to determine the *worst-case* performance as a measure of degradation.

The concern for both robust stability and performance can be traced back to the earliest developments in control theory. Design specifications such as gain and phase margin have traditionally been used to gauge system reliability in the face of uncertainty. In the modern control literature considerable effort has focused on rigorous robustness analysis and design techniques in a variety of settings. Analysis and synthesis results have been developed for both state-space and frequency-domain plant models to address structured parameter variations as well as normed-neighborhood uncertainty [3]–[7].

The present paper is concerned solely with the analysis of structured real-valued parameter uncertainty within the context of state-space models. One motivation for such

* Received by the editors June 20, 1988; accepted for publication (in revised form) July 10, 1989. This research was supported in part by the Air Force Office of Scientific Research under contracts F49620-86-C-0002 and F49620-89-C-0011.

† Harris Corporation, Government Aerospace Systems Division, MS 22/4842, Melbourne, Florida 32902.

‡ Department of Mechanical and Aerospace Engineering, Florida Institute of Technology, Melbourne, Florida 32901.

problems is illustrated by the examples given in [1] and [2]. These examples show that standard linear-quadratic methods used to design either full-state feedback controllers or dynamic compensators may result in closed-loop systems that are arbitrarily sensitive to structured real-valued plant parameter variations. A particularly effective technique for analyzing robust stability is to construct a quadratic Lyapunov function $V(x) = x^T P x$, which guarantees stability of the system as the uncertain parameters vary over a specified range. This technique has been extensively developed for both analysis and synthesis (see, e.g., [8]–[37]).

Although both robust stability and performance are of interest in practice, most of the literature involving quadratic Lyapunov functions is confined to the problem of robust stability. A notable exception is the early work of Chang and Peng [9], which also provides bounds on worst-case quadratic performance within the context of full-state-feedback control design. In the present paper, we further extend the approach of [9] to obtain a series of results for analyzing both robust stability and performance. As will be seen, these results also provide substantial unification of more recent results pertaining to robust stability alone.

To illustrate the basis for our approach, consider the system

$$\dot{x}(t) = (A + \Delta A)x(t) + Dw(t), \qquad t \in [0, \infty), \quad x(0) = 0, \tag{1.1}$$

$$y(t) = Ex(t), \tag{1.2}$$

where $x(t)$ is an n-vector, A is an $n \times n$ matrix denoting the nominal dynamics matrix, ΔA denotes an uncertain perturbation of A belonging to a specified set $\mathcal{U}$, $Dw(t)$ is (for now) a white noise signal of intensity $V \triangleq DD^T$, and $y(t)$ is a q-vector of outputs. System (1.1), (1.2) may, for example, denote a control system in closed-loop configuration.

For the system (1.1) the performance measure involves the steady-state second moment of the outputs $y(t)$. In practice the diagonal elements of the second moment are measures of the ability of the external disturbances $Dw(t)$ to excite specified states. In the presence of uncertainties ΔA, it is of interest to determine the *worst-case* steady-state values of the second moments of selected states. Thus, we define the scalar performance criterion

$$J_S(\mathcal{U}) \triangleq \sup_{\Delta A \in \mathcal{U}} \limsup_{t \to \infty} \mathbb{E}\{y^T(t)y(t)\}, \tag{1.3}$$

where $\mathbb{E}$ denotes expectation and lim sup is a technicality to ensure that $J_S(\mathcal{U})$ is a well-defined quantity even when $A + \Delta A$ has eigenvalues in the closed right half plane. To evaluate (1.3) define the second-moment matrix

$$Q(t) \triangleq \mathbb{E}[x(t)x^T(t)],$$

which satisfies the Lyapunov differential equation

$$\dot{Q}_{\Delta A}(t) = (A + \Delta A)Q_{\Delta A}(t) + Q_{\Delta A}(t)(A + \Delta A)^T + V, \tag{1.4}$$

so that (1.3) becomes

$$J_S(\mathcal{U}) = \sup_{\Delta A \in \mathcal{U}} \limsup_{t \to \infty} \operatorname{tr} Q_{\Delta A}(t)R, \tag{1.5}$$

where $R \triangleq E^T E$. To guarantee both robust stability and performance we consider modified algebraic Lyapunov equations of the form

$$0 = AQ + QA^T + \Omega(Q) + V, \tag{1.6}$$

where $\Omega(\cdot)$ is a matrix operator satisfying

$$\Delta A Q + Q \Delta A^T \leqq \Omega(Q) \tag{1.7}$$

for all $\Delta A \in \mathscr{U}$ and all nonnegative-definite matrices Q. The ordering in (1.7) is defined with respect to the cone of nonnegative-definite matrices. Our results are based on the following robust stability and performance result (for convenience, assume that V is positive definite). If there exists a positive-definite solution Q to (1.6), where $\Omega(\cdot)$ satisfies (1.7), then $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$ and, furthermore,

$$J_S(\mathscr{U}) \leqq \operatorname{tr} QR. \tag{1.8}$$

The robust stability result is a direct consequence of Lyapunov theory, while the performance bound (1.8) follows from the fact that since $A + \Delta A$ is asymptotically stable, $Q_{\Delta A} \triangleq \lim_{t \to \infty} Q_{\Delta A}(t)$ exists, is independent of $Q_{\Delta A}(0)$, and satisfies

$$0 = (A + \Delta A) Q_{\Delta A} + Q_{\Delta A} (A + \Delta A)^T + V. \tag{1.9}$$

Now subtracting (1.9) from (1.6) yields

$$0 = (A + \Delta A)(Q - Q_{\Delta A}) + (Q - Q_{\Delta A})(A + \Delta A)^T + \Omega(Q) - (\Delta A Q + Q \Delta A^T) + V,$$

which, by (1.7) and the fact that $A + \Delta A$ is stable, implies

$$Q_{\Delta A} \leqq Q. \tag{1.10}$$

Now (1.5) and (1.10) yield the bound (1.8).

Since the ordering induced by the cone of nonnegative-definite matrices is only a partial ordering, it should not be expected that there exists an operator $\Omega(\cdot)$ satisfying (1.7), which is a least upper bound. Indeed, there are many alternative definitions for the bound $\Omega(\cdot)$. To illustrate some of these alternatives, assume for convenience that ΔA is of the form

$$\Delta A = \sigma_1 A_1, \qquad |\sigma_1| \leqq \delta_1, \tag{1.11}$$

where σ_1 is an uncertain real scalar parameter assumed only to satisfy the stated bounds, and A_1 is a known matrix denoting the structure of the parametric uncertainty. The bound $\Omega(\cdot)$ utilized in [9] and [12] for full-state-feedback design was chosen to be

$$\Omega(Q) = \delta_1 |A_1 Q + Q A_1^T|, \tag{1.12}$$

where $|\cdot|$ denotes the nonnegative-definite matrix obtained by replacing each eigenvalue by its absolute value. More recently, the quadratic (in Q) bound

$$\Omega(Q) = \delta_1 [A_L A_L^T + Q A_R^T A_R Q] \tag{1.13}$$

has been considered, where A_L, A_R are a factorization of A_1 of the form $A_1 = A_L A_R$. Bound (1.13) was studied in [29] for robustness analysis and in [17], [25], [28], [30], [33], and [36] for robust controller synthesis. A third bound that has also been considered is the linear (in Q) bound

$$\Omega(Q) = \delta_1 [\alpha Q + \alpha^{-1} A_1 Q A_1^T], \tag{1.14}$$

where α is an arbitrary positive scalar. As shown in [33], bound (1.14) arises from a multiplicative white noise model with exponential disturbance weighting. Control-design applications of bound (1.14) are given in [23], [27], [33]–[35]. The principal contribution of the present paper is thus a unified development of bounds (1.12)–(1.14) for both robust stability and performance analysis. In addition, we present a systematic

approach that pays careful attention to the structure of the uncertainty set $\mathscr{U}$. For example, we show that bound (1.12) guarantees stability over a rectangular uncertainty set while (1.14) is most naturally associated with an ellipsoidal region. Furthermore, to provide a methodical development, we identify three classes of bounds (Types I, II, and III) that operate by exploiting, respectively, the symmetry of $\Delta AQ + Q\Delta A^T$, the structure of Q, and the structure of ΔA. This approach clarifies the relationships among different bounds and suggests several new bounds. The principal goal in this regard is to demonstrate the richness of quadratic Lyapunov bounds to stimulate future developments.

Finally, the present paper also considers an alternative cost functional for robust performance analysis. Specifically, in place of white noise disturbances, we reinterpret $w(t)$ in (1.1) as a deterministic L_2 signal as in H_∞ theory [6]. By imposing an L_∞ norm on the output $y(t)$ (rather than an L_2 norm as in H_∞ theory), the corresponding performance measure is given by (see [38])

$$J_D(\mathscr{U}) = \sup_{\Delta A \in \mathscr{U}} \limsup_{t \to \infty} \lambda_{\max}(Q_{\Delta A}(t)R), \tag{1.15}$$

in contrast to (1.5). Both performance measures $J_S(\mathscr{U})$ and $J_D(\mathscr{U})$ are considered in the paper.

The contents of the paper are as follows. After summarizing notation later in this section, the Robust Stability Problem, Stochastic Robust Performance Problem, and Deterministic Robust Performance Problem are introduced in § 2. In § 3 the basic result guaranteeing robust stability and performance (Theorem 3.1) is stated. This result is easily stated and forms the basis for all later developments. A dual version of Theorem 3.1 (Theorem 4.1) provides additional sufficient conditions and clarifies connections to traditional robust stability results. The bound $\Omega(\cdot)$ and its dual $\Lambda(\cdot)$ are given concrete forms in § 5. In § 6, the bounds of § 5 are merged with Theorem 3.1 to yield the main results guaranteeing robust stability and performance (Theorems 6.1–6.5) via modified Lyapunov equations. In § 7 we analyze the modified Lyapunov equations with regard to existence, uniqueness, and monotonicity of solutions. Additional bounds are derived in § 8 by utilizing a recursive substitution technique, while both upper and lower bounds are obtained in § 9. Finally, illustrative examples are considered in §§ 10 and 11.

Notation. Note: All matrices have real entries.

$\mathbb{R}$, $\mathbb{R}^{r\times s}$, $\mathbb{R}^r$, $\mathbb{E}$	real numbers, $r \times s$ real matrices, $\mathbb{R}^{r\times 1}$, expectation,
I_r	$r \times r$ identity matrix,
asymptotically stable matrix	matrix with eigenvalues in open left half plane,
$\mathbb{S}^r$	$r \times r$ symmetric matrices,
$\mathbb{N}^r$	$r \times r$ symmetric nonnegative-definite matrices,
$\mathbb{P}^r$	$r \times r$ symmetric positive-definite matrices,
$Z_1 \geqq Z_2$	$Z_1 - Z_2 \in \mathbb{N}^r$, $Z_1, Z_2 \in \mathbb{S}^r$,
$Z_1 > Z_2$	$Z_1 - Z_2 \in \mathbb{P}^r$, $Z_1, Z_2 \in \mathbb{S}^r$,
tr Z, Z^T	trace of Z, transpose of Z,
$\lambda_i(Z)$	eigenvalue of matrix Z,
$\lambda_{\max}(Z)$	maximum eigenvalue of matrix Z having real spectrum,
$\|\cdot\|_2$	Euclidean vector norm,
$\|\cdot\|_s$	spectral matrix norm (largest singular value),
$\|\cdot\|_F$	Frobenius matrix norm.

2. Robust stability and performance problems. Let $\mathscr{U} \subset \mathbb{R}^{n\times n}$ denote a set of perturbations ΔA of a given nominal dynamics matrix $A \in \mathbb{R}^{n\times n}$. Throughout the paper it is assumed that A is asymptotically stable and that $0 \in \mathscr{U}$. We begin by considering the question of whether or not $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$.

ROBUST STABILITY PROBLEM. Determine whether the linear system

$$\dot{x}(t) = (A + \Delta A)x(t), \qquad t \in [0, \infty), \tag{2.1}$$

is asymptotically stable for all $\Delta A \in \mathscr{U}$.

To consider the problem of robust performance it is necessary to introduce external disturbances. In this paper we consider both stochastic and deterministic disturbance models. The stochastic disturbance model involves white noise signals as in standard LQG theory, whereas the deterministic disturbance model involves L_2 signals as in H_∞ theory [6]. By defining an appropriate performance measure for each disturbance class it turns out that we can provide a simultaneous treatment of both cases.

We first consider the case of stochastic disturbances. In this case the robust performance problem concerns the worst-case magnitude of the expected value of a quadratic form involving outputs $y(t) = Ex(t)$, where $E \in \mathbb{R}^{q\times n}$, when the system is subjected to a standard white noise disturbance $w(t) \in \mathbb{R}^d$ with weighting $D \in \mathbb{R}^{n\times d}$.

STOCHASTIC ROBUST PERFORMANCE PROBLEM. For the disturbed linear system

$$\dot{x}(t) = (A + \Delta A)x(t) + Dw(t), \qquad t \in [0, \infty), \quad x(0) = 0, \tag{2.2}$$

$$y(t) = Ex(t), \tag{2.3}$$

where $w(\cdot)$ is a zero-mean d-dimensional white noise signal with intensity I_d, determine a performance bound β_S satisfying

$$J_S(\mathscr{U}) \triangleq \sup_{\Delta A \in \mathscr{U}} \limsup_{t \to \infty} \mathbb{E}\{\|y(t)\|_2^2\} \leqq \beta_S. \tag{2.4}$$

The system (2.2), (2.3) may denote, for example, a control system in closed-loop configuration subjected to external white noise disturbances for which $y(t)$ may be the state regulation error. Such specializations are not required for this development, however.

Of course, since D and E may be rank deficient, there may be cases in which a finite performance bound β_S satisfying (2.4) exists while (2.1) is not asymptotically stable over $\mathscr{U}$. In practice, however, robust performance is mainly of interest when (2.1) is robustly stable. In this case the performance $J_S(\mathscr{U})$ is given in terms of the steady-state second moment of the state. The following result from linear system theory will be useful. For convenience define the $n \times n$ nonnegative-definite matrices

$$R \triangleq E^T E, \qquad V \triangleq DD^T.$$

LEMMA 2.1. *Suppose $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$. Then*

$$J_S(\mathscr{U}) = \sup_{\Delta A \in \mathscr{U}} \operatorname{tr} Q_{\Delta A} R, \tag{2.5}$$

where the $n \times n$ matrix $Q_{\Delta A} \triangleq \lim_{t\to\infty} \mathbb{E}[x(t)x^T(t)]$ is given by

$$Q_{\Delta A} = \int_0^\infty e^{(A+\Delta A)t} V e^{(A+\Delta A)^T t}\, dt, \tag{2.6}$$

which is the unique, nonnegative-definite solution to

$$0 = (A + \Delta A)Q_{\Delta A} + Q_{\Delta A}(A + \Delta A)^T + V. \tag{2.7}$$

To state the Deterministic Robust Performance Problem some additional notation is required. For a measurable function $z\colon [0, \infty) \to \mathbb{R}^r$ define

$$\|z(\cdot)\|_{2,2} \triangleq \left\{\int_0^\infty \|z(t)\|_2^2 \, dt\right\}^{1/2}, \tag{2.8}$$

which is an L_2 function norm with a Euclidean spatial norm, and define

$$\|z(\cdot)\|_{\infty,2} \triangleq \text{ess. } \sup_{t \in [0,\infty)} \|z(t)\|_2,$$

which is an L_∞ function norm with a Euclidean spatial norm. We now reconsider (2.2) with $w(\cdot)$ interpreted as a square-integrable function. In this case the robust performance problem concerns the worst-case L_∞ norm of the output $y(t)$.

DETERMINISTIC ROBUST PERFORMANCE PROBLEM. For the disturbed linear system (2.2), (2.3), where $\|w(\cdot)\|_{2,2} \leqq 1$, determine a performance bound β_D satisfying

$$J_D(\mathscr{U}) \triangleq \sup_{\Delta A \in \mathscr{U}} \sup_{\|w(\cdot)\|_{2,2} \leqq 1} \|y(\cdot)\|_{\infty,2}^2 \leqq \beta_D. \tag{2.9}$$

The performance measure $J_D(\mathscr{U})$ in (2.9) is given by the following result.

LEMMA 2.2. *Suppose $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$. Then*

$$J_D(\mathscr{U}) = \sup_{\Delta A \in \mathscr{U}} \lambda_{\max}(Q_{\Delta A} R), \tag{2.10}$$

where $Q_{\Delta A}$ is the unique, nonnegative-definite solution to (2.7).

Proof. The result is an immediate consequence of Theorem 1(b) of [38]. □

Remark 2.1. Although $J_S(\mathscr{U})$ and $J_D(\mathscr{U})$ arise from different mathematical settings they are quite similar in form. Note that in general $J_D(\mathscr{U}) \leqq J_S(\mathscr{U})$, and $J_D(\mathscr{U}) = J_S(\mathscr{U})$ if rank $R = 1$.

Remark 2.2. In Lemma 2.2 $Q_{\Delta A}$ can be viewed as the controllability Gramian for the pair $(A + \Delta A, D)$ rather than the state covariance. Note that $Q_{\Delta A}$ is independent of $x(0)$ and $Q_{\Delta A}(0)$.

Remark 2.3. The stochastic performance measure $J_S(\mathscr{U})$ given by (2.5) can also be written as

$$J_S(\mathscr{U}) = \sup_{\Delta A \in \mathscr{U}} \int_0^\infty \|Ee^{(A+\Delta A)t} D\|_F^2 \, dt, \tag{2.11}$$

which involves the L_2 norm of the impulse response of (2.2), (2.3). This stochastic performance measure can thus also be given a deterministic interpretation by letting $w(t)$ denote impulses at time $t = 0$. For details of this formulation see [46, p. 331].

In the present paper our approach is to obtain robust stability as a consequence of sufficient conditions for robust performance. Such conditions are developed in the following sections.

3. Sufficient conditions for robust stability and performance. The key step in obtaining robust stability and performance is to bound the uncertain terms $\Delta A Q + Q \Delta A^T$ in the Lyapunov equation (2.7) by means of a function $\Omega(Q)$. The nonnegative-definite solution Q of this modified Lyapunov equation is then guaranteed to be an upper bound for $Q_{\Delta A}$. The following easily proved result is fundamental and forms the basis for all later developments. The result is based on Lyapunov function theory as applied to linear systems. For our purposes, a suitable statement of this result is given by Lemma 12.2 of [39]. Essentially this result states that if the matrix equation $0 = \Phi F + F\Phi^T + SS^T$ has a solution $F \geqq 0$ and (Φ, S) is stabilizable, then Φ is an asymptotically stable matrix. Of

course, (Φ, S) is stabilizable (regardless of Φ) if S has full row rank, and we note (see [39, Thm. 3.6]) that if (Φ, S) is stabilizable then so is $(\Phi, [SS^T + H]^{1/2})$ for all nonnegative-definite matrices H.

THEOREM 3.1. *Let* $\Omega : \mathbb{N}^n \to \mathbb{N}^n$ *be such that*

$$\Delta A Q + Q \Delta A^T \leqq \Omega(Q), \qquad \Delta A \in \mathscr{U}, \quad Q \in \mathbb{N}^n, \tag{3.1}$$

and suppose there exists $Q \in \mathbb{N}^n$ *satisfying*

$$0 = AQ + QA^T + \Omega(Q) + V. \tag{3.2}$$

Then

$$(A + \Delta A, D) \text{ is stabilizable}, \qquad \Delta A \in \mathscr{U}, \tag{3.3}$$

if and only if

$$A + \Delta A \text{ is asymptotically stable}, \qquad \Delta A \in \mathscr{U}. \tag{3.4}$$

In this case,

$$Q_{\Delta A} \leqq Q, \qquad \Delta A \in \mathscr{U}, \tag{3.5}$$

where $Q_{\Delta A} \in \mathbb{N}^n$ *is given by* (2.7), *and*

$$J_S(\mathscr{U}) \leqq \operatorname{tr} QR, \tag{3.6}$$

$$J_D(\mathscr{U}) \leqq \lambda_{\max}(QR). \tag{3.7}$$

In addition, if there exists $\Delta A \in \mathscr{U}$ *such that* $(A + \Delta A, D)$ *is controllable, then* Q *is positive definite.*

Proof. We stress that in (3.1), Q denotes an arbitrary element of $\mathbb{N}^n$, whereas in (3.2) Q denotes a specific solution of the modified Lyapunov equation. This minor abuse of notation considerably simplifies the presentation. Now note that for all $\Delta A \in \mathbb{R}^{n \times n}$, (3.2) is equivalent to

$$0 = (A + \Delta A)Q + Q(A + \Delta A)^T + \Omega(Q) - (\Delta A Q + Q \Delta A^T) + V. \tag{3.8}$$

Hence, by assumption, (3.8) has a solution $Q \in \mathbb{N}^n$ for all $\Delta A \in \mathbb{R}^{n \times n}$. If ΔA is restricted to the set $\mathscr{U}$ then, by (3.1), $\Omega(Q) - (\Delta A Q + Q \Delta A^T)$ is nonnegative definite. Thus if the stabilizability condition (3.3) holds for all $\Delta A \in \mathscr{U}$, then it follows from Theorem 3.6 of [39] that $(A + \Delta A, [V + \Omega(Q) - (\Delta A Q + Q \Delta A^T)]^{1/2})$ is stabilizable for all $\Delta A \in \mathscr{U}$. It now follows from (3.8) and Lemma 12.2 of [39] that $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$. Conversely, if $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$, then (3.3) is immediate. Next, subtracting (2.7) from (3.8) yields

$$0 = (A + \Delta A)(Q - Q_{\Delta A}) + (Q - Q_{\Delta A})(A + \Delta A)^T + \Omega(Q) - (\Delta A Q + Q \Delta A^T), \qquad \Delta A \in \mathscr{U},$$

or, equivalently, since $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$

$$Q - Q_{\Delta A} = \int_0^\infty e^{(A + \Delta A)t} [\Omega(Q) - (\Delta A Q + Q \Delta A^T)] e^{(A + \Delta A)^T t} \, dt \geqq 0, \qquad \Delta A \in \mathscr{U}, \tag{3.9}$$

which implies (3.5). The performance bound (3.6) is now an immediate consequence of (2.5) and (3.5). To prove (3.7) we note that if $0 \leqq M_1 \leqq M_2$ then $\lambda_{\max}(M_1) \leqq \lambda_{\max}(M_2)$ (see, e.g., Corollary 7.7.4 of [40]). Thus

$$\begin{aligned} J_D(\mathscr{U}) &= \sup_{\Delta A \in \mathscr{U}} \lambda_{\max}(Q_{\Delta A} R) = \sup_{\Delta A \in \mathscr{U}} \lambda_{\max}(E Q_{\Delta A} E^T) \\ &\leqq \lambda_{\max}(EQE^T) = \lambda_{\max}(QR). \end{aligned} \tag{3.10}$$

Finally, it follows from (3.8) that if $(A + \Delta A, D)$ is controllable for some $\Delta A \in \mathscr{U}$, then the controllability Gramian Q for the pair

$$(A+\Delta A,[V+\Omega(Q)-(\Delta AQ+Q\Delta A^T)]^{1/2})$$

is positive definite. □

For convenience we shall say that $\Omega(\cdot)$ *bounds* $\mathscr{U}$ if (3.1) is satisfied. To apply Theorem 3.1, we first specify a function $\Omega(\cdot)$ and an uncertainty set $\mathscr{U}$ such that $\Omega(\cdot)$ bounds $\mathscr{U}$. If the existence of a nonnegative-definite solution Q to (3.2) can be determined analytically or numerically and (3.3) is satisfied, then robust stability is guaranteed and the performance bounds (3.6), (3.7) can be computed. We can then enlarge $\mathscr{U}$, modify $\Omega(\cdot)$, and again attempt to solve (3.2). If, however, a nonnegative-definite solution to (3.2) cannot be determined, then $\mathscr{U}$ must be decreased in size until (3.2) is solvable. For example, $\Omega(\cdot)$ can be replaced by $\varepsilon\Omega(\cdot)$ to bound $\varepsilon\mathscr{U}$, where $\varepsilon > 1$ enlarges $\mathscr{U}$ and $\varepsilon < 1$ shrinks $\mathscr{U}$. Of course, the actual range of uncertainty that can be bounded depends on the nominal matrix A, the function $\Omega(\cdot)$, and the structure of $\mathscr{U}$. In § 5 the uncertainty set $\mathscr{U}$ and bound $\Omega(\cdot)$ satisfying (3.1) are given concrete forms. We complete this section with several observations.

Remark 3.1. If only robust stability is of interest, then the noise intensity V need not have physical significance. In this case we may set $D = I_n$ to satisfy (3.3).

Remark 3.2. Since A is asymptotically stable, Q satisfying (3.2) is given by

$$Q=\int_0^\infty e^{At}[\Omega(Q)+V]e^{A^Tt}\,dt, \tag{3.11}$$

or, equivalently,

$$Q=\int_0^\infty e^{At}\bar{\Omega}(Q)e^{A^Tt}\,dt+Q_0, \tag{3.12}$$

where $Q_0 \in \mathbb{N}^n$ is defined by

$$Q_0 \triangleq \int_0^\infty e^{At}Ve^{A^Tt}\,dt \tag{3.13}$$

and satisfies

$$0=AQ_0+Q_0A^T+V. \tag{3.14}$$

Note that $Q_0 \leqq Q$ and that the nominal performances $J_S(\{0\})$ and $J_D(\{0\})$ are given by $\operatorname{tr} Q_0R$ and $\lambda_{\max}(Q_0R)$, respectively.

Remark 3.3. Using (3.11) it is also useful to note that the bound for $J_S(\mathscr{U})$ given by (3.6) can be written as

$$\operatorname{tr} QR=\operatorname{tr}\int_0^\infty e^{At}[\Omega(Q)+V]e^{A^Tt}\,dtR=\operatorname{tr} P_0[\Omega(Q)+V], \tag{3.15}$$

where $P_0 \in \mathbb{N}^n$ is defined by

$$P_0 \triangleq \int_0^\infty e^{A^Tt}Re^{At}\,dt \tag{3.16}$$

and satisfies

$$0=A^TP_0+P_0A+R. \tag{3.17}$$

The bound tr $P_0[\Omega(Q) + V]$ can be viewed as a dual formulation of the bound tr QR since the roles of A and A^T are reversed. Dual bounds are developed in the following section. Note that tr $Q_0 R$ = tr $P_0 V$.

Remark 3.4. If $\Omega(\cdot)$ bounds $\mathcal{U}$ then clearly $\Omega(\cdot)$ bounds the convex hull of $\mathcal{U}$. Hence, only convex uncertainty sets $\mathcal{U}$ need be considered. Next, we shall later use the obvious fact that if $\Omega'(\cdot)$ bounds $\mathcal{U}'$ and $\Omega''(\cdot)$ bounds $\mathcal{U}''$, then $\Omega'(\cdot) + \Omega''(\cdot)$ bounds $\mathcal{U}' + \mathcal{U}''$. Hence if $\mathcal{U}$ can be decomposed additively then it suffices to bound each component separately. Finally, if $\Omega(\cdot)$ bounds $\mathcal{U}$ and there exists $\Omega' : \mathbb{N}^n \to \mathbb{N}^n$ such that $\Omega(Q) \leqq \Omega'(Q)$ for all $Q \in \mathbb{N}^n$, then $\Omega'(\cdot)$ also bounds $\mathcal{U}$. That is, any *overbound* $\Omega'(\cdot)$ for $\Omega(\cdot)$ also bounds $\mathcal{U}$. Of course, as we shall see, it is quite possible that an overbound $\Omega'(\cdot)$ for $\Omega(\cdot)$ may actually bound a set $\mathcal{U}'$ that is larger than the "original" uncertainty set $\mathcal{U}$.

4. Dual sufficient conditions for robust stability and performance. As noted in Remark 3.3, the performance bound tr QR given by (3.6) can be expressed equivalently in terms of a dual variable P_0 for which the roles of A and A^T are reversed. Using a similar technique, additional conditions for robust stability and performance can be obtained by developing a dual version of Theorem 3.1. A prime motivation for developing such dual bounds is to draw connections with previous results in the literature relating to robust stability. Specifically, we shall show that traditional robust stability techniques based on the quadratic Lyapunov function $V(x) = x^T P x$ correspond to dual conditions. Robust performance bounds within the dual formulation, however, are difficult to motivate without first developing the primal performance bounds as was done in the previous section. In addition, the dual bounds may, for certain problems, yield larger stability regions and sharper performance bounds than the primal bounds.

LEMMA 4.1. *Suppose $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathcal{U}$. Then*

$$J_S(\mathcal{U}) = \sup_{\Delta A \in \mathcal{U}} \text{tr } P_{\Delta A} V, \tag{4.1}$$

where $P_{\Delta A} \in \mathbb{R}^{n \times n}$ is the unique, nonnegative-definite solution to

$$0 = (A + \Delta A)^T P_{\Delta A} + P_{\Delta A}(A + \Delta A) + R. \tag{4.2}$$

Proof. It need only be noted that

$$\text{tr } Q_{\Delta A} R = \text{tr} \int_0^\infty e^{(A+\Delta A)t} V e^{(A+\Delta A)^T t}\, dt R = \text{tr } P_{\Delta A} V,$$

where

$$P_{\Delta A} \triangleq \int_0^\infty e^{(A+\Delta A)^T t} R e^{(A+\Delta A)t}\, dt$$

satisfies (4.2). □

The proof of Lemma 4.1 relies on the fact that tr $Q_{\Delta A} R$ = tr $P_{\Delta A} V$. However, it is not necessarily true that $\lambda_{\max}(Q_{\Delta A} R) = \lambda_{\max}(P_{\Delta A} V)$ even when $\Delta A = 0$. For example, if

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}, \quad R = I_2, \quad V = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

then

$$Q_0 R = \begin{bmatrix} 1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{4} \end{bmatrix} \quad \text{and} \quad P_0 V = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

and thus $\lambda_{\max}(Q_0R) = (15 + \sqrt{145})/24$ and $\lambda_{\max}(P_0V) = (5 + \sqrt{17})/8$. Thus to obtain a suitable dual version of $J_D(\mathscr{U})$ we need to define a dual deterministic cost $\hat{J}_D(\mathscr{U})$, which is distinct from $J_D(\mathscr{U})$. This can be done if the disturbance signals are taken to be integrable rather than square integrable. Thus, for measurable $z : [0, \infty) \to \mathbb{R}^r$ define

$$\|z(\cdot)\|_{1,2} \triangleq \int_0^\infty \|z(t)\|_2 \, dt, \tag{4.3}$$

which is an L_1 function norm with a Euclidean spatial norm. The dual deterministic cost $\hat{J}_D(\mathscr{U})$ is thus defined by

$$\hat{J}_D(\mathscr{U}) \triangleq \sup_{\Delta A \in \mathscr{U}} \sup_{\|w(\cdot)\|_{1,2} \leq 1} \|y(\cdot)\|_{2,2}^2. \tag{4.4}$$

The following dual result follows from Theorem 1(a) of [38].

LEMMA 4.2. *Suppose $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}$. Then*

$$\hat{J}_D(\mathscr{U}) = \lambda_{\max}(P_{\Delta A}V), \tag{4.5}$$

where $P_{\Delta A} \in \mathbb{R}^{n \times n}$ is the unique, nonnegative-definite solution to (4.2).

The dual version of Theorem 3.1 can now be stated.

THEOREM 4.1. *Let $\Lambda : \mathbb{N}^n \to \mathbb{N}^n$ be such that*

$$\Delta A^T P + P \Delta A \leqq \Lambda(P), \qquad \Delta A \in \mathscr{U}, \quad P \in \mathbb{N}^n, \tag{4.6}$$

and suppose there exists $P \in \mathbb{N}^n$ satisfying

$$0 = A^T P + PA + \Lambda(P) + R. \tag{4.7}$$

Then

$$(E, A + \Delta A) \text{ is detectable}, \qquad \Delta A \in \mathscr{U}, \tag{4.8}$$

if and only if

$$A + \Delta A \text{ is asymptotically stable}, \qquad \Delta A \in \mathscr{U}. \tag{4.9}$$

In this case,

$$P_{\Delta A} \leqq P, \qquad \Delta A \in \mathscr{U}, \tag{4.10}$$

where $P_{\Delta A}$ is given by (4.2), *and*

$$J_S(\mathscr{U}) \leqq \operatorname{tr} PV, \tag{4.11}$$

$$\hat{J}_D(\mathscr{U}) \leqq \lambda_{\max}(PV). \tag{4.12}$$

In addition, if there exists $\Delta A \in \mathscr{U}$ such that $(E, A + \Delta A)$ is observable, then P is positive definite.

Proof. The proof is completely analogous to the proof of Theorem 3.1. □

Remark 4.1. Note that $\hat{J}_D(\mathscr{U}) \leqq J_S(\mathscr{U})$ and that $\hat{J}_D(\mathscr{U}) = J_S(\mathscr{U})$ if rank $V = 1$. Combining this fact with Remark 2.1, it follows that $J_D(\mathscr{U}) = \hat{J}_D(\mathscr{U})$ if both rank $R = 1$ and rank $V = 1$. In general, however, we should not expect that $J_D(\mathscr{U}) = \hat{J}_D(\mathscr{U})$.

It is quite possible that the *bounds* tr QR and tr PV for $J_S(\mathscr{U})$ given by (3.6) and (4.11) may be different in spite of the fact, as shown in the proof of Lemma 4.1, that tr $Q_{\Delta A}R$ = tr $P_{\Delta A}V$. That is, depending on $\Omega(\cdot)$ and $\Lambda(\cdot)$ either bound (3.6) or bound (4.11) may be better for a particular problem. In general, we have the following result.

PROPOSITION 4.1. *Let* $\Omega(\cdot)$, $\Lambda(\cdot)$, Q, *and* P *be as in Theorems* 3.1 *and* 4.1, *and let* Q_0 *and* P_0 *be given by* (3.13) *and* (3.16), *respectively. Then*

$$\text{tr } Q_0\Lambda(P) < \text{tr } P_0\Omega(Q) \Leftrightarrow \text{tr } QR > \text{tr } PV, \tag{4.13}$$

$$\text{tr } Q_0\Lambda(P) = \text{tr } P_0\Omega(Q) \Leftrightarrow \text{tr } QR = \text{tr } PV, \tag{4.14}$$

$$\text{tr } Q_0\Lambda(P) > \text{tr } P_0\Omega(Q) \Leftrightarrow \text{tr } QR < \text{tr } PV. \tag{4.15}$$

Proof. Note that

$$\text{tr } QR = \int_0^\infty e^{At}[\Omega(Q) + V]e^{A^Tt}\,dt\,R = \text{tr } P_0\Omega(Q) + \text{tr} \int_0^\infty e^{At}Ve^{A^Tt}\,dt\,R$$

and

$$\text{tr } PV = \text{tr} \int_0^\infty e^{A^Tt}[\Lambda(P) + R]e^{At}\,dt\,V = \text{tr } Q_0\Lambda(P) + \text{tr} \int_0^\infty e^{A^Tt}Re^{At}\,dt\,V$$

so that

$$\text{tr } QR - \text{tr } PV = \text{tr } P_0\Omega(Q) - \text{tr } Q_0\Lambda(P),$$

which yields (4.13)–(4.15). □

Remark 4.2. To draw connections with traditional Lyapunov theory, let R and V be positive definite and assume that there exists a positive-definite solution to (4.7). Then $V(x) \triangleq x^TPx$ satisfies $\dot{V}(x(t)) < 0$ for $x(\cdot)$ satisfying (2.1) and for all $\Delta A \in \mathscr{U}$. Thus $V(\cdot)$ is a Lyapunov function for (2.1) that guarantees robust asymptotic stability over $\mathscr{U}$.

5. Construction of the bounds $\Omega(\cdot)$ and $\Lambda(\cdot)$. As discussed in § 1, we consider three distinct classes of bounds $\Omega(\cdot)$ denoted by Type I, Type II, and Type III. Roughly speaking, these bounds exploit, respectively, the symmetry of the Lyapunov terms $\Delta AQ + Q\Delta A^T$, the structure of Q, and the structure of ΔA. The dual bounds $\Lambda(\cdot)$ can be constructed similarly by replacing Q and ΔA by P and ΔA^T. Hence these bounds will not be discussed separately. For convenience in discussing the set $\mathscr{U}$, we shall use the terms *rectangle* and *ellipse* to refer to closed regions bounded by such figures in multiple dimensions. As usual, a polytope is the convex hull of a finite number of points.

5.1. Type I bounds. We begin by constructing bounds $\Omega(\cdot)$ that exploit only the symmetry of the Lyapunov terms $\Delta AQ + Q\Delta A^T$. First we require the following well-known definition of a function of a symmetric matrix as an extension of a real-valued function (see, e.g., [40, p. 300]). Specifically, if $f\colon \mathbb{R} \to \mathbb{R}$, then (with a minor abuse of notation) $f\colon \mathbb{S}^n \to \mathbb{S}^n$ can be defined by setting

$$f(S) \triangleq Uf(D)U^T,$$

where $S = UDU^T$, U is orthogonal, D is real diagonal, and $f(D)$ is the diagonal matrix obtained by applying f to each diagonal element of D. Note that if f is the polynomial $f(x) = \sum_{i=0}^r a_ix^i$ then $f(S) = \sum_{i=0}^r a_iS^i$. Note also that if $f(x) = |x|$ then $f(S) = (S^2)^{1/2}$, where $(\cdot)^{1/2}$ denotes the (unique) nonnegative-definite square root. As in [41, p. 262], we use the notation $|S|$ to denote $(S^2)^{1/2}$. Finally, note that if $f\colon \mathbb{R} \to \mathbb{R}$ and $g\colon \mathbb{R} \to \mathbb{R}$ are such that $f(x) \leqq g(x)$, $x \in \mathbb{R}$, then $f(S) \leqq g(S)$, $S \in \mathbb{S}^n$.

As a concretization of the uncertainty set $\mathscr{U}$, consider the set

$$\mathscr{U}_1 \triangleq \left\{\Delta A \in \mathbb{R}^{n\times n}\colon \Delta A = \sum_{i=1}^p \sigma_iA_i,\ |\sigma_i| \leqq \delta_i,\ i = 1, \cdots, p\right\}, \tag{5.1}$$

where, for $i = 1, \cdots, p$: $A_i \in \mathbb{R}^{n\times n}$ is a given matrix denoting the structure of the parametric uncertainty, σ_i is a real uncertain parameter, and δ_i denotes the range of parameter uncertainty. Clearly, the multidimensional set of uncertain parameters $(\sigma_1, \cdots, \sigma_p)$ is the rectangle $[-\delta_1, \delta_1] \times \cdots \times [-\delta_p, \delta_p]$ and $\mathscr{U}_1$ is a symmetric polytope of matrices in $\mathbb{R}^{n\times n}$. Note that the symmetry of the uncertainty interval $[-\delta_i, \delta_i]$ entails no loss of generality since the nominal value of A can be redefined if necessary. Furthermore, it is also possible, without loss of generality, to define $\delta_i = 1$ by replacing A_i by $\delta_i A_i$. For clarity, however, we choose not to employ this scaling. We begin by considering the bound utilized by Chang and Peng in [9].

PROPOSITION 5.1. *The function*

$$\Omega_1(Q) \triangleq \sum_{i=1}^{p} \delta_i |A_iQ + QA_i^T| \tag{5.2}$$

bounds $\mathscr{U}_1$.

Proof. For $i = 1, \cdots, p$ and $|\sigma_i| \leqq \delta_i$,

$$\sigma_i(A_iQ + QA_i^T) \leqq |\sigma_i(A_iQ + QA_i^T)| = |\sigma_i|\,|A_iQ + QA_i^T| \leqq \delta_i |A_iQ + QA_i^T|.$$

Summing over i yields

$$\Delta AQ + Q\Delta A^T = \sum_{i=1}^{p} \sigma_i(A_iQ + QA_i^T) \leqq \sum_{i=1}^{p} \delta_i |A_iQ + QA_i^T|,$$

which implies (3.1) with $\Omega(\cdot) = \Omega_1(\cdot)$ and $\mathscr{U} = \mathscr{U}_1$. □

Remark 5.1. It is tempting to prove Proposition 5.1 by writing

$$\sum_{i=1}^{p} \sigma_i(A_iQ + QA_i^T) \leqq \left| \sum_{i=1}^{p} \sigma_i(A_iQ + QA_i^T) \right| \leqq \sum_{i=1}^{p} |\sigma_i(A_iQ + QA_i^T)|.$$

However, counterexamples show that the inequality $|M_1 + M_2| \leqq |M_1| + |M_2|$ is not generally true for arbitrary symmetric matrices M_1, M_2.

Remark 5.2. Because of its simplicity it is tempting to conjecture that $\Omega_1(\cdot)$ is the best bound for $\Delta AQ + Q\Delta A^T$ over the set $\mathscr{U}_1$. To show that this is not the case, let $Q = \frac{1}{2}I_2$, $p = 1$, $A_1 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$, and $\delta_1 = 1$. Then $\sigma_1(A_1Q + QA_1^T) \leqq \delta_1 |A_1Q + QA_1^T| = I_2$, $|\sigma_1| \leqq 1$. However, it is also true that

$$\sigma_1(A_1Q + QA_1^T) \leqq \begin{bmatrix} 2 & \frac{3}{2} \\ \frac{3}{2} & 2 \end{bmatrix}, \qquad |\sigma_1| \leqq 1.$$

Neither bound, however, is an overbound for the other. This is a consequence of the fact that the nonnegative-definite matrix ordering is only a partial order.

As mentioned earlier, an overbound for $\Omega_1(\cdot)$ will also bound $\mathscr{U}_1$. The following result is immediate.

LEMMA 5.1. *For* $i = 1, \cdots, p$, *let* $f_i : \mathbb{R} \to \mathbb{R}$ *satisfy*

$$f_i(x) \geqq |x|, \qquad x \in \mathbb{R}. \tag{5.3}$$

Then the function

$$\Omega_2(Q) \triangleq \sum_{i=1}^{p} \delta_i f_i(A_iQ + QA_i^T) \tag{5.4}$$

is an overbound for $\Omega_1(\cdot)$ *and hence also bounds* $\mathscr{U}_1$.

One particular choice of f_i satisfying (5.3) will be considered here, namely, the polynomial

$$f_i(x) = \tfrac{1}{4}\beta_i + \beta_i^{-1}x^2, \tag{5.5}$$

where β_i is an arbitrary positive constant. Thus $\Omega_2(\cdot)$ has the following specialization.

COROLLARY 5.1. *Let $\beta_1, \cdots, \beta_p$ be arbitrary positive constants. Then the function*

$$\Omega_3(Q) \triangleq \frac{1}{4}\sum_{i=1}^{p}\delta_i\beta_i I_n + \sum_{i=1}^{p}\left(\frac{\delta_i}{\beta_i}\right)(A_iQ + QA_i^T)^2 \tag{5.6}$$

is an overbound for $\Omega_1(\cdot)$ and hence also bounds $\mathcal{U}_1$.

Although overbounding $\Omega_1(\cdot)$ by $\Omega_3(\cdot)$ results in a looser bound for $\mathcal{U}_1$, it turns out that $\Omega_3(\cdot)$ actually bounds a set that is larger than $\mathcal{U}_1$. Specifically, in place of $\mathcal{U}_1$ consider

$$\mathcal{U}_2 \triangleq \left\{\Delta A \in \mathbb{R}^{n\times n}: \Delta A = \sum_{i=1}^{p}\sigma_i A_i, \sum_{i=1}^{p}\frac{\sigma_i^2}{\alpha_i^2} \leqq 1\right\}, \tag{5.7}$$

where $\alpha_1, \cdots, \alpha_p$ are given positive constants. Note that (5.7) replaces the rectangle of uncertain parameters $(\sigma_1, \cdots, \sigma_p)$ by an ellipse. Thus the set $\mathcal{U}_2$ of matrix perturbations is an ellipse of matrices in $\mathbb{R}^{n\times n}$ in contrast to the polytope $\mathcal{U}_1$. Of course, $\mathcal{U}_1 = \mathcal{U}_2$ if $p = 1$ and $\alpha_1 = \delta_1$. Again it is possible to take $\alpha_i = 1$ without loss of generality by replacing A_i by $\alpha_i A_i$. We again choose not to do this, however. The following result provides a convenient characterization of the relationship between the rectangle $\mathcal{U}_1$ and the ellipse $\mathcal{U}_2$.

PROPOSITION 5.2. *Suppose $\mathcal{U}_1$ is defined by the positive constants $\delta_1, \cdots, \delta_p$, and let $\mathcal{U}_2$ be characterized by*

$$\alpha_i = \left(\frac{\alpha\delta_i}{\beta_i}\right)^{1/2}, \qquad i = 1, \cdots, p, \tag{5.8}$$

where α is defined by

$$\alpha = \sum_{i=1}^{p}\delta_i\beta_i \tag{5.9}$$

and $\beta_1, \cdots, \beta_p$ are arbitrary positive constants. Then the ellipse

$$\left\{(\sigma_1, \cdots, \sigma_p): \sum_{i=1}^{p}\frac{\sigma_i^2}{\alpha_i^2} \leqq 1\right\}$$

circumscribes the rectangle $\{(\sigma_1, \cdots, \sigma_p): |\sigma_i| \leqq \delta_i, i = 1, \cdots, p\}$ and thus $\mathcal{U}_2$ contains $\mathcal{U}_1$. Furthermore, $\Omega_3(\cdot)$ actually bounds $\mathcal{U}_2$.

Proof. If $|\sigma_i| \leqq \delta_i$, $i = 1, \cdots, p$, then it follows from (5.8) and (5.9) that

$$\sum_{i=1}^{p}\frac{\sigma_i^2}{\alpha_i^2} = \alpha^{-1}\sum_{i=1}^{p}\frac{\beta_i\sigma_i^2}{\delta_i} \leqq \alpha^{-1}\sum_{i=1}^{p}\beta_i\delta_i = 1.$$

Thus the ellipse contains the rectangle. If, in addition, $(\sigma_1, \cdots, \sigma_p)$ is a vertex of the rectangle, i.e., $|\sigma_i| = \delta_i$, $i = 1, \cdots, p$, then $\sum_{i=1}^{p} \sigma_i^2/\alpha_i^2 = 1$, which corresponds to a point on the boundary of the ellipse. To show that $\Omega_3(\cdot)$ actually bounds $\mathscr{U}_2$ note that

$$0 \leqq \sum_{i=1}^{p} \left[\frac{1}{2}\left(\frac{\alpha^{1/2}\sigma_i}{\alpha_i}\right) I_n - \left(\frac{\alpha_i}{\alpha^{1/2}}\right)(A_iQ + QA_i^T) \right]^2$$

$$= \frac{\alpha}{4} \sum_{i=1}^{p} \left(\frac{\sigma_i^2}{\alpha_i^2}\right) I_n + \alpha^{-1} \sum_{i=1}^{p} \alpha_i^2 (A_iQ + QA_i^T)^2 - (\Delta AQ + Q\Delta A^T).$$

Since $\sum_{i=1}^{p} \sigma_i^2/\alpha_i^2 \leqq 1$ in $\mathscr{U}_2$, it follows that

$$\Delta AQ + Q\Delta A^T \leqq \frac{\alpha}{4} I_n + \alpha^{-1} \sum_{i=1}^{p} \alpha_i^2 (A_iQ + QA_i^T)^2.$$

Utilizing (5.8) and (5.9) to substitute for α and α_i yields (3.1) with $\Omega(\cdot) = \Omega_3(\cdot)$ and $\mathscr{U} = \mathscr{U}_2$. □

Proposition 5.2 shows that each choice of constants $\beta_1, \cdots, \beta_p > 0$ leads to a particular ellipse $\mathscr{U}_2$ that contains the polytope $\mathscr{U}_1$. Furthermore, $\Omega_3(\cdot)$, which by Corollary 5.1 bounds $\mathscr{U}_1$, actually bounds the larger set $\mathscr{U}_2$. For convenience, we now dispense with the constants $\beta_1, \cdots, \beta_p$ that relate the rectangle $\mathscr{U}_1$ to the ellipse $\mathscr{U}_2$ and we characterize $\Omega_3(\cdot)$ entirely in terms of $\alpha, \alpha_1, \cdots, \alpha_p$.

COROLLARY 5.2. *Let α be an arbitrary positive constant. Then the function*

$$\Omega_4(Q) \triangleq \frac{\alpha}{4} I_n + \alpha^{-1} \sum_{i=1}^{p} \alpha_i^2 (A_iQ + QA_i^T)^2 \tag{5.10}$$

bounds $\mathscr{U}_2$.

Remark 5.3. Within the context of Corollary 5.2, the positive constant α plays no role in defining the set $\mathscr{U}_2$, although $\Omega_4(\cdot)$ is guaranteed to bound $\mathscr{U}_2$ for all choices of α. It can be expected, however, that certain choices of α provide better bounds than other choices. This will be seen by example in § 10.

The following variation of $\Omega_4(\cdot)$ was suggested by D. C. Hyland.

PROPOSITION 5.3. *Let α be an arbitrary positive constant. Then, for $Q > 0$,*

$$\Omega_{4'}(Q) \triangleq \frac{\alpha}{2} Q + \frac{\alpha^{-1}}{2} \sum_{i=1}^{p} \alpha_i^2 [A_i^2 Q + A_i Q A_i^T + Q A_i^T Q^{-1} A_i Q + Q A_i^{2T}] \tag{5.10$'$}$$

bounds $\mathscr{U}_2$.

Proof. Note that

$$0 \leqq \sum_{i=1}^{p} \left[\frac{1}{2}\left(\frac{\alpha^{1/2}\sigma_i}{\alpha_i}\right) Q^{1/2} - \left(\frac{\alpha_i}{\alpha^{1/2}}\right)(A_iQ + QA_i^T)Q^{-1/2} \right]$$

$$\times \left[\frac{1}{2}\left(\frac{\alpha^{1/2}\sigma_i}{\alpha_i}\right) Q^{1/2} - \left(\frac{\alpha_i}{\alpha^{1/2}}\right)(A_iQ + QA_i^T)Q^{-1/2} \right]^T$$

$$= \frac{\alpha}{4} \sum_{i=1}^{p} \left(\frac{\sigma_i^2}{\alpha_i^2}\right) Q + \alpha^{-1} \sum_{i=1}^{p} \alpha_i^2 (A_iQ + QA_i^T)Q^{-1}(A_iQ + QA_i^T) - (\Delta AQ + Q\Delta A^T),$$

which yields the desired result. □

Remark 5.4. The bound $\Omega_{4'}(Q)$ is of interest since it involves terms that arise from a multiplicative white noise model with a Stratonovich correction. Specifically, the term

$A_iQA_i^T$ arises from an Ito model [33], whereas the terms A_i^2Q and QA_i^{2T} can be viewed as the shift $A \to A + \frac{1}{2}\sum_{i=1}^{p} A_i^2$ due to the Stratonovich interpretation of stochastic integration [43]. These terms have interesting ramifications in designing controllers for flexible structures [23].

5.2. Type II bounds. We now consider additional bounds for $\mathscr{U}$ that exploit the structure of Q. For these bounds the natural uncertainty set is given by $\mathscr{U}_2$.

PROPOSITION 5.4. *Let α be an arbitrary positive number and, for each $Q \in \mathbb{N}^n$, let $Q_1 \in \mathbb{R}^{n\times m}$ and $Q_2 \in \mathbb{R}^{m\times n}$ satisfy*

$$Q = Q_1Q_2. \tag{5.11}$$

Then the function

$$\Omega_5(Q) \triangleq \alpha Q_2^TQ_2 + \alpha^{-1}\sum_{i=1}^{p} \alpha_i^2 A_iQ_1Q_1^TA_i^T \tag{5.12}$$

bounds $\mathscr{U}_2$.

Proof. Note that

$$0 \leqq \sum_{i=1}^{p}\left[\left(\frac{\alpha^{1/2}\sigma_i}{\alpha_i}\right)Q_2^T - \left(\frac{\alpha_i}{\alpha^{1/2}}\right)A_iQ_1\right]\left[\left(\frac{\alpha^{1/2}\sigma_i}{\alpha_i}\right)Q_2^T - \left(\frac{\alpha_i}{\alpha^{1/2}}\right)A_iQ_1\right]^T$$

$$= \alpha\sum_{i=1}^{p}\left(\frac{\sigma_i^2}{\alpha_i^2}\right)Q_2^TQ_2 + \alpha^{-1}\sum_{i=1}^{p}\alpha_i^2A_iQ_1Q_1^TA_i^T - \sum_{i=1}^{p}\sigma_i(A_iQ + QA_i^T),$$

which, since $\sum_{i=1}^{p}\sigma_i^2/\alpha_i^2 \leqq 1$, yields (3.1) with $\Omega(\cdot) = \Omega_5(\cdot)$ and $\mathscr{U} = \mathscr{U}_2$. □

We consider three specializations of $\Omega_5(\cdot)$. Specifically, we set $m = n$ and define

$$Q_1 = Q, \qquad Q_2 = I_n, \tag{5.13}$$

$$Q_1 = Q_2 = Q^{1/2}, \tag{5.14}$$

$$Q_1 = I_n, \qquad Q_2 = Q. \tag{5.15}$$

COROLLARY 5.3. *Let α be an arbitrary positive number. Then the functions*

$$\Omega_6(Q) \triangleq \alpha I_n + \alpha^{-1}\sum_{i=1}^{p}\alpha_i^2A_iQ^2A_i^T, \tag{5.16}$$

$$\Omega_7(Q) \triangleq \alpha Q + \alpha^{-1}\sum_{i=1}^{p}\alpha_i^2A_iQA_i^T, \tag{5.17}$$

$$\Omega_8(Q) \triangleq \alpha Q^2 + \alpha^{-1}\sum_{i=1}^{p}\alpha_i^2A_iA_i^T \tag{5.18}$$

bound $\mathscr{U}_2$.

Remark 5.5. Note that the term $A_iQ^2A_i^T$ appearing in $\Omega_6(\cdot)$ also appears in $\Omega_4(\cdot)$. Furthermore, both $\Omega_4(\cdot)$ and $\Omega_6(\cdot)$ involve a term proportional to I_n. Despite these similarities, neither bound $\Omega_4(\cdot)$ nor $\Omega_6(\cdot)$ is an overbound for the other. Furthermore, the term $A_iQA_i^T$ appears in both $\Omega_7(\cdot)$ and $\Omega_{4'}(\cdot)$. However, neither $\Omega_7(\cdot)$ nor $\Omega_{4'}(\cdot)$ is an overbound for the other.

Remark 5.6. The bound $\Omega_7(\cdot)$ given by (5.17) has the distinction that it is linear in Q. This bound was originally studied in [27] for systems with multiplicative white noise and was shown to yield robust stability and performance in [33] and [35]. A similar bound was studied in [34].

Remark 5.7. By using (5.11) additional bounds can be developed. For example, by setting

$$Q_1 = Q^{1/4}, \qquad Q_2 = Q^{3/4}, \tag{5.19}$$

$\Omega_5(\cdot)$ becomes

$$\Omega_9(Q) = \alpha Q^{3/2} + \alpha^{-1} \sum_{i=1}^{p} \alpha_i^2 A_i Q^{1/2} A_i^T. \tag{5.20}$$

Remark 5.8. When $p = 1$ and α is replaced by $\alpha\alpha_1$, $\Omega_7(\cdot)$ becomes

$$\Omega_7'(Q) = \alpha_1[\alpha Q + \alpha^{-1} A_1 Q A_1^T].$$

A sum of such terms with $\alpha_i = \delta_i$ can be used to bound the smaller rectangular set $\mathscr{U}_1$. Similar remarks apply to $\Omega_6(\cdot)$, $\Omega_8(\cdot)$, and $\Omega_9(\cdot)$.

5.3. Type III bounds. We now consider bounds that exploit the structure of ΔA itself. It turns out that these bounds permit consideration of an uncertainty set $\mathscr{U}$ that is larger than $\mathscr{U}_2$. Specifically, define

$$\mathscr{U}_3 \triangleq \{\Delta A \in \mathbb{R}^{n\times n} : \Delta A = A_L A_R, A_L A_L^T \leqq M, A_R^T A_R \leqq N\}, \tag{5.21}$$

where $A_L \in \mathbb{R}^{n\times r}$ and $A_R \in \mathbb{R}^{r\times n}$ are uncertain matrices, r is an arbitrary positive integer, and M, $N \in \mathbb{N}^n$ are given uncertainty bounds. The bound $\Omega_{10}(\cdot)$ for $\mathscr{U}_3$ is given by the following result.

PROPOSITION 5.5. *Let α be an arbitrary positive constant. Then the function*

$$\Omega_{10}(Q) \triangleq \alpha^{-1} M + \alpha QNQ \tag{5.22}$$

bounds $\mathscr{U}_3$.

Proof. Note that

$$\begin{aligned} 0 &\leqq [\alpha^{-1/2} A_L - \alpha^{1/2} Q A_R^T][\alpha^{-1/2} A_L - \alpha^{1/2} Q A_R^T]^T \\ &= \alpha^{-1} A_L A_L^T + \alpha Q A_R^T A_R Q - [A_L A_R Q + Q(A_L A_R)^T] \\ &\leqq \alpha^{-1} M + \alpha QNQ - (\Delta A Q + Q \Delta A^T), \end{aligned}$$

which yields (3.1) with $\Omega(\cdot) = \Omega_{10}(\cdot)$ and $\mathscr{U} = \mathscr{U}_3$. □

Remark 5.9. The bound $\Omega_{10}(\cdot)$ was developed in [29] for robust analysis and independently in [25] and [28] for robust full-state feedback. Applications to fixed-order dynamic compensation are given in [36].

Remark 5.10. Without loss of generality we can set $\alpha = 1$ in (5.22) by replacing M and N by $\alpha^{-1} M$ and αN, respectively. Again for clarity we choose not to employ this scaling.

Note that $\Omega_8(\cdot)$ is of the form $\Omega_{10}(\cdot)$ with $M = \sum_{i=1}^{p} \alpha_i^2 A_i A_i^T$ and $N = I_n$. Thus $\Omega_8(\cdot)$ also bounds $\mathscr{U}_3$ for this choice of M and N. It turns out in this case that $\mathscr{U}_3$ is actually larger than $\mathscr{U}_2$. To see this consider the more general case in which M and N satisfy

$$\sum_{i=1}^{p} \alpha_i^2 A_i A_i^T \leqq M, \qquad I_n \leqq N. \tag{5.23}$$

In this case $\Omega_{10}(\cdot)$ is an overbound for $\Omega_8(\cdot)$ and thus bounds $\mathscr{U}_2$. As in the case of $\Omega_3(\cdot)$ overbounding $\Omega_1(\cdot)$, we should not be surprised to find that $\Omega_{10}(\cdot)$ with (5.23) actually bounds a set that is larger than $\mathscr{U}_2$. Indeed, we now show that $\mathscr{U}_2$ is actually a very special subset of $\mathscr{U}_3$ when M and N defining $\mathscr{U}_2$ satisfy (5.23).

PROPOSITION 5.6. *If M and N satisfy* (5.23) *then $\mathcal{U}_2$ is a subset of $\mathcal{U}_3$. Hence $\Omega_{10}(\cdot)$ also bounds $\mathcal{U}_2$.*

Proof. If $\Delta A \in \mathcal{U}_2$ then $\Delta A = \sum_{i=1}^{p} \sigma_i A_i$, where $\sum_{i=1}^{p} \sigma_i^2/\alpha_i^2 \leqq 1$. Alternatively, we can write $\Delta A = A_L A_R$, where $r = pn$ and

$$A_L = [\alpha_1 A_1 \cdots \alpha_p A_p], \qquad A_R = \begin{bmatrix} (\sigma_1/\alpha_1) I_n \\ \vdots \\ (\sigma_p/\alpha_p) I_n \end{bmatrix}. \tag{5.24}$$

Note that with M and N satisfying (5.23) and A_L and A_R defined by (5.24), it follows that $A_L A_L^T \leqq M$ and $A_R^T A_R \leqq N$. Thus $\Delta A \in \mathcal{U}_3$. □

The following result provides further conditions under which $\Omega_{10}(\cdot)$ bounds $\mathcal{U}_2$.

PROPOSITION 5.7. *Suppose $A_i = D_i E_i$, $i = 1, \cdots, p$, where $D_i \in \mathbb{R}^{n \times n_i}$ and $E_i \in \mathbb{R}^{n_i \times n}$, and suppose that*

$$\sum_{i=1}^{p} \alpha_i^2 D_i D_i^T \leqq M, \qquad \sum_{i=1}^{p} E_i^T E_i \leqq N. \tag{5.25}$$

Then $\mathcal{U}_2$ is a subset of $\mathcal{U}_3$ and thus $\Omega_{10}(\cdot)$ also bounds $\mathcal{U}_2$.

Proof. The result follows as in the proof Proposition 5.6. □

Remark 5.11. When $p = 1$, $A_1 = D_1 E_1$, $M = \alpha_1^2 D_1 D_1^T$, and $N = E_1^T E_1$, it is convenient to replace α by $\alpha\alpha_1$ so that $\Omega_{10}(\cdot)$ becomes

$$\Omega_{10'}(Q) = \alpha_1 [\alpha^{-1} D_1 D_1^T + \alpha Q E_1^T E_1 Q]. \tag{5.26}$$

In certain situations it is desirable to consider subsets of $\mathcal{U}_3$ of special structure. For example, define

$$\mathcal{U}_4 \triangleq \{ \Delta A \in \mathbb{R}^{n \times n} : \Delta A = D_0 A_L A_R E_0, \| A_L \|_s \leqq 1, \| A_R \|_s \leqq 1 \},$$

where $D_0 \in \mathbb{R}^{n \times n_1}$ and $E_0 \in \mathbb{R}^{n_2 \times n}$ are known matrices denoting the structure of the uncertainty, and $A_L \in \mathbb{R}^{n_1 \times r}$ and $A_R \in \mathbb{R}^{r \times n_2}$ are uncertain matrices [28]. Finer structure can be included within $\mathcal{U}_4$ by replacing $D_0 M N E_0$ by a sum of terms $D_i M_i N_i E_i$, where D_i, E_i are known and M_i, N_i are uncertain [36]. Note, however, that even though $\mathcal{U}_4$ is a proper subset of $\mathcal{U}_3$, the *form* of the bound $\Omega_{10}(\cdot)$ does not change. Thus such refinements render the bound $\Omega_{10}(\cdot)$ conservative with respect to $\mathcal{U}_4$ since the *larger* uncertainty set $\mathcal{U}_3$ is actually being bounded.

6. Robust stability and performance via modified Lyapunov equations. We now combine the principal results of §§ 3, 4, and 5 to obtain a series of conditions guaranteeing robust stability and performance. In particular, we focus on bounds Ω_1, Ω_4, Ω_6, Ω_7, and Ω_{10}. For simplicity we shall frequently assume that V is positive definite so that (3.3) is satisfied. In this case it follows that the solution Q of (3.2) is positive definite. Our first result is a corollary of Theorem 3.1 with $\Omega(\cdot) = \Omega_1(\cdot)$ and $\mathcal{U} = \mathcal{U}_1$.

THEOREM 6.1. *Let $V \in \mathbb{P}^n$, $\delta_1, \cdots, \delta_p > 0$, and suppose there exists $Q \in \mathbb{P}^n$ satisfying*

$$0 = AQ + QA^T + \sum_{i=1}^{p} \delta_i | A_i Q + Q A_i^T | + V. \tag{MLE1}$$

Then $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathcal{U}_1$, and

$$J_S(\mathcal{U}_1) \leqq \operatorname{tr} QR, \tag{6.1}$$

$$J_D(\mathcal{U}_1) \leqq \lambda_{\max}(QR). \tag{6.2}$$

For the next result define

$$A_\alpha \triangleq A + \frac{\alpha}{2} I_n \tag{6.3}$$

and

$$\gamma_i \triangleq \frac{\alpha_i^2}{\alpha}, \qquad i = 1, \cdots, p. \tag{6.4}$$

Setting $\Omega(\cdot) = \Omega_4(\cdot), \Omega_6(\cdot), \Omega_7(\cdot)$ and $\mathscr{U} = \mathscr{U}_2$ yields the following corollary of Theorem 3.1.

THEOREM 6.2. *Let $V \in \mathbb{P}^n$, $\alpha, \alpha_1, \cdots, \alpha_p > 0$, and suppose there exists $Q \in \mathbb{P}^n$ satisfying either*

$$0 = AQ + QA^T + \sum_{i=1}^{p} \gamma_i (A_i Q + Q A_i^T)^2 + \frac{\alpha}{4} I_n + V, \tag{MLE2}$$

$$0 = AQ + QA^T + \sum_{i=1}^{p} \gamma_i A_i Q^2 A_i^T + \alpha I_n + V, \tag{MLE3}$$

or

$$0 = A_\alpha Q + Q A_\alpha^T + \sum_{i=1}^{p} \gamma_i A_i Q A_i^T + V. \tag{MLE4}$$

Then $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}_2$, and

$$J_S(\mathscr{U}_2) \leqq \operatorname{tr} QR, \tag{6.5}$$

$$J_D(\mathscr{U}_2) \leqq \lambda_{\max}(QR). \tag{6.6}$$

Next we set $\Omega(\cdot) = \Omega_{10}(\cdot)$ and $\mathscr{U} = \mathscr{U}_3$.

THEOREM 6.3. *Let $V \in \mathbb{P}^n$, $\alpha > 0$, $M \in \mathbb{N}^n$, and $N \in \mathbb{N}^n$, and suppose there exists $Q \in \mathbb{P}^n$ satisfying*

$$0 = AQ + QA^T + \alpha QNQ + \alpha^{-1} M + V. \tag{MLE5}$$

Then $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathscr{U}_3$, and

$$J_S(\mathscr{U}_3) \leqq \operatorname{tr} QR, \tag{6.7}$$

$$J_D(\mathscr{U}_3) \leqq \lambda_{\max}(QR). \tag{6.8}$$

Remark 6.1. Note that (MLE5) is a Riccati equation. This is precisely the equation studied in [29].

Additional sufficient conditions can be obtained by considering "mixed" bounds. That is, we can construct modified Lyapunov equations by combining two or more different bounds. Although mixed bounds will not be considered further in this paper, we present one such result for illustrative purposes.

THEOREM 6.4. *Let $V \in \mathbb{P}^n$, $\alpha, \delta_1, \cdots, \delta_p > 0$, $M \in \mathbb{N}^n$, and $N \in \mathbb{N}^n$, and suppose there exists $Q \in \mathbb{P}^n$ satisfying*

$$0 = AQ + QA^T + \sum_{i=1}^{p} \delta_i |A_i Q + Q A_i^T| + \alpha QNQ + \alpha^{-1} M + V. \tag{MLE1, 5}$$

Then $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathcal{U}_1 + \mathcal{U}_3$, and

$$J_S(\mathcal{U}_1 + \mathcal{U}_3) \leqq \operatorname{tr} QR, \tag{6.9}$$

$$J_D(\mathcal{U}_1 + \mathcal{U}_3) \leqq \lambda_{\max}(QR). \tag{6.10}$$

As noted previously, the bound $\Lambda(\cdot)$ can readily be constructed by replacing ΔA by ΔA^T in the definitions of $\Omega_1(\cdot)$ through $\Omega_{10}(\cdot)$. Denote these bounds by $\Lambda_1(\cdot)$ through $\Lambda_{10}(\cdot)$, respectively. For illustration we state the dual of Theorem 6.1 involving $\Lambda_1(\cdot)$. The dual versions of (MLE1)–(MLE5) will be denoted by (MLED1)–(MLED5).

THEOREM 6.5. *Let $R \in \mathbb{P}^n$, $\delta_1, \cdots, \delta_p > 0$, and suppose there exists $P \in \mathbb{P}^n$ satisfying*

$$0 = A^T P + PA + \sum_{i=1}^{p} \delta_i |A_i^T P + PA_i| + R. \tag{MLED1}$$

Then $A + \Delta A$ is asymptotically stable for all $\Delta A \in \mathcal{U}_1$, and

$$J_S(\mathcal{U}_1) \leqq \operatorname{tr} PV, \tag{6.11}$$

$$\hat{J}_D(\mathcal{U}_1) \leqq \lambda_{\max}(PV). \tag{6.12}$$

It is reasonable to expect that the sufficient conditions given by Theorems 3.1 and 4.1 are generally different. For example, the modified Lyapunov equations and their duals need not both possess a solution, while the bounds tr QR and tr PV need not be equal. An exception is the case in which $\Omega(\cdot) = \Omega_7(\cdot)$ and $\Lambda(\cdot) = \Lambda_7(\cdot)$. Note that the dual of (MLE4) is given by

$$0 = A_\alpha^T P + PA_\alpha + \sum_{i=1}^{p} \gamma_i A_i^T PA_i + V. \tag{MLED4}$$

PROPOSITION 6.1. *Let $\alpha, \alpha_1, \cdots, \alpha_p > 0$ and assume there exist $Q, P \in \mathbb{N}^n$ satisfying* (MLE4) *and* (MLED4). *Then*

$$\operatorname{tr} QR = \operatorname{tr} PV. \tag{6.13}$$

Proof. Note that

$$\begin{aligned} \operatorname{tr} QR &= -\operatorname{tr} Q\left(A_\alpha^T P + PA_\alpha + \sum_{i=1}^{p} \gamma_i A_i^T PA_i\right) \\ &= -\operatorname{tr} P\left(A_\alpha Q + QA_\alpha^T + \sum_{i=1}^{p} \gamma_i A_i QA_i^T\right) \\ &= \operatorname{tr} PV. \end{aligned}$$

□

Remark 6.2. By setting $\Omega(\cdot) = \Omega_7(\cdot)$ and $\Lambda(\cdot) = \Lambda_7(\cdot)$ it follows from (4.14) that

$$\operatorname{tr} Q_0\left(\alpha P + \sum_{i=1}^{p} \gamma_i A_i^T PA_i\right) = \operatorname{tr} P_0\left(\alpha Q + \sum_{i=1}^{p} \gamma_i A_i QA_i^T\right). \tag{6.14}$$

7. Existence, uniqueness, and monotonicity of solutions to the modified Lyapunov equations. It is important to stress that the sufficient conditions for robustness given by Theorems 6.1–6.5 assume only that there exist nonnegative-definite solutions Q, P sat-

isfying the modified Lyapunov equations. Indeed, no *explicit* assumptions on the problem data A, V, R, and $\mathscr{U}$ were utilized for assuring robust stability and performance. In applying Theorems 6.1–6.5 to specific problems it thus suffices to show that a nonnegative-definite solution Q *exists* in order to obtain robust stability, while, for robust performance, the bounds (6.1), (6.2), (6.5)–(6.8) require explicit knowledge of Q. Thus, any computational method that yields a nonnegative-definite solution will suffice to guarantee both robust stability and performance.

Before considering the numerical solution of the modified Lyapunov equations, several relevant issues require discussion. For example, before seeking to compute solutions to (MLE1)–(MLE5) it would be desirable to determine a priori whether these equations actually possess nonnegative-definite solutions. For example, it may be useful to obtain sufficient and/or necessary conditions for the *existence* of nonnegative-definite solutions. Thus, if the sufficient conditions are satisfied then existence (and hence robustness) is assured, whereas if the necessary conditions are *not* satisfied then existence is ruled out. If, on the other hand, either the sufficient conditions are not satisfied or the necessary conditions *are* satisfied, then nothing can be surmised. Finally, such conditions need to be easily verifiable and reasonably nonconservative since otherwise it would be more prudent to attempt to numerically solve the modified Lyapunov equations themselves.

It is quite possible that at least some of the modified Lyapunov equations possess multiple nonnegative-definite solutions. In this case we may seek the minimal solution (i.e., the smallest with respect to the nonnegative-definite matrix ordering) to minimize the performance bounds. If multiple solutions exist, none of which is minimal, then the best bound would depend on the matrix R.

Since the matrix Q determines the performance bound, it is reasonable to expect Q to be *monotonic* in $\mathscr{U}$. That is, if $\mathscr{U}$ decreases in size, then the solution Q is more likely to exist while decreasing in the nonnegative-definite matrix ordering. For example, consider $\mathscr{U}'_1$ characterized by δ'_i, where $\delta'_i \leqq \delta_i$, $i = 1, \cdots, p$. Then we might expect $Q' \leqq Q$, where Q' is the solution to (MLE1) with δ_i replaced by δ'_i. Finally, monotonicity with respect to V should also be expected. Because of linearity, the analysis of bound $\Omega_7(\cdot)$ is simplest and it is possible to obtain necessary and sufficient conditions for the existence of solutions to (MLE4). The basic tool required is the Kronecker matrix algebra [42]. For convenience, define

$$\mathscr{A} \triangleq A_\alpha \oplus A_\alpha + \sum_{i=1}^{p} \gamma_i A_i \otimes A_i, \tag{7.1}$$

where $\otimes$ denotes the Kronecker product and $A_\alpha \oplus A_\alpha \triangleq A_\alpha \otimes I_n + I_n \otimes A_\alpha$ is the Kronecker sum.

PROPOSITION 7.1. *If $V \in \mathbb{N}^n$ and $\mathscr{A}$ is asymptotically stable, then there exists a unique $Q \in \mathbb{R}^{n \times n}$ satisfying* (MLE4), *and $Q \geqq 0$. Conversely, if for all $V \in \mathbb{N}^n$ there exists $Q \geqq 0$ satisfying* (MLE4), *then $\mathscr{A}$ is asymptotically stable.*

Proof. Since (MLE4) is equivalent to

$$Q = -\mathrm{vec}^{-1}[\mathscr{A}^{-1} \mathrm{vec}\, V], \tag{7.2}$$

existence and uniqueness hold. Here, vec and vec^{-1} denote the column-stacking operation [42] and its inverse. To prove that Q is nonnegative definite, we rewrite (7.2) as

$$Q = \int_0^\infty \mathrm{vec}^{-1}[e^{\mathscr{A}t} \mathrm{vec}\, V]\, dt \tag{7.3}$$

and show that the integrand is nonnegative-definite for all $t \in [0, \infty)$. (Note that the following argument for fixed $t \geqq 0$ does not require that $\mathscr{A}$ be stable.) Using the exponential product formula,[1] the exponential in (7.3) can be written as

$$e^{\mathscr{A}t} = \lim_{k \to \infty} \left\{ \exp\left[\frac{1}{k}(A_\alpha \oplus A_\alpha)t\right] \exp\left[\frac{1}{k}\sum_{i=1}^{p} \gamma_i (A_i \otimes A_i)t\right] \right\}^k. \tag{7.4}$$

For convenience, let S and N be $r \times r$ matrices with $N \geqq 0$. Since (see [42])

$$\operatorname{vec}^{-1}[(S \otimes S) \operatorname{vec} N] = SNS^T \geqq 0 \tag{7.5}$$

and

$$(S \otimes S)^k = S^k \otimes S^k, \tag{7.6}$$

it follows that

$$\operatorname{vec}^{-1}[e^{S \otimes S} \operatorname{vec} N] = \sum_{k=0}^{\infty} (k!)^{-1} S^k N S^{kT} \geqq 0. \tag{7.7}$$

Furthermore,

$$\operatorname{vec}^{-1}[e^{S \oplus S} \operatorname{vec} N] = \operatorname{vec}^{-1}[(e^S \otimes e^S) \operatorname{vec} N] = e^S N e^{S^T} \geqq 0. \tag{7.8}$$

Applying (7.7) and (7.8) alternately with (7.4) and using induction on k, it follows that the integrand of (7.3) is nonnegative definite. To prove the converse, note that it follows from (MLE4) that Q satisfies

$$Q = \operatorname{vec}^{-1}[e^{\mathscr{A}t} \operatorname{vec} Q] + \int_0^t \operatorname{vec}^{-1}[e^{\mathscr{A}s} \operatorname{vec} V]\, ds, \qquad t \in [0, \infty). \tag{7.9}$$

Since the integral term on the right-hand side of (7.9) is nonnegative definite, is bounded from above by Q, and $V \in \mathbb{N}^n$ is arbitrary, it follows that $\mathscr{A}$ is asymptotically stable. □

We now show that if $\mathscr{A}$ is asymptotically stable then actually A_α (and thus A) is asymptotically stable. This shows that the assumption that $\mathscr{A}$ is asymptotically stable is consistent with the original hypothesis that A is asymptotically stable.

PROPOSITION 7.2. *Assume $\mathscr{A}$ is asymptotically stable, let $\alpha_i' \in [0, \alpha_i]$, $i = 1, \cdots, p$, and define*

$$\mathscr{A}' \triangleq A_\alpha \oplus A_\alpha + \sum_{i=1}^{p} \left(\frac{\alpha_i'^2}{\alpha}\right) A_i \otimes A_i.$$

Then $\mathscr{A}'$ is also asymptotically stable. In particular, A_α and A are asymptotically stable.

Proof. Let $V \in \mathbb{N}^n$ be arbitrary and let Q be the unique, nonnegative-definite solution of (MLE4). Equivalently, Q satisfies

$$0 = A_\alpha Q + Q A_\alpha^T + \sum_{i=1}^{p} \left(\frac{\alpha_i'^2}{\alpha}\right) A_i Q A_i^T + V',$$

where

$$V' \triangleq \sum_{i=1}^{p} \alpha^{-1}(\alpha_i^2 - \alpha_i'^2) A_i Q A_i^T + V.$$

---

[1] The exponential product formula is essential to the proof here since (1) $A_\alpha \oplus A_\alpha$ cannot be expressed as a Kronecker product $S \otimes S$, and (2) $A_\alpha \oplus A_\alpha$ and $\sum_{i=1}^{p} \gamma_i A_i \otimes A_i$ do not generally commute.

Since $V' \in \mathbb{N}^n$, the stability of $\mathscr{A}'$ now follows as in the proof of the converse of Proposition 7.1. Finally, if V is chosen to be positive definite then $\sum_{i=1}^{p} (\alpha_i'^2/\alpha) A_i Q A_i^T + V'$ is also positive definite and it follows from Lemma 12.2 of [39] that A_α, and hence A, is asymptotically stable. □

Hence it follows from Proposition 7.2 that a necessary condition for $\mathscr{A}$ to be asymptotically stable is that

$$\alpha < 2 \max_{i=1,\cdots,n} \operatorname{Re} \lambda_i(A). \tag{7.10}$$

We now have the following monotonicity result.

PROPOSITION 7.3. *Let $\mathscr{U}_2' \subset \mathscr{U}_2$, where $\mathscr{U}_2'$ is defined as in* (5.7) *with α_i replaced by $\alpha_i' \in [0, \alpha_i]$, $i = 1, \cdots, p$. Furthermore, let $V \in \mathbb{P}^n$, assume $\mathscr{A}$ is asymptotically stable, and let $Q \in \mathbb{P}^n$ satisfy* (MLE4). *Then there exists $Q' \in \mathbb{P}^n$ satisfying*

$$0 = A_\alpha Q' + Q' A_\alpha^T + \sum_{i=1}^{p} \left(\frac{\alpha_i'^2}{\alpha} \right) A_i Q' A_i^T + V, \tag{7.11}$$

and, furthermore,

$$Q' \leqq Q. \tag{7.12}$$

Consequently,

$$\operatorname{tr} Q'R \leqq \operatorname{tr} QR, \tag{7.13}$$

$$\lambda_{\max}(Q'R) \leqq \lambda_{\max}(QR). \tag{7.14}$$

Proof. Subtracting (7.11) from (MLE4) yields

$$0 = A_\alpha (Q - Q') + (Q - Q') A_\alpha^T + \sum_{i=1}^{p} \left(\frac{\alpha_i'^2}{\alpha} \right) A_i (Q - Q') A_i^T + V',$$

where V' is defined in the proof of Proposition 7.2. Since, by the converse portion of Proposition 7.1, $\mathscr{A}'$ is asymptotically stable, $Q - Q' \geqq 0$, which yields (7.12) and thus (7.13) and (7.14). □

Returning now to the existence question, Proposition 7.1 shows that a solution to (MLE4) exists so long as $\alpha_1, \cdots, \alpha_p$ are sufficiently small such that $\mathscr{A}$ remains asymptotically stable for some $\alpha > 0$. To this end we can treat this as a stability perturbation problem and apply results from [3]. Within our modified Lyapunov equation approach we have the following related result. For this and the following result let $\|\cdot\|$ denote an arbitrary vector norm on $\mathbb{R}^{n^2}$ and the corresponding induced matrix norm.

PROPOSITION 7.4. *If*

$$\left\| (A \oplus A)^{-1} \left(\alpha I_{n^2} + \alpha^{-1} \sum_{i=1}^{p} \alpha_i^2 A_i \otimes A_i \right) \right\| < 1, \tag{7.15}$$

then for all $V \in \mathbb{N}^n$ there exists $Q \in \mathbb{N}^n$ satisfying (MLE4) *and hence $\mathscr{A}$ is asymptotically stable.*

Proof. Define $\{Q_k\}_{k=0}^{\infty}$ where Q_0 satisfies (3.14) and Q_{k+1} satisfies

$$0 = A Q_{k+1} + Q_{k+1} A^T + \Omega_7(Q_k) + V.$$

Note that $Q_k \geqq 0$, $k = 1, 2, \cdots$. Hence it follows that

$$\text{vec } Q_{k+1} - \text{vec } Q_k = -(A \oplus A)^{-1}[\text{vec } \Omega_7(Q_k) - \text{vec } \Omega_7(Q_{k-1})]$$

and thus

$$\|\text{vec } Q_{k+1} - \text{vec } Q_k\| \leqq \left\| (A \oplus A)^{-1}\left(\alpha I_{n^2} + \alpha^{-1} \sum_{i=1}^{p} \alpha_i^2 A_i \otimes A_i\right) \right\| \|\text{vec } Q_k - \text{vec } Q_{k-1}\|.$$

Using (7.15) it follows that $Q \triangleq \lim_{k \to \infty} Q_k$ exists. Thus $Q \geqq 0$ and satisfies (MLE4). Finally, by the converse of Proposition 7.1, $\mathscr{A}$ is asymptotically stable. □

Since (MLE5) is nonlinear, a slightly different approach is required for existence. For the following result let κ, $\beta > 0$ satisfy

$$\|e^{At}\| \leqq \kappa e^{-\beta t}, \qquad t \geqq 0, \tag{7.16}$$

where $\|\cdot\|$ denotes an arbitrary submultiplicative matrix norm that is monotonic on $\mathbb{N}^n$, and define $\rho \triangleq 2\beta/\kappa^2$.

PROPOSITION 7.5. *Suppose* $V \in \mathbb{N}^n$ *and*

$$4\alpha \|N\| \|\alpha^{-1} M + V\| < \rho^2. \tag{7.17}$$

Then there exists $Q \in \mathbb{N}^n$ *satisfying* (MLE5).

Proof. Consider the sequence $\{Q_k\}_{k=0}^{\infty}$ where Q_0 satisfies (3.14) and Q_{k+1} is given by

$$0 = AQ_{k+1} + Q_{k+1}A^T + \alpha Q_k N Q_k + \alpha^{-1} M + V.$$

Clearly, $Q_k \geqq 0$, $k = 0, 1, \cdots$. Next we have

$$Q_{k+1} = \int_0^{\infty} e^{At}[\alpha Q_k N Q_k + \alpha^{-1} M + V] e^{A^T t}\, dt, \tag{7.18}$$

which yields

$$\|Q_{k+1}\| \leqq \alpha \rho^{-1} \|N\| \|Q_k\|^2 + \rho^{-1} \|\alpha^{-1} M + V\|. \tag{7.19}$$

Similarly, from (3.14) we obtain

$$\|Q_0\| \leqq \rho^{-1} \|V\| \leqq \rho^{-1} \|\alpha^{-1} M + V\|.$$

Now suppose that

$$\|Q_k\| \leqq 2\rho^{-1} \|\alpha^{-1} M + V\|.$$

Then (7.17) and (7.19) imply

$$\begin{aligned} \|Q_{k+1}\| &\leqq \alpha \rho^{-1} \|N\| [2\rho^{-1} \|\alpha^{-1} M + V\|]^2 + \rho^{-1} \|\alpha^{-1} M + V\| \\ &< 2\rho^{-1} \|\alpha^{-1} M + V\|. \end{aligned}$$

Thus $\|Q_k\| \leqq 2\rho^{-1} \|\alpha^{-1} M + V\|$, $k = 0, 1, \cdots$. Next, (7.18) yields

$$\begin{aligned} Q_{k+1} - Q_k &= \alpha \int_0^{\infty} e^{At}[Q_k N Q_k - Q_{k-1} N Q_{k-1}] e^{A^T t}\, dt \\ &= \alpha \int_0^{\infty} e^{At}[Q_k N (Q_k - Q_{k-1}) + (Q_k - Q_{k-1}) N Q_{k-1}] e^{A^T t}\, dt \end{aligned}$$

and thus

$$\begin{aligned}\|Q_{k+1}-Q_k\| &\leqq \alpha\rho^{-1}\|N\|(\|Q_k\|+\|Q_{k-1}\|)\|Q_k-Q_{k-1}\| \\ &\leqq 4\alpha\rho^{-2}\|N\|\|\alpha^{-1}M+V\|\|Q_k-Q_{k-1}\| \\ &\leqq \varepsilon\|Q_k-Q_{k+1}\|,\end{aligned}$$

where $\varepsilon \triangleq 4\alpha\rho^{-2}\|N\|\|\alpha^{-1}M+V\|$. Since by (7.17) $\varepsilon < 1$, $\lim_{k\to\infty} Q_k$ exists, is nonnegative definite, and satisfies (MLE5). □

8. Additional upper bounds via recursive substitution. In this section we obtain additional upper bounds for $J_S(\mathscr{U})$ and $J_D(\mathscr{U})$ by utilizing a recursive substitution technique. The main idea involves rewriting (2.7) as

$$Q_{\Delta A} = -\mathrm{vec}^{-1}\{(A\oplus A)^{-1}(\Delta A\oplus\Delta A)\,\mathrm{vec}\,Q_{\Delta A}\}+Q_0 \tag{8.1}$$

and substituting this expression into the terms $\Delta AQ_{\Delta A} + Q_{\Delta A}\Delta A^T$ appearing in (2.7). This technique yields an equation that is, as expected, equivalent to (2.7) but that permits the development of additional bounds. As will be seen, the ability to develop new bounds exploits the fact that the substitution technique leads to terms that are quadratic in ΔA. We begin the development with the following technical result that does not require that A be asymptotically stable.

PROPOSITION 8.1. *Suppose $A\oplus A$ is invertible and let $\Delta A\in\mathbb{R}^{n\times n}$. If $Q_{\Delta A}$ satisfies* (2.7), *then $Q_{\Delta A}$ also satisfies*

$$\begin{aligned}0 = AQ_{\Delta A}+Q_{\Delta A}A^T-\mathrm{vec}^{-1}[&(\Delta A\oplus\Delta A)(A\oplus A)^{-1}(\Delta A\oplus\Delta A)\,\mathrm{vec}\,Q_{\Delta A} \\ &+(\Delta A\oplus\Delta A)(A\oplus A)^{-1}\,\mathrm{vec}\,V]+V.\end{aligned} \tag{8.2}$$

Conversely, if $Q_{\Delta A}$ satisfies (8.2) *and $(A-\Delta A)\oplus(A-\Delta A)$ is invertible, then $Q_{\Delta A}$ also satisfies* (2.7).

Proof. To obtain (8.2) substitute (8.1) into (2.7) as noted above. Conversely, adding the zero term $(\Delta A\oplus\Delta A)(A\oplus A)^{-1}(A\oplus A)\,\mathrm{vec}\,Q_{\Delta A}-(\Delta A\oplus\Delta A)\,\mathrm{vec}\,Q_{\Delta A}$ to (8.2), it follows that (8.2) can be written as

$$0=[(A-\Delta A)\oplus(A-\Delta A)](A\oplus A)^{-1}[(A+\Delta A)\oplus(A+\Delta A)\,\mathrm{vec}\,Q_{\Delta A}+\mathrm{vec}\,V],$$

which, under the invertibility assumption, implies that $Q_{\Delta A}$ satisfies (2.7). □

The following result is analogous to Theorem 3.1. We shall say that $\mathscr{U}$ is symmetric if $\Delta A\in\mathscr{U}$ implies $-\Delta A\in\mathscr{U}$.

THEOREM 8.1. *Suppose $\mathscr{U}$ is symmetric, let $\Omega_0\in\mathbb{N}^n$ satisfy*

$$\Delta AQ_0+Q_0\Delta A^T\leqq\Omega_0,\qquad \Delta A\in\mathscr{U}, \tag{8.3}$$

where Q_0 satisfies (3.14), *let $\hat{\Omega}:\mathbb{N}^n\to\mathbb{N}^n$ satisfy*

(8.4)

$$-\mathrm{vec}^{-1}[(\Delta A\oplus\Delta A)(A\oplus A)^{-1}(\Delta A\oplus\Delta A)\,\mathrm{vec}\,Q]\leqq\hat{\Omega}(Q),\qquad \Delta A\in\mathscr{U},\quad Q\in\mathbb{N}^n,$$

and suppose there exists $Q\in\mathbb{N}^n$ satisfying

$$0=AQ+QA^T+\hat{\Omega}(Q)+\Omega_0+V. \tag{8.5}$$

Then

$$(A+\Delta A,D)\ \text{is stabilizable},\qquad \Delta A\in\mathscr{U}, \tag{8.6}$$

if and only if

(8.7) $$A+\Delta A \text{ is asymptotically stable}, \qquad \Delta A \in \mathcal{U}.$$

In this case,

(8.8) $$Q_{\Delta A} \leqq Q, \qquad \Delta A \in \mathcal{U},$$

where $Q_{\Delta A}$ *satisfies* (2.7), *and*

(8.9) $$J_S(\mathcal{U}) \leqq \operatorname{tr} QR,$$

(8.10) $$J_D(\mathcal{U}) \leqq \lambda_{\max}(QR).$$

Proof. The equivalence of (8.6) and (8.7) follows from (8.5) as in the proof of Theorem 3.1. Next (8.8) follows by comparing (8.5) and (8.2) while using (8.3) and (8.4). Since $\mathcal{U}$ is assumed to be symmetric, it follows from (8.7) that $A - \Delta A$ is asymptotically stable, $\Delta A \in \mathcal{U}$, and hence $(A - \Delta A) \oplus (A - \Delta A)$ is invertible, $\Delta A \in \mathcal{U}$. Thus, the converse portion of Proposition 8.1 implies that $Q_{\Delta A}$ satisfying (8.2) also satisfies (2.7). Thus, the bound (8.8) can be used to obtain (8.9) and (8.10). □

The principal difference between (8.4) and (3.1) is that ΔA appears linearly in (3.1), whereas it appears quadratically in (8.4). By exploiting this structure we can obtain new bounds for $Q_{\Delta A}$. To simplify matters, we now consider the bound in (8.4) in two special cases. In the first case we set $\mathcal{U} = \mathcal{U}_1$ and $p = 1$ so that $\Delta A = \sigma_1 A_1$, $|\sigma_1| \leqq \delta_1$. In this case (8.4) becomes

(8.11)

$$-\sigma_1^2 \operatorname{vec}^{-1}[(A_1 \oplus A_1)(A \oplus A)^{-1}(A_1 \oplus A_1) \operatorname{vec} Q] \leqq \hat{\Omega}(Q), \qquad |\sigma_1| \leqq \delta_1, \quad Q \in \mathbb{N}^n.$$

One choice of $\hat{\Omega}(\cdot)$ that immediately suggests itself can be obtained by defining the matrix function $|\cdot|_+$ on the set of symmetric matrices by

(8.12) $$|S|_+ \triangleq \tfrac{1}{2}(S + |S|),$$

which effectively replaces the negative eigenvalues of S by zeros. We shall thus utilize the fact that

(8.13) $$\sigma_1^2 S \leqq \delta_1^2 |S|_+, \qquad |\sigma_1| \leqq \delta_1,$$

for all symmetric S.

COROLLARY 8.1. *Let* $V \in \mathbb{P}^n$, $\mathcal{U} = \mathcal{U}_1$, $p = 1$, *let* $\Omega_0 \in \mathbb{N}^n$ *satisfy* (8.3), *and suppose there exists* $Q \in \mathbb{N}^n$ *satisfying*

(8.14) $$0 = AQ + QA^T + \delta_1^2 |-\operatorname{vec}^{-1}[(A_1 \oplus A_1)(A \oplus A)^{-1}(A_1 \oplus A_1) \operatorname{vec} Q]|_+ + \Omega_0 + V.$$

Then (8.7)–(8.10) *are satisfied.*

For the next specialization we shall assume that

(8.15) $$(\Delta A)A = A(\Delta A), \qquad \Delta A \in \mathcal{U},$$

which holds, for example, for modal systems with frequency uncertainty (see § 10). It thus follows that $(A \oplus A)^{-1}(\Delta A \oplus \Delta A) = (\Delta A \oplus \Delta A)(A \oplus A)^{-1}$ and thus (8.4) can be rewritten as

(8.16) $$\Delta A^2 \hat{Q} + 2\Delta A \hat{Q} \Delta A^T + \hat{Q} \Delta A^{2T} \leqq \hat{\Omega}(Q), \qquad \Delta A \in \mathcal{U}, \quad Q \in \mathbb{N}^n,$$

where $\hat{Q} \in \mathbb{N}^n$ satisfies

(8.17) $$0 = A\hat{Q} + \hat{Q}A^T + Q.$$

Assuming in addition to (8.15) that $\Delta A = \sigma_1 A_1$, $|\sigma_1| \leqq \delta_1$, (8.14) becomes

$$0 = AQ + QA^T + \delta_1^2 |\, A_1^2 \hat{Q} + 2A_1 \hat{Q} A_1^T + \hat{Q} A_1^{2T} |_+ + \Omega_0 + V. \tag{8.18}$$

Remark 8.1. It is interesting to note that the left-hand side of (8.16) is of the same form as $\Omega_4(\cdot)$. Specifically, the term $\Delta A^2 \hat{Q} + \hat{Q} \Delta A^{2T}$ is analogous to $A_i^2 Q + Q A_i^{2T}$ whereas $2\Delta A \hat{Q} \Delta A^T$ is similar to $A_i Q A_i^T$.

9. An alternative approach yielding upper and lower bounds. In this section we develop a variation on the results of § 3 that has the additional benefit of yielding both upper and lower performance bounds. The basic approach was suggested by results obtained in [44]. To simplify the presentation we assume as in the preceding section that $\mathscr{U}$ is symmetric. This symmetry assumption of course holds for all of the uncertainty sets considered in previous sections. The underlying idea involves bounding the deviation of $Q_{\Delta A}$ from Q_0 rather than bounding $Q_{\Delta A}$ directly.

THEOREM 9.1. *Let* $\Omega_0 \in \mathbb{N}^n$ *satisfy*

$$\Delta A Q_0 + Q_0 \Delta A^T \leqq \Omega_0, \qquad \Delta A \in \mathscr{U}, \tag{9.1}$$

let $\Omega : \mathbb{N}^n \to \mathbb{N}^n$ *be such that* (3.1) *is satisfied, and suppose there exists* $\Delta \mathscr{Q} \in \mathbb{N}^n$ *satisfying*

$$0 = A \Delta \mathscr{Q} + \Delta \mathscr{Q} A^T + \Omega(\Delta \mathscr{Q}) + \Omega_0. \tag{9.2}$$

Then

$$(A + \Delta A, \Omega_0^{1/2}) \text{ is stabilizable}, \qquad \Delta A \in \mathscr{U}, \tag{9.3}$$

if and only if

$$A + \Delta A \text{ is asymptotically stable}, \qquad \Delta A \in \mathscr{U}. \tag{9.4}$$

In this case,

$$Q_0 - \Delta \mathscr{Q} \leqq Q_{\Delta A} \leqq Q_0 + \Delta \mathscr{Q}, \qquad \Delta A \in \mathscr{U}, \tag{9.5}$$

where $Q_{\Delta A}$ *is given by* (2.7), *and*

$$\operatorname{tr}(Q_0 + \Delta \mathscr{Q}) R \leqq J_S(\mathscr{U}) \leqq \operatorname{tr}(Q_0 + \Delta \mathscr{Q}) R, \tag{9.6}$$

$$\lambda_{\max}[(Q_0 - \Delta \mathscr{Q}) R] \leqq J_D(\mathscr{U}) \leqq \lambda_{\max}[(Q_0 + \Delta \mathscr{Q}) R]. \tag{9.7}$$

Proof. Define

$$\Delta Q \triangleq Q_{\Delta A} - Q_0 \tag{9.8}$$

and subtract (3.14) from (2.7) to obtain

$$0 = (A + \Delta A) \Delta Q + \Delta Q (A + \Delta A)^T + \Delta A Q_0 + Q_0 \Delta A^T. \tag{9.9}$$

Now rewrite (9.2) as

$$0 = (A + \Delta A) \Delta \mathscr{Q} + \Delta \mathscr{Q} (A + \Delta A)^T + \Omega(\Delta \mathscr{Q}) - (\Delta A \Delta \mathscr{Q} + \Delta \mathscr{Q} \Delta A^T) + \Omega_0. \tag{9.10}$$

Using (9.10), the equivalence of (9.3) and (9.4) is immediate as in the proof of Theorem 3.1. Next, subtracting (9.9) from (9.10) yields

$$\begin{aligned} 0 = {} & (A + \Delta A)(\Delta \mathscr{Q} - \Delta Q) + (\Delta \mathscr{Q} - \Delta Q)(A + \Delta A)^T + \Omega(\Delta \mathscr{Q}) \\ & - (\Delta A \Delta \mathscr{Q} + \Delta \mathscr{Q} \Delta A^T) + \Omega_0 - (\Delta A Q_0 + Q_0 \Delta A^T). \end{aligned} \tag{9.11}$$

Using (3.1) and (9.1) it follows from (9.11) that

$$\Delta \mathscr{Q} - \Delta Q \geqq 0,$$

or, equivalently,

$$Q_{\Delta A} \leqq Q_0 + \Delta\mathcal{Q}. \tag{9.12}$$

To obtain the lower bound rewrite (9.9) as

$$0 = (A+\Delta A)(-\Delta Q) + (-\Delta Q)(A+\Delta A)^T - (\Delta A Q_0 + Q_0 \Delta A^T). \tag{9.13}$$

Also, note that because of the assumed symmetry of $\mathcal{U}$, (9.1) holds with ΔA appearing in the inequality replaced by $-\Delta A$. Hence it can be shown similarly that

$$\Delta\mathcal{Q} + \Delta Q \geqq 0,$$

or, equivalently,

$$Q_0 - \Delta\mathcal{Q} \leqq Q_{\Delta A}. \tag{9.14}$$

Finally, (9.6) and (9.7) follow from (9.5). □

Remark 9.1. To compare the upper bound in (9.5) with (3.5), rewrite (9.2) as

$$0 = A(Q_0 + \Delta\mathcal{Q}) + (Q_0 + \Delta\mathcal{Q})A^T + \Omega(\Delta\mathcal{Q}) + \Omega_0 + V. \tag{9.15}$$

If $\Omega(\Delta\mathcal{Q}) + \Omega_0 = \Omega(Q_0 + \Delta\mathcal{Q})$ then (9.15) has the same form as (3.2) and thus the two upper bounds are identical. This will be the case, for example, if $\Omega(\cdot) = \Omega_7(\cdot)$ and Ω_0 is chosen to be $\Omega_7(Q_0)$ since $\Omega_7(\cdot)$ is linear. If, however, $\Omega(\Delta\mathcal{Q}) + \Omega_0 < \Omega(Q_0 + \Delta\mathcal{Q})$ then the upper bound in (9.5) will be sharper. In any case it is clear that the individual treatment of $\Delta\mathcal{Q}$ and Q_0 yields potentially new upper bounds.

Remark 9.2. Theorem 9.1 does not guarantee that the lower bound $Q_0 - \Delta\mathcal{Q}$ for $Q_{\Delta A}$ is nonnegative definite. However, $Q_{\Delta A}$ is always nonnegative definite and thus the lower bound in (9.5) may be of limited usefulness. Nevertheless, if $Q_0 - \Delta\mathcal{Q}$ is indefinite then, depending on R, the lower bounds in (9.6) and (9.7) may still be positive and thus be meaningful lower bounds.

10. Analytical examples. In this section we consider simple analytical examples that illustrate the principal results of the paper. These examples also provide insight into the individual characteristics of different bounds as a prelude to numerical examples considered in the following section.

To begin we consider the simplest possible example. Set $n = 1$, $A < 0$, $R > 0$, $V > 0$, $A_1 = 1$, and $\mathcal{U} = \{\Delta A : |\Delta A| \leqq \delta_1\}$. For $\delta_1 < -A$, $Q_{\Delta A} = V/2(|A| - \Delta A)$ and $J_S(\mathcal{U}) = J_D(\mathcal{U}) = RV/2(|A| - \delta_1)$, where this worst-case performance is achieved for $\Delta A = \delta_1$. Solving (MLE1) yields $Q = V/2(|A| - \delta_1)$, which is a nonconservative result for both robust stability and performance. The same result is obtained from (MLE4) by setting $\alpha = \alpha_1 = \delta_1$. To apply (MLE5), set $\delta_1 = \sqrt{MN}$. Choosing $\alpha = 2\delta_1(|A| - \delta_1)NV$ again yields the nonconservative result. Finally, the same result follows from Theorem 8.1.

For the second example we consider nondestabilizing uncertainty in the imaginary component of an uncertain eigenvalue, i.e., frequency uncertainty, in contrast to uncertainty in the real part considered in the previous example. Let $n = 2$,

$$A = \begin{bmatrix} -\nu & \omega \\ -\omega & -\nu \end{bmatrix}, \qquad \nu > 0, \quad \omega \geqq 0,$$

$V = R = I_2$, and $\mathcal{U} = \{\Delta A : \Delta A = \sigma_1 A_1, |\sigma_1| \leqq \delta_1\}$, where

$$A_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Obviously, $A + \Delta A$ remains asymptotically stable for all values of σ_1 since ΔA affects only the imaginary part of the poles of ΔA. The question then is whether the robustness tests are able to guarantee this robustness. Note also that because of the choice of V, $Q_{\Delta A} = Q_0 = (2\nu)^{-1} I_2$ for all $\Delta A \in \mathscr{U}$. For this example we note that (MLE1) is satisfied by $Q = (2\nu)^{-1} I_2$, which is independent of δ_1. Thus (MLE1) possesses a nonnegative-definite solution for all $\delta_1 > 0$, which shows that (MLE1) is nonconservative with respect to robust stability and performance. Since $A(\Delta A) = (\Delta A)A$, it can also be seen that the same result holds for (8.18). The situation is considerably different for (MLE4) and (MLE5). To analyze (MLE4) note that $\mathscr{A}$ has an eigenvalue $-2\nu + \alpha + \delta_1$. (This can be shown by diagonalizing A and A_1 and thus $\mathscr{A}$.) Since, by Proposition 7.1, $\mathscr{A}$ must be asymptotically stable, we require $\delta_1 < 2\nu$. This is, of course, an extremely conservative result, especially when the damping ν is small. For (MLE5) we can factor $A_1 = D_1 E_1$. Thus, let $D_1 = I_2$ and $E_1 = A_1$ and define $M = \delta_1^2 I_2$ and $N = I_2$. Assuming that Q is a multiple of I_2, it follows that Q is nonnegative definite only if $\delta_1 \leqq \nu$, which is again an extremely conservative result. The reason for this conservatism becomes clear by noting that M and N as given above will also serve as bounds for perturbations of the form $\sigma_1 I_2$ for which the range of nondestabilizing σ_1 is $|\sigma_1| < \delta_1$. This will also be the case for all factorizations $D_1 E_1$ of A_1 since $D_1 D_1^T$ and $E_1^T E_1$ must be positive definite and thus will also serve as bounds for destabilizing perturbations such as $\sigma_1 I_2$.

Finally, we consider a nondestabilizing uncertainty affecting the interaction of a pair of real poles. Let $n = 2$, $A = -I_2$, $V = R = I_2$, and $\mathscr{U} = \{\Delta A : \Delta A = \sigma_1 A_1, |\sigma_1| \leqq \delta_1\}$, where

$$A_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Obviously, $A + \Delta A$ remains asymptotically stable for all values of σ_1 since ΔA does not affect the nominal poles. Note that

$$Q_{\Delta A} = \begin{bmatrix} \sigma_1^2/4 + \frac{1}{2} & \sigma_1/4 \\ \sigma_1/4 & \frac{1}{2} \end{bmatrix}$$

and $J_S(\mathscr{U}) = \frac{1}{4}\delta_1^2 + 1$, where this worst-case performance is achieved for $\sigma_1 = \delta_1$. In this case (MLE1) has the solution $Q = (2 - \delta_1)^{-1} I_2$, which is valid only for $\delta_1 < 2$, an extremely conservative robust stability result. Furthermore, the corresponding performance bound $\operatorname{tr} QR = 2(2 - \delta_1)^{-1}$ is conservative with respect to the actual worst-case performance $\frac{1}{4}\delta_1^2 + 1$. In contrast, (MLE4) has the solution

$$Q = \begin{bmatrix} (2-\alpha\delta_1)^{-1} + \alpha^{-1}\delta_1(2-\alpha\delta_1)^{-2} & 0 \\ 0 & (2-\alpha\delta_1)^{-1} \end{bmatrix},$$

which is nonnegative definite for all δ_1 so long as $\alpha < 2/\delta_1$. Hence (MLE4) is nonconservative with respect to robust stability. For robust performance,

$$\operatorname{tr} QR = 2(2-\alpha\delta_1)^{-1} + \alpha^{-1}\delta_1(2-\alpha\delta_1)^{-2},$$

which can be shown to be an upper bound for $\frac{1}{4}\delta_1^2 + 1$. Choosing, for example, $\alpha = \delta_1^{-1}$ yields $\operatorname{tr} QR = \delta_1^2 + 2$. The parameter α can also be chosen to minimize $\operatorname{tr} QR$, although this is somewhat tedious to carry out analytically. Finally, (MLE5) has the solution

$$Q = \begin{bmatrix} \frac{1}{2}(1 + \alpha^{-1}\delta_1) & 0 \\ 0 & [1 - (1 - \alpha\delta_1)^{1/2}]/\alpha\delta_1 \end{bmatrix},$$

which exists so long as $\alpha \leqq 1/\delta_1$. Hence (MLE5) is also nonconservative with respect to robust stability. Choosing $\alpha = 1/\delta_1$ yields tr $QR = \frac{1}{2}\delta_1^2 + \frac{3}{2}$, which lies above the nonconservative bound $\frac{1}{4}\delta_1^2 + 1$. Again, α can be chosen to minimize tr QR.

11. Numerical examples. In this section we consider additional examples illustrating the results developed in earlier sections. In contrast to the analytical examples considered in § 10, however, we consider more complex examples by numerically solving the modified Lyapunov equations. Here we focus on (MLE4) and (MLE5), which are the easiest to solve numerically. Specifically, we solved (MLE4) by using the representation (7.2) (although this may not be practical when n is large), and we solved (MLE5) by means of a standard Riccati package. To simplify matters we consider only uncertainties ΔA of the form $\sigma_1 A_1$. Evaluation and presentation of robust stability and performance results for multiparameter uncertainty can be fairly complex and thus are deferred to a future numerical study.

Since both robustness tests (MLE4) and (MLE5) depend on an arbitrary positive constant α, it is desirable to determine the value of α that yields the tightest (i.e., lowest) performance bound for each robust stability range. To this end we performed a simple one-dimensional search to determine the best such α. Although analytical techniques may assist in determining optimal values of α more efficiently, the search technique proved to be adequate for the examples considered here.

As a first example we consider the control system given in [1] to demonstrate the lack of a guaranteed gain margin for LQG controllers. Hence consider

$$\dot{x}_0(t) = A_0 x_0(t) + B_0 u(t) + w_1(t), \tag{11.1}$$

$$y(t) = C_0 x_0(t) + w_2(t), \tag{11.2}$$

with controller

$$\dot{x}_c(t) = A_c x_c(t) + B_c y(t), \tag{11.3}$$

$$u(t) = C_c x_c(t), \tag{11.4}$$

and performance

$$J = \lim_{t \to \infty} \mathbb{E}[x_0^T(t) R_1 x_0(t) + u^T(t) R_2 u(t)]. \tag{11.5}$$

The data are

$$A_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_0 = [1 \quad 0],$$

$$V_1 = R_1 = \rho \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad V_2 = R_2 = 1,$$

where V_1 and V_2 are the intensities of $w_1(t)$ and $w_2(t)$, respectively. Uncertainty ΔB_0 in B_0 is thus represented by $\sigma_1 B_1$, where $B_1 = [0\ 1]^T$. Thus the closed-loop system corresponds to

$$A = \begin{bmatrix} A_0 & B_0 C_c \\ B_c C_0 & A_c \end{bmatrix}, \qquad A_1 = \begin{bmatrix} 0 & B_1 C_c \\ 0 & 0 \end{bmatrix},$$

$$R = \begin{bmatrix} R_1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad V = \begin{bmatrix} V_1 & 0 \\ 0 & B_c V_2 B_c^T \end{bmatrix},$$

where the zero in the (2, 2) block of R denotes the fact that we are considering the robust performance bound for the state regulation cost only. Choosing $\rho = 60$, it follows that the LQG gains are given by

$$A_c = \begin{bmatrix} -9 & 1 \\ -20 & -9 \end{bmatrix}, \quad B_c = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \quad C_c = [-10 \quad -10].$$

For this controller the actual stability region corresponds to $\sigma_1 \in (-.07, .01)$ so that the largest symmetric region about $\sigma_1 = 0$ is $|\sigma_1| < .01$. The worst-case performance over each stability region $|\sigma_1| < \delta_1$ is denoted by the solid line in Fig. 1, whereas the performance bounds obtained from (MLE4) and (MLE5) are shown for several values of δ_1. For (MLE5) we set $D_1 = [0\ 1\ 0\ 0]^T$ and $E_1 = [0\ 0\ C_c]$. Note that (MLE5) yields considerably tighter estimates of worst-case performance, particularly as δ_1 approaches .01. For (MLE4) optimal values of α were in the range .0012 to .0058, whereas for (MLE5) (with $\Omega_{10'}(\cdot)$, see (5.26)) α was in the range .0143 to .0020.

As a second example we consider a pair of nominally uncoupled oscillators with uncertain coupling. This example was considered in [45] using the majorant Lyapunov technique. Let

$$A = \begin{bmatrix} -\nu & \omega_1 & 0 & 0 \\ -\omega_1 & -\nu & 0 & 0 \\ 0 & 0 & -\nu & \omega_2 \\ 0 & 0 & -\omega_2 & -\nu \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$\nu = .2, \quad \omega_1 = .2, \quad \omega_2 = 1.8, \quad R = V = I_4,$$

and, for (MLE5), define $D_1 = A_1$ and $E_1 = I_4$. We consider bounds on $J_S(\mathscr{U})$ only.

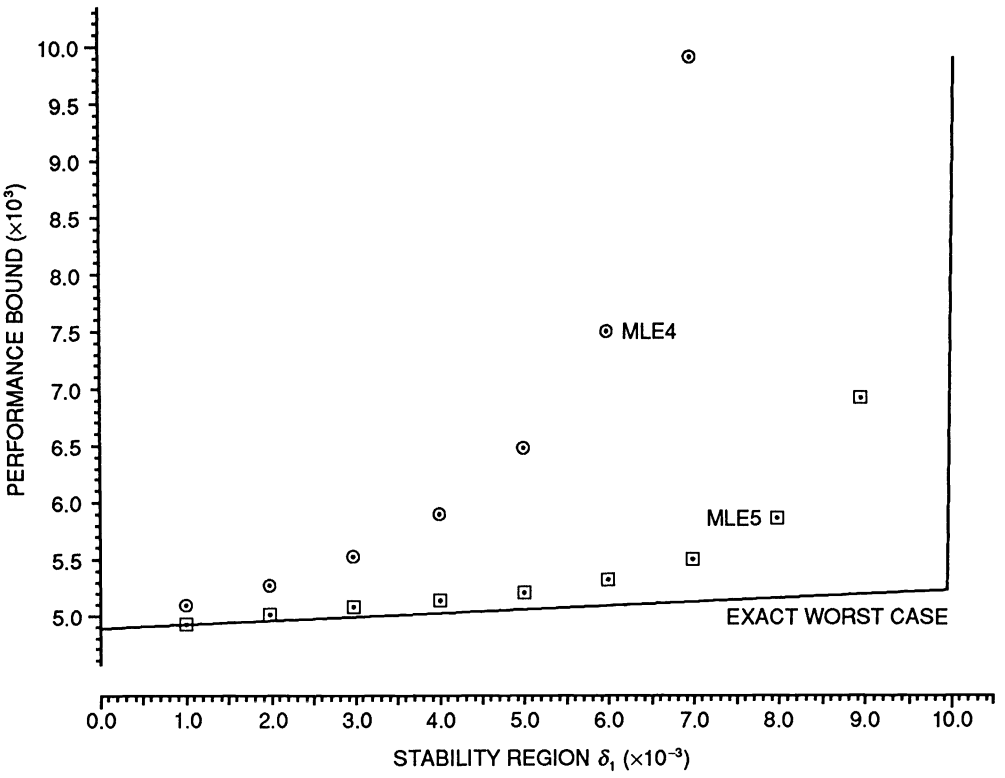


FIG. 1

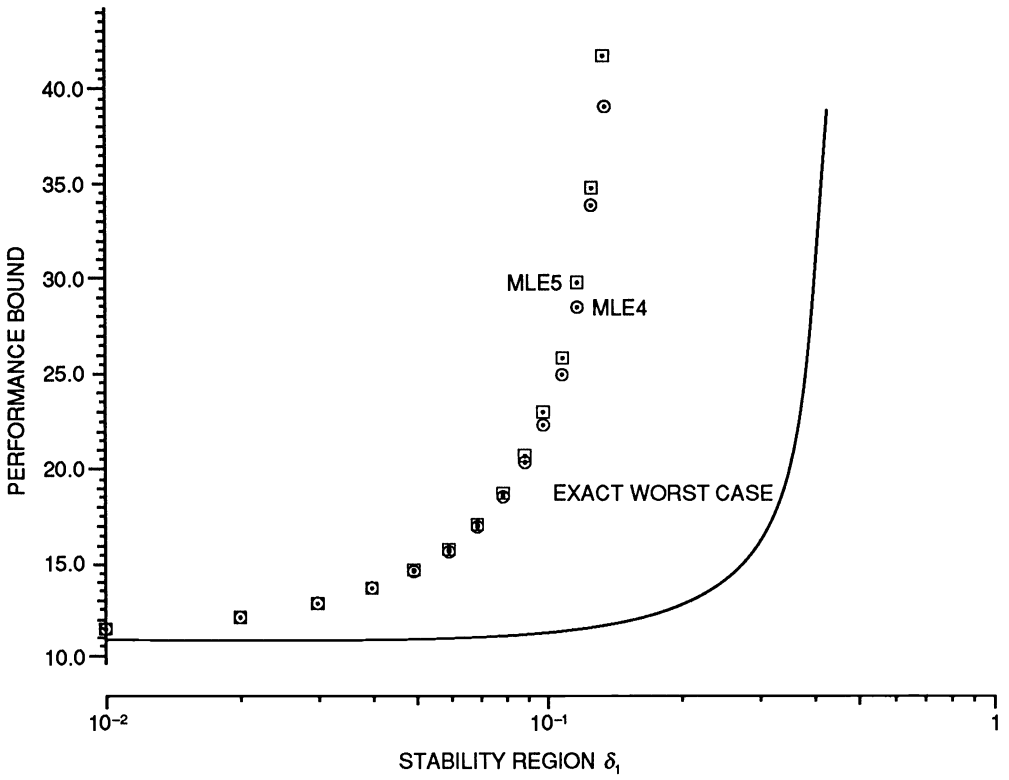


FIG. 2

Figure 2 illustrates the exact worst-case performance along with performance bounds obtained from (MLE4) and (MLE5). For (MLE4) optimal values of α ranged from .036 to .141, whereas for (MLE5) optimal α was between .361 and .096. Although (MLE4) was slightly less conservative than (MLE5), both bounds were able to guarantee robust stability only for $\delta_1 = .15$, whereas the largest stability region is actually $\delta_1 = .54$. It is interesting to contrast this result with [45] where the majorant Lyapunov technique yielded a robust stability range of $\delta_1 = .4$ for a richer class of off-diagonal blocks having maximum singular value less than δ_1.

12. Conclusion. A variety of quadratic Lyapunov bounds have been developed for both robust stability and performance. It seems clear, however, that no single quadratic Lyapunov bound is superior to the others. Although the conservatism of each bound is problem dependent, it is desirable to better understand the nature of the conservatism in order to utilize the bounds in an effective manner. In addition, the issue of *necessity* remains to be addressed. That is, if a system is robustly quadratically stable (i.e., robustly stable with a corresponding Lyapunov function), then is such a Lyapunov function necessarily given by one of the modified Lyapunov equations given in this paper? Furthermore, a better understanding is needed of the gap between robust stability and robust quadratic stability.

Acknowledgment. We thank A. W. Daubendiek for producing the numerical results in § 11.

Note added in proof. (1) The assumption $x(0) = 0$ in (2.2) is stronger than necessary for the treatment of (2.4). If $x(0) \neq 0$, then Lemma 2.1 remains unchanged since the

effect of $x(0)$ vanishes as $t \to \infty$. If, however, $x(0) = 0$, then $Q_{\Delta A}(t)$ is increasing on $[0, \infty)$ and (2.4) is equivalent to

$$J_S(\mathscr{U}) = \sup_{\Delta A \in \mathscr{U}} \sup_{t \in [0,\infty)} \mathbb{E}\{\| y(t) \|_2^2\} \leqq \beta_S.$$

For $J_D(\mathscr{U})$, $x(0) = 0$ is essential since $\| y(\cdot) \|_{\infty,2}$ involves the supremum over $[0, \infty)$. If $x(0) \neq 0$, then the analysis can possibly be redone by considering the supremum over $[t, \infty)$ and letting $t \to \infty$ to eliminate the effect of the initial condition.

(2) A relationship between the linear bound $\Omega_7(\cdot)$ and the quadratic bound $\Omega_{10}(\cdot)$ can be seen as follows. If $\Delta A = \sigma_1 A_1$, $|\sigma_1| \leqq \delta_1$, then factor $\Delta A = A_L A_R$ as in $\mathscr{U}_3$ according to $A_L = \sigma_1 A_1 Q^{1/2}$ and $A_R = Q^{-1/2}$ with bounds $M = \delta_1^2 A_1 Q A_1^T$ and $N = Q^{-1}$. The unusual feature here is that the "splitting" of ΔA is Q-dependent. Then, by (5.22),

$$\Omega_{10}(Q) = \alpha^{-1} \delta_1^2 A_1 Q A_1^T + \alpha Q,$$

which has the form of $\Omega_5(Q)$.

REFERENCES

[1] J. C. DOYLE, *Guaranteed margins for* LQG *regulators*, IEEE Trans. Automat. Control, 23 (1978), pp. 756–757.

[2] E. SOROKA AND U. SHAKED, *On the robustness of* LQ *regulators*, IEEE Trans. Automat. Control, 29 (1984), pp. 664–665.

[3] C. VAN LOAN, *How near is a stable matrix to an unstable matrix?*, Contemporary Mathematics, Vol. 47, American Mathematical Society, Providence, RI, 1985, pp. 465–478.

[4] D. HINRICHSEN AND A. J. PRITCHARD, *Stability radii of linear systems*, Systems Control Lett., 7 (1986), pp. 1–10.

[5] J. M. MARTIN AND G. A. HEWER, *Smallest destabilizing perturbations for linear systems*, Internat. J. Control, 45 (1987), pp. 1495–1504.

[6] B. A. FRANCIS, *A Course in H_∞ Control Theory*, Springer-Verlag, New York, 1987.

[7] C. KENNEY AND A. J. LAUB, *Controllability and stability radii for companion form systems*, Math. Contr. Sig. Sys., 1 (1988), pp. 239–256.

[8] G. S. MICHAEL AND C. W. MERRIAM, *Stability of parametrically disturbed linear optimal control systems*, J. Math. Anal. Appl., 28 (1969), pp. 294–302.

[9] S. S. L. CHANG AND T. K. C. PENG, *Adaptive guaranteed cost control of systems with uncertain parameters*, IEEE Trans. Automat. Control, 17 (1972), pp. 474–483.

[10] R. V. PATEL, M. TODA, AND B. SRIDHAR, *Robustness of linear quadratic state feedback designs in the presence of system uncertainty*, IEEE Trans. Automat. Control, 22 (1977), pp. 945–949.

[11] G. LEITMANN, *Guaranteed asymptotic stability for a class of uncertain linear dynamical systems*, J. Optim. Theory Appl., 27 (1979), pp. 99–106.

[12] A. VINKLER AND L. J. WOOD, *Multistep guaranteed cost control of linear systems with uncertain parameters*, J. Guid. Control, 2 (1979), pp. 449–456.

[13] M. ESLAMI AND D. L. RUSSELL, *On stability with large parameter variations stemming from the direct method of Lyapunov*, IEEE Trans. Automat. Control, 25 (1980), pp. 1231–1234.

[14] R. V. PATEL AND M. TODA, *Quantitative measures of robustness for multivariable systems*, in Proc. Joint Automat. Control Conference paper TP8-A, San Francisco, CA, June 1980.

[15] M. CORLESS AND G. LEITMANN, *Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamical systems*, IEEE Trans. Automat. Control, 26 (1981), pp. 1139–1144.

[16] J. S. THORP AND B. R. BARMISH, *On guaranteed stability of uncertain linear systems via linear control*, J. Optim. Theory Appl., 35 (1981), pp. 559–579.

[17] E. NOLDUS, *Design of robust state feedback laws*, Internat. J. Control, 35 (1982), pp. 935–944.

[18] B. R. BARMISH, M. CORLESS, AND G. LEITMANN, *A new class of stabilizing controllers for uncertain dynamical systems*, SIAM J. Control Optim., 21 (1983), pp. 246–255.

[19] B. R. BARMISH, I. R. PETERSEN, AND A. FEUER, *Linear ultimate boundedness control of uncertain dynamic systems*, Automatica, 19 (1983), pp. 523–532.

[20] B. R. BARMISH, *Necessary and sufficient conditions for quadratic stabilizability of an uncertain linear system*, J. Optim. Theory Appl., 46 (1985), pp. 399–408.

[21] R. K. YEDAVALLI, S. S. BANDA, AND D. B. RIDGELY, *Time-domain stability robustness measures for linear regulators*, J. Guid. Control Dyn., 8 (1985), pp. 520–524.

[22] R. K. YEDAVALLI, *Improved measures of stability robustness for linear state space models*, IEEE Trans. Automat. Control, 30 (1985), pp. 577–579.

[23] D. S. BERNSTEIN AND D. C. HYLAND, *The optimal projection/maximum entropy approach to designing low-order, robust controllers for flexible structures*, in Proc. 24th IEEE Conference on Decision and Control, Fort Lauderdale, FL, December 1985, pp. 745–752.

[24] A. R. GALIMIDI AND B. R. BARMISH, *The constrained Lyapunov problem and its application to robust output feedback stabilization*, IEEE Trans. Automat. Control, 31 (1986), pp. 410–419.

[25] I. R. PETERSEN AND C. V. HOLLOT, *A Riccati equation approach to the stabilization of uncertain systems*, Automatica, 22 (1986), pp. 397–411.

[26] D. S. BERNSTEIN AND S. W. GREELEY, *Robust output-feedback stabilization: Deterministic and stochastic perspectives*, in Proc. Amer. Control Conference, Seattle, WA, June 1986, pp. 1818–1826.

[27] ———, *Robust controller synthesis using the maximum entropy design equations*, IEEE Trans. Automat. Control, 31 (1986), pp. 362–364.

[28] I. R. PETERSEN, *A stabilization algorithm for a class of uncertain linear systems*, Systems Control Lett., 8 (1987), pp. 351–357.

[29] D. HINRICHSEN AND A. J. PRITCHARD, *Stability radii and structured perturbations and the algebraic Riccati equation*, Systems Control Lett., 8 (1987), pp. 105–113.

[30] D. S. BERNSTEIN AND W. M. HADDAD, *The optimal projection equations with Petersen–Hollot bounds: Robust controller synthesis with guaranteed structured stability radius*, in Proc. IEEE Conference Decision and Control, Los Angeles, CA, December 1987, pp. 1308–1318.

[31] K. ZHOU AND P. P. KHARGONEKAR, *Stability robustness bounds for linear state-space models with structured uncertainty*, IEEE Trans. Automat. Control, 32 (1987), pp. 621–623.

[32] S. P. BHATTACHARYYA, *Robust Stabilization Against Structured Perturbations*, Springer-Verlag, New York, 1987.

[33] D. S. BERNSTEIN, *Robust static and dynamic output-feedback stabilization: Deterministic and stochastic perspectives*, IEEE Trans. Automat. Control, 32 (1987), pp. 1076–1084.

[34] O. I. KOSMIDOU AND P. BERTRAND, *Robust-controller design for systems with large parameter variations*, Internat. J. Control, 45 (1987), pp. 927–938.

[35] D. S. BERNSTEIN AND W. M. HADDAD, *Robust stability and performance for fixed-order dynamic compensation via the optimal projection equations with guaranteed cost bounds*, in Proc. Amer. Control Conference, Atlanta, GA, June 1988, pp. 2471–2476, Math. Control Sig. Sys., 3 (1990), to appear.

[36] ———, *The optimal projection equations with Petersen–Hollot bounds: Robust stability and performance via fixed-order dynamic compensation for systems with structured real-valued parameter uncertainty*, IEEE Trans. Automat. Control, 33 (1988), pp. 578–582.

[37] ———, *Robust stability and performance analysis for linear dynamic systems*, IEEE Trans. Automat. Control, 34 (1989), pp. 751–758.

[38] D. A. WILSON, *Convolution and Hankel operator norms for linear systems*, IEEE Trans. Automat. Control, 34 (1988), pp. 94–97.

[39] W. M. WONHAM, *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York, 1979.

[40] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, London, 1985.

[41] T. KATO, *Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin, New York, 1980.

[42] J. W. BREWER, *Kronecker products and matrix calculus in system theory*, IEEE Trans. Circuits and Systems, 25 (1978), pp. 772–781.

[43] L. ARNOLD, *Stochastic Differential Equations: Theory and Applications*, John Wiley, New York, 1974.

[44] E. G. COLLINS, JR. AND D. C. HYLAND, *Improved robust performance bounds in covariance majorant analysis*, Internat. J. Control, 50 (1989), pp. 495–509.

[45] D. C. HYLAND AND D. S. BERNSTEIN, *The majorant Lyapunov equation: A nonnegative matrix equation for guaranteed robust stability and performance of large scale systems*, IEEE Trans. Automat. Control, 32 (1987), pp. 1005–1013.

[46] R. E. SKELTON, *Dynamic Systems Control*, John Wiley, New York, 1988.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 272–277, April 1990

008

ON THE SINGULAR VALUES OF A PRODUCT OF OPERATORS*

RAJENDRA BHATIA† AND FUAD KITTANEH‡

Abstract. For compact Hilbert space operators A and B, the singular values of A^*B are shown to be dominated by those of $\frac{1}{2}(AA^* + BB^*)$.

Key words. compact operator, singular values, unitarily invariant norm

AMS(MOS) subject classifications. 15A42, 15A60, 47A30, 47B05, 47B10

1. Introduction. Let A be a compact operator on a separable Hilbert space. The *singular values* of A, i.e., the eigenvalues of the operator $(A^*A)^{1/2}$, enumerated in decreasing order, will be denoted by $s_j(A), j = 1, 2, \cdots$.

There is a considerable body of literature dealing with inequalities for the singular values of products and sums of operators. See, e.g., [3], [4], [7], [8], [11], and references therein.

The main result of this note is of this *genre*:

THEOREM 1. *Let A, B be compact operators. Then for $j = 1, 2, \cdots$, we have*

$$2s_j(A^*B) \leqq s_j(AA^* + BB^*). \tag{1}$$

Let $|||\,.\,|||$ denote a *unitarily invariant norm*. (Such a norm is defined on all operators when the space is finite dimensional, and on a *norm ideal* associated with $|||\,.\,|||$ when the space is infinite dimensional [3], [9]. We will not make repeated mention of this ideal for the sake of brevity.) A consequence of this theorem is given in the following corollary.

COROLLARY 2. *Let A, B be compact operators. Then for every unitarily invariant norm we have*

$$2|||A^*B||| \leqq |||AA^* + BB^*|||. \tag{2}$$

In the special case when A and B are Hermitian, (1) reduces to

$$2s_j(AB) \leqq s_j(A^2 + B^2). \tag{3}$$

When the space is one-dimensional, this reduces to the familiar arithmetic mean-geometric mean inequality for real numbers:

$$2|ab| \leqq a^2 + b^2. \tag{4}$$

Thus (3) may be regarded as a "noncommutative arithmetic mean-geometric mean inequality."

In § 2 we give a proof of Theorem 1, followed by several remarks and corollaries in § 3.

2. Proofs. Let Y be a compact Hermitian operator with spectral decomposition

$$Y = \sum_j \lambda_j(\,.\,, e_j)e_j.$$

* Received by the editors November 10, 1988; accepted for publication (in revised form) April 12, 1989. The work presented in this paper was stimulated by the Mathematical Sciences Lecture Series on Matrix Spectral Inequalities at The Johns Hopkins University, June 20–24, 1988. The principal lecturer for the 1988 series was Professor Robert C. Thompson.

† Indian Statistical Institute, 7, SJS Sansanwal Marg, New Delhi 110016, India.

‡ Department of Mathematics, Kuwait University, P.O. Box 5969, 13060-Safat, Kuwait.

Let

$$Y_+ = \sum_{\lambda_j \geq 0} \lambda_j(.,e_j)e_j,$$

$$Y_- = -\sum_{\lambda_j < 0} \lambda_j(.,e_j)e_j.$$

Then Y_+ and Y_- are positive semidefinite compact operators and $Y = Y_+ - Y_-$. This is called the *Jordan decomposition* of Y. The following lemma is an easy consequence of the minmax principle. (See [3, p. 26]).

LEMMA 3. *Let Y be a compact Hermitian operator with Jordan decomposition $Y = Y_+ - Y_-$. Suppose Y can also be represented as $Y = Y_1 - Y_2$, where Y_1 and Y_2 are compact positive semidefinite operators. Then*

$$s_j(Y_+) \leqq s_j(Y_1) \quad \text{and} \quad s_j(Y_-) \leqq s_j(Y_2)$$

for all $j = 1, 2, \cdots$.

Proof of Theorem 1. Let A, B be given compact operators on a separable Hilbert space H. Let X be the operator on $H \oplus H$ with the block decomposition

$$X = \begin{bmatrix} A & B \\ O & O \end{bmatrix}.$$

Then

$$XX^* = \begin{bmatrix} AA^* + BB^* & O \\ O & O \end{bmatrix}, \qquad X^*X = \begin{bmatrix} A^*A & A^*B \\ B^*A & B^*B \end{bmatrix}.$$

Let I be the identity operator on H and let $U = \left[\begin{smallmatrix} I & O \\ O & -I \end{smallmatrix}\right]$. Then U is a unitary operator and

$$X^*X - U(X^*X)U^* = \begin{bmatrix} O & 2A^*B \\ 2B^*A & O \end{bmatrix} = Y, \text{ say.} \tag{5}$$

Note that the left-hand side of (5) gives a decomposition of Y as a difference of two positive semidefinite operators. Hence, using the above lemma we get

$$s_j(Y_+) \leqq s_j(X^*X) = s_j(XX^*) = s_j(AA^* + BB^*)$$

and

$$s_j(Y_-) \leqq s_j(UX^*XU^*) = s_j(X^*X) = s_j(AA^* + BB^*).$$

(Here, we have ignored zero singular values in the sense that XX^* is not really $AA^* + BB^*$ but $(AA^* + BB^*) \oplus O$. This has no effect on our argument.) The above two inequalities together imply

$$s_j(Y) \leqq s_j(Z), \qquad j = 1, 2, \cdots, \tag{6}$$

where

$$Z = \begin{bmatrix} AA^* + BB^* & O \\ O & AA^* + BB^* \end{bmatrix}.$$

But the singular values of Y are the singular values of $2A^*B$, each counted with twice the multiplicity. A similar consideration applies to Z. So (6) is equivalent to (1). □

Every unitarily invariant norm is a monotone function of the singular values of an operator. So Corollary 2 is an immediate consequence of Theorem 1.

3. Remarks. (1.) It is natural to wonder whether there is an operator inequality implying (1). More specifically, do we have

$$2|A^*B| \leqq AA^* + BB^*? \tag{7}$$

This turns out to be false even for Hermitian A, B, as seen from the example

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

Notice, however, that from (1) one can conclude that there exists a unitary operator U such that

$$2|A^*B| \leqq U(AA^* + BB^*)U^*. \tag{8}$$

To see this, just choose an orthonormal basis in which $|A^*B|$ is diagonal and a unitary operator U such that $U(AA^* + BB^*)U^*$ is also diagonal in this basis. For other results of this type, see [10], [11].

(2.) Every unitarily invariant norm has the property that $|||T||| = |||T^*|||$ for all T. Hence, from (2) it follows that

$$|||A^*B + B^*A||| \leqq |||AA^* + BB^*|||. \tag{9}$$

It follows from inequalities (11) and (16) in [1] that every unitarily invariant norm satisfies the following Cauchy–Schwarz type inequality:

$$|||X^*Y||| \leqq (|||X^*X||| \, |||Y^*Y|||)^{1/2} \tag{10}$$

for all X, Y.

Now, given A and B let

$$X = \begin{bmatrix} A & 0 \\ B & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} B & 0 \\ A & 0 \end{bmatrix}.$$

Then (10) gives

$$|||(A^*B + B^*A) \oplus O||| \leqq |||(A^*A + B^*B) \oplus O|||$$

for every unitarily invariant norm. Now note that if T and S are any two operators such that $|||T \oplus O||| \leqq |||S \oplus O|||$ for all unitarily invariant norms, then we also have $|||T||| \leqq |||S|||$ for all such norms. (This is a simple consequence of the fact that the family of inequalities $|||T||| \leqq |||S|||$ is equivalent to the weak majorization of the sequence $s_j(T)$ by $s_j(S)$, and this is unaffected by the addition or removal of zeros to both sides.) Thus we have

$$|||A^*B + B^*A||| \leqq |||A^*A + B^*B||| \tag{11}$$

for every unitarily invariant norm.

When A or B is nonnormal, (9) and (11) are not equivalent. This can be seen from the example

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

In [5] Horn and Mathias have shown that (10) is satisfied by several matrix norms other than the unitarily invariant ones. Call a norm ν on matrices a Cauchy–Schwarz norm if it satisfies (10) and if $\nu(T) \leqq \nu(S)$ whenever $\nu(T \oplus O) \leqq \nu(S \oplus O)$. Then the

above argument shows that (11) holds for all such norms. Examples of such norms may be found in [5].

(3.) For unitarily invariant norms, (11) follows from another argument. Notice that $(A \pm B)^*(A \pm B) \geqq 0$ and hence, $A^*A + B^*B \geqq \pm(A^*B + B^*A)$.

Next, suppose that X and Y are compact Hermitian operators with $\pm Y \leqq X$. Choose an orthonormal basis $\{e_j\}$ with $Ye_j = \lambda_j e_j$, $j = 1, 2, \cdots$, and $|\lambda_1| \geqq |\lambda_2| \geqq \cdots$. Then by a well-known property of singular values (see, e.g., [3]) we have for $k = 1, 2, \cdots$

$$\sum_{j=1}^{k} s_j(X) \geqq \sum_{j=1}^{k} (Xe_j, e_j) \geqq \sum_{j=1}^{k} |(Ye_j, e_j)| = \sum_{j=1}^{k} |\lambda_j| = \sum_{j=1}^{k} s_j(Y).$$

Hence $|||Y||| \leqq |||X|||$. This gives (11).

Note that $\pm Y \leqq X$ does not imply $|Y| \leqq X$. The example given in [6], namely,

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

shows that $\pm Y \leqq X$ does not imply even the weaker assertion $s_j(Y) \leqq s_j(X)$ for all j.

(4.) By a well-known result of Horn [4], we have the weak majorization $s_j(A^*B) \prec_w s_j(A^*)s_j(B)$. If we apply the arithmetic mean-geometric mean inequality for positive real numbers here, we get $2s_j(A^*B) \prec_w s_j(AA^*) + s_j(BB^*)$. The inequality (2) derived above is stronger than this majorization.

(5.) Using the above lemma it can be shown that for any positive semidefinite compact operators X, Y

$$|||X - Y||| \leqq |||X \oplus Y|||. \tag{12}$$

To see this, just decompose $X - Y$ into its Jordan parts.

As a corollary we obtain, for any compact operator A and for Schatten p-norms, $1 \leqq p \leqq \infty$,

$$\|A^*A - AA^*\|_p \leqq 2^{1/p}\|A^*A\|_p = 2^{1/p}\|A\|_{2p}^2. \tag{13}$$

The case $p = \infty$ of this inequality

$$\|A^*A - AA^*\| \leqq \|A\|^2$$

has been obtained by Fong [2], using a different method.

(6.) We remark that (12) is also true for general (i.e., not necessarily compact) positive semidefinite operators X, Y with the usual operator norm. To see this, let $Z = X - Y$. Then

$$Z \leqq X \leqq \|X\| I$$

and

$$-Z \leqq Y \leqq \|Y\| I.$$

These two inequalities together imply

$$\|X - Y\| = \|Z\| \leqq \max(\|X\|, \|Y\|) = \|X \oplus Y\|. \tag{14}$$

Now using (14), it follows from (5) that for any two operators A, B

$$2\|A^*B\| \leqq \|AA^* + BB^*\|. \tag{15}$$

(7.) In a recent paper [12] Yang has given a proof of the following fact: if A, B are positive semidefinite matrices, then

$$(\operatorname{tr} AB)^{1/2} \leqq \tfrac{1}{2}(\operatorname{tr} A + \operatorname{tr} B). \tag{16}$$

We should point out that much stronger results exist in the literature. From the majorization result of Horn [4] already mentioned above, we get, in particular,

$$\sum s_j(AB) \leqq \sum s_j(A)s_j(B),$$

and hence,

$$\begin{aligned}\left[\sum s_j(AB)\right]^{1/2} &\leqq \left[\sum s_j(A)s_j(B)\right]^{1/2} \\ &\leqq \sum \left[s_j(A)s_j(B)\right]^{1/2} \\ &\leqq \sum \tfrac{1}{2}\left[s_j(A)+s_j(B)\right].\end{aligned}$$

In other words, for any two matrices A and B we have

$$(\operatorname{tr} |AB|)^{1/2} \leqq \tfrac{1}{2}(\operatorname{tr} |A| + \operatorname{tr} |B|). \tag{17}$$

Since $|\operatorname{tr} X| \leqq \operatorname{tr} |X|$ for any matrix X, the inequality (16) is a special case of (17).

Even stronger than (17) is the inequality

$$\operatorname{tr} |AB|^{1/2} \leqq \tfrac{1}{2}(\operatorname{tr} |A| + \operatorname{tr} |B|). \tag{18}$$

This is true and is, in fact, a special case of a much more general result. In [1] it was shown that

$$|||\, |AB|^{1/2} \,||| \leqq (|||A|||\, |||B|||)^{1/2}$$

for every unitarily invariant norm. (This is another formulation of the Cauchy–Schwarz inequality mentioned earlier.) Apply the arithmetic-geometric mean inequality to get

$$|||\, |AB|^{1/2} \,||| < \tfrac{1}{2}(|||A||| + |||B|||), \tag{19}$$

for all unitarily invariant norms. The inequality (18) is the special case of (19) for the trace norm.

Acknowledgment. We are thankful to Dr. R. B. Bapat, who read an earlier version of this note and pointed out that our arguments therein led to (1), whereas we had only observed the weaker result (2).

REFERENCES

[1] R. BHATIA, *Perturbation inequalities for the absolute value map in norm ideals of operators*, J. Operator Theory, 19 (1988), pp. 129–136.

[2] C. K. FONG, *Norm estimates related to self commutators*, Linear Algebra Appl., 74 (1986), pp. 151–156.

[3] I. C. GOHBERG AND M. G. KREIN, *Introduction to the theory of linear nonselfadjoint operators*, in Translations of Mathematical Monographs, American Mathematical Society, Providence, RI, 1969.

[4] A. HORN, *On the singular values of a product of completely continuous operators*, Proc. Nat. Acad. Sci. U.S.A., 36 (1950), pp. 374–375.

[5] R. HORN AND R. MATHIAS, *An analog of the Cauchy–Schwarz inequality for Hadamard products and unitarily invariant norms*, Tech. Report 485, Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD, May 1988.

[6] F. KITTANEH, *Inequalities for the Schatten p-norm* IV, Comm. Math. Phys., 106 (1986), pp. 581–585.
[7] A. S. MARKUS, *The eigen- and singular values of the sum and product of linear operators*, Russian Math. Surveys, 19 (1964), pp. 92–120.
[8] A. W. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, New York, 1979.
[9] R. SCHATTEN, *Norm Ideals of Completely Continuous Operators*, Springer-Verlag, Berlin, New York, 1960.
[10] R. C. THOMPSON, *Matrix type metric inequalities*, Linear and Multilinear Algebra, 5 (1978), pp. 303–319.
[11] ———, *Matrix spectral inequalities*, Conference Lecture Notes, Mathematical Sciences Lecture Series, Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD, June 20–24, 1988.
[12] Y. YANG, *A matrix trace inequality*, J. Math. Anal. Appl., 133 (1988), pp. 573–574.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 278–300, April 1990

009

POINTS OF CONTINUITY OF THE KRONECKER CANONICAL FORM*

INMACULADA DE HOYOS†

Abstract. In this paper a map that associates with each matrix pencil another matrix pencil in canonical form for the strict equivalence of pencils (Kronecker canonical form) is defined. Then the pencils where this map is continuous are characterized.

The continuity of the canonical form obtained for the equivalence of matrix quadruples and triples from the Kronecker canonical form of the corresponding pencils is studied.

Key words. continuity, majorization, column and row minimal indices, r-numbers, s-numbers, Segre and Weyr characteristics of a finite or infinite eigenvalue

AMS(MOS) subject classifications. 15A60, 15A21, 93B10

Introduction. In the last few years there has been an interest in the study of the change of some canonical forms when small additive perturbations on the corresponding matrices are made. In [3, pp. 475–479] Gohberg, Lancaster, and Rodman, in [1] Den Boer and Thijsse, and in [6] Markus and Parilis give the results on the perturbation of the Jordan canonical form. In [5] the perturbation of the canonical form associated with the Γ-equivalence of matrix pairs is studied. The more general case is treated by Pokrzywa in [9], where some results about the perturbation of the Kronecker canonical form of matrix pencils are obtained.

All these results allow us to study the points of continuity of the maps that associate with each matrix A, each matrix pair (A, B), or each matrix pencil $H(\lambda)$, the corresponding canonical form. This was done for a matrix and a matrix pair in [4], and now our aim is to study what the pencils are like at which the Kronecker canonical form is continuous.

We will also study the points of continuity of the canonical forms (that we will call of Kronecker) associated with the equivalence of matrix quadruples (A, B, C, D) and with the equivalence of matrix triples (A, B, C). Both quadruples and triples appear in the study of linear multivariable systems (see [7], [8], and [10]), but we will introduce them as more general cases than that of matrix pairs and they will have a relation with particular cases of matrix pencils. Although we do not have results on perturbation of canonical forms of quadruples and triples, it is possible to study the continuity by means of the results in [9], [5], and [4].

In the first section we summarize some results of [4] and [9] that we will need in later sections, making the adequate changes in the notation.

In § 2, after analyzing some cases of matrix pencils where the Kronecker canonical form is not continuous, we characterize the points of continuity according to the relations between the number of rows and columns of the considered pencils.

In §§ 3 and 4, respectively, we study the relation between the equivalence of quadruples (of triples, respectively) and the strict equivalence of some pencils. Then we can characterize the points of continuity of the corresponding canonical forms, always depending on the relation among the numbers of rows and columns of the matrices A, B, C, and D for quadruples (A, B, and C for triples).

In all the cases the lower semicontinuity of the matrix rank, as a function of the matrix, is an essential fact.

* Received by the editors April 11, 1988; accepted for publication (in revised form) May 22, 1989.

† Departamento de Matemáticas, Facultad de Farmacia, Universidad del País Vasco, Apartado de Correos 450, E-01080 Vitoria-Gasteiz, Spain.

1. Preliminaries. If n is a nonnegative integer, then a *partition* of n is a decreasing infinite sequence of nonnegative integers nearly all being zero such that the sum of all the nonnull components is equal to n.

If $a = (a_1, a_2, \cdots)$ is a given partition, the *conjugate partition* $\bar{a}$ is the partition whose ith component is

$$\bar{a}_i := \text{Card}\,\{j : a_j \geqq i\}, \qquad i = 1, 2, \cdots.$$

If $a = (a_1, a_2, \cdots)$ and $b = (b_1, b_2, \cdots)$ are two given partitions, we will write $a \prec\prec b$ if

$$\sum_{i=1}^{j} a_i \leqq \sum_{i=1}^{j} b_i, \qquad j = 1, 2, \cdots.$$

If, further, the sums of all the nonnull components of each partition are equal, we will write $a \prec b$.

These two order relations are known, respectively, as *weak majorization* and *majorization*, because the second one implies the first one.

If A is a matrix, A^T denotes the tranpose matrix of A.

1.1. Continuity points of matrix canonical forms. In [4] it is shown that there does not exist any canonical form for the similarity of matrices (of order n, with entries in $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{C}$) being continuous everywhere. A *canonical form* is a map that associates a matrix with another similar one, and such that two matrices are similar if and only if they have the same associated matrix by this map.

Nevertheless, it is possible to characterize the points of continuity of some concrete canonical forms. For example, if we consider the rational canonical form defined following the divisibility order of the invariant factors of each matrix it turns out that the only continuity points are the nonderogatory matrices. If we define the Jordan canonical form, for $n \times n$ complex matrices, by ordering the eigenvalues according to the lexicographic order in $\mathbb{C}$, it is proved that it is continuous uniquely at the matrices with n eigenvalues of different real part. In an analogous way it is shown that the real Jordan canonical form is continuous at $A \in \mathbb{R}^{n \times n}$ if and only if the eigenvalues of A are simple and have different real parts, whenever they are not conjugate complex numbers (see [4]).

Finally matrix pairs (A, B) are considered, where A is an $n \times n$ matrix and B is an $n \times m$ matrix both of which have entries in $\mathbb{R}$ or $\mathbb{C}$, and a canonical form for the Γ-*equivalence of matrix pairs* is defined. This equivalence relation is also called *block similarity* and the canonical form is called the *Brunovsky canonical form* (see [3, pp. 193 and 196]). This canonical form is continuous at a pair (A, B) if and only if (A, B) is a completely controllable pair and its controllability indices constitute a partition of n that is minimal for the order relation of majorization, defined in the set of all the partitions of possible controllability indices for a completely controllable pair (A, B) (see [4]).

These results are based on [5] concerning the perturbation of linear control systems, that is to say on the change of the invariants for the Γ-equivalence under small perturbations of a matrix pair (A, B).

1.2. Strict equivalence of matrix pencils. The definitions and results that we give in this section are those that appear in [2, pp. 21–37] and [3, pp. 662–678], adapted to more adequate notation for our aim.

From now on $GL_n(\mathbb{C})$ denotes the linear group of order n over $\mathbb{C}$, and I the identity matrix of adequate order.

A *matrix pencil* is a matrix polynomial of degree one, denoted by $H(\lambda) = H_1 + \lambda H_2$, or simply by H.

Let $H(\lambda) = H_1 + \lambda H_2$ and $F(\lambda) = F_1 + \lambda F_2$ be two matrix pencils where H_1, H_2, F_1, and F_2 are complex matrices of size $m \times n$. The pencils $H(\lambda)$ and $F(\lambda)$ are said to be *strictly equivalent* if there exist $P \in GL_m(\mathbb{C})$ and $Q \in GL_n(\mathbb{C})$ such that $PH(\lambda)Q = F(\lambda)$, that is to say, $PH_1Q = F_1$ and $PH_2Q = F_2$.

We will call *normal rank of the matrix pencil* $H(\lambda) = H_1 + \lambda H_2$ the order of its greatest minor different from polynomial zero. We will denote it by rkn $(H(\lambda))$, or simply by rkn (H). The normal rank defined in this way coincides with the rank of $H(\lambda)$ considered as a matrix with entries in $\mathbb{C}(\lambda)$, field of quotients of $\mathbb{C}[\lambda]$. It is immediate that two strictly equivalent pencils have the same normal rank.

It is known that two pencils are strictly equivalent if and only if they have the same (column and row) minimal indices and the same (finite and infinite) elementary divisors. That is to say, a complete set of invariants for the relation of strict equivalence of pencils is formed by the following types of invariants, associated with each pencil H:

(i) *Column minimal indices* are denoted by

$$\varepsilon_1 \geqq \cdots \geqq \varepsilon_{r_1} > \varepsilon_{r_1+1} = \cdots = \varepsilon_{r_0} = 0$$

and we define, for $i = 0, 1, 2, \cdots$,

$$r_i := \text{Card}\,\{j : \varepsilon_j \geqq i\}.$$

Thus $\varepsilon := (\varepsilon_1, \cdots, \varepsilon_{r_1}, 0, \cdots)$ and $r := (r_1, \cdots, r_{\varepsilon_1}, 0, \cdots)$ are conjugate partitions. Moreover, $r_0 = n - \text{rkn}\,(H)$.

The partition r will be called *the partition of the r-numbers of the pencil H*. And if H has no column minimal indices, or equivalently, H has no r-numbers, we will consider $\varepsilon = (0) = (0, 0, \cdots)$ and $r = (0) = (0, 0, \cdots)$.

(ii) *Row minimal indices* denoted by

$$\eta_1 \geqq \cdots \eta_{s_1} > \eta_{s_1+1} = \cdots = \eta_{s_0} = 0$$

and we define, for $i = 0, 1, 2, \cdots$,

$$s_i := \text{Card}\,\{j : \eta_j \geqq i\}.$$

Thus $\eta := (\eta_1, \cdots, \eta_{s_1}, 0, \cdots)$ and $s := (s_1, \cdots, s_{\eta_1}, 0, \cdots)$ are conjugate partitions. Moreover, $s_0 = m - \text{rkn}\,(H)$.

The partition s will be called *the partition of the s-numbers of the pencil H*. And if H has no row minimal indices, or equivalently, H has no s-numbers, we will consider $\eta = (0)$ and $s = (0)$.

(iii) *Infinite elementary divisors* of the form

$$\mu^{n_{\infty 1}}, \cdots, \mu^{n_{\infty \nu_\infty}} \quad \text{with} \quad n_{\infty 1} \geqq \cdots \geqq n_{\infty \nu_\infty} \geqq 1.$$

We will say that $\eta_\infty := (\eta_{\infty 1}, \cdots, \eta_{\infty \nu_\infty}, 0, \cdots)$ is the *partition of the Segre characteristic of the pencil H for the eigenvalue infinite* and that its conjugate partition $m_\infty := (m_{\infty 1}, m_{\infty 2}, \cdots)$ is the *partition of the Weyr characteristic of the pencil H for the eigenvalue infinite*.

It follows that $m_{\infty 1} = \nu_\infty$ and $\nu_\infty = \text{rkn}\,(H) - \text{rk}\,(H_2)$. If ∞ is not an eigenvalue of H we will take $\eta_\infty = (0)$ and $m_\infty = (0)$.

(iv) *Finite elementary divisors* of the form

$$(\lambda - \lambda_1)^{n_{11}}, \cdots, (\lambda - \lambda_1)^{n_{1\nu_1}}, \cdots, (\lambda - \lambda_u)^{n_{u1}}, \cdots, (\lambda - \lambda_u)^{n_{u\nu_u}}$$

with

$$n_{i1} \geqq \cdots \geqq n_{i\nu_i} \geqq 1 \quad \text{for} \quad i = 1, \cdots, u.$$

We will say that $n_{\lambda_i} := (n_{i1}, \cdots, n_{iv_i}, 0, \cdots)$ is the *partition of the Segre characteristic of the pencil H for the eigenvalue* λ_i, of $H(i = 1, \cdots, u)$. The conjugate partition of n_{λ_i}, $m_{\lambda_i} := (m_{i1}, m_{i2}, \cdots)$ will be called the *partition of the Weyr characteristic of the pencil H for the eigenvalue* λ_i, of $H(i = 1, \cdots, u)$.

It follows that $m_{i1} = v_i \, (i = 1, \cdots, u)$.

We generalize the notation of (iii) and (iv) as follows. If $\alpha \in \bar{\mathbb{C}} := \mathbb{C} \cup \{\infty\}$, $n_\alpha := (n_{\alpha 1}, n_{\alpha 2}, \cdots)$ will be the partition of the Segre characteristic of H for α, if α is an eigenvalue of H (finite or infinite), and it will be the null partition (0) if α is not an eigenvalue of H. This is analogous for the conjugate partition $m_\alpha := (m_{\alpha 1}, m_{\alpha 2}, \cdots)$.

It is known (see [2], [3], and [9]) that each matrix pencil corresponds with a canonical form called the *Kronecker canonical form*, which is determined, except for the order of the blocks, by the invariants described above. The blocks associated with each type of invariant are as follows:

(i) If $\varepsilon_j \geqq 0$ is a column minimal index of the pencil, then the corresponding block R_{ε_j} has ε_j rows and $\varepsilon_j + 1$ columns and

$$R_{\varepsilon_j} := \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \lambda & 1 \end{bmatrix}.$$

If $\varepsilon_j = 0$ then R_0 is a column of zeros.

(ii) If $\eta_j \geqq 0$ is a row minimal index of the pencil, then the corresponding block $L_{\eta_j} := R_{\eta_j}^T$, that is to say, L_{η_j} is a block of dimension $(\eta_j + 1) \times \eta_j$. Analogously, if $\eta_j = 0$ then L_0 is a row of zeros.

(iii) If $\mu^{n_{\infty j}}$ is an infinite elementary divisor of the pencil, then the associated block $J_{n_{\infty j}}(\infty)$ is square of order $n_{\infty j}$ and

$$J_{n_{\infty j}}(\infty) := \begin{bmatrix} 1 & \lambda & & \\ & \ddots & \ddots & \\ & & 1 & \lambda \end{bmatrix}.$$

(iv) If $(\lambda - \alpha)^{n_{\alpha j}}$ is a finite elementary divisor of the pencil, then the associated block $J_{n_{\alpha j}}(\alpha)$ is square of order $n_{\alpha j}$ and is a Jordan block:

$$J_{n_{\alpha j}}(\alpha) := \begin{bmatrix} \lambda - \alpha & 1 & & \\ & \ddots & \ddots & \\ & & 1 & \\ & & & \lambda - \alpha \end{bmatrix}.$$

Remarks. (1) Instead of associating with the finite elementary divisors blocks $J_{n_{\alpha j}}(\alpha)$ we may consider the invariant factors of $H(\lambda)$ and associate with them blocks of the form $\lambda I - M_f$, where M_f is the companion matrix of one of the invariant factors.

In general, it is possible to take any pencil $\lambda I - M$ with M a matrix similar to that constituted by the Jordan blocks or to that formed by the companion matrices of the invariant factors.

(2) If a pencil only has column minimal indices and finite elementary divisors, as invariant factors, it turns out to be strictly equivalent to a pencil of the form $[-A, -B] + \lambda[I, 0]$, with A a square matrix. In this case the column minimal indices of the pencil coincide with the controllability indices of the pair (A, B) for the Γ-equivalence and the conjugate partition r is the partition of the r-numbers of (A, B) (see [5]). Moreover, the finite elementary divisors of the pencil are the elementary divisors of the pair (A, B) and thus, the Segre and Weyr characteristics of the pencil coincide with those of the pair (A, B). All this happens because the relation of strict equivalence of pencils is, in a certain sense, an extension of the Γ-equivalence of matrix pairs.

1.3. Perturbation of matrix pencils. A sequence of matrix pencils $H_{(k)}(\lambda) = H_{(k)1} + \lambda H_{(k)2}$, $k = 1, 2, \cdots$ is said to *converge* to a pencil $H(\lambda) = H_1 + \lambda H_2$ as $k \to \infty$ if the sequences of matrices $H_{(k)1}$ and $H_{(k)2}$ converge to H_1 and H_2, respectively, as $k \to \infty$. It is understood that the convergence is defined with respect to any matrix norm, for example, with respect to the matrix norm defined for $A = (a_{ij}) \in \mathbb{C}^{m\times n}$ as $\|A\| := \sum_{i,j} |a_{ij}|$, that will be employed from now on.

Let $\mathscr{P}_{m\times n}$ be the set of all the matrix pencils of size $m \times n$. We define in $\mathscr{P}_{m\times n}$ a metric d in the following way.

Let $H(\lambda) = H_1 + \lambda H_2 \in \mathscr{P}_{m\times n}$ and $F(\lambda) = F_1 + \lambda F_2 \in \mathscr{P}_{m\times n}$.

$$d(H(\lambda), F(\lambda)) := \|H(\lambda) - F(\lambda)\| := \|H_1 - F_1\| + \|H_2 - F_2\|,$$

where this matrix norm is the one defined above. The convergence of sequences of pencils obtained with this metric coincides with the convergence we have just defined.

If the sequence of pencils $H_{(k)}(\lambda) \in \mathscr{P}_{m\times n}$ converges to the pencil $H(\lambda) \in \mathscr{P}_{m\times n}$, as $k \to \infty$, we write $H_{(k)} \to H$.

If ε is a positive real number and $\alpha \in \bar{\mathbb{C}}$, then

$$B(\alpha, \varepsilon) := \{ z \in \mathbb{C} \| z - \alpha | < \varepsilon \} \quad \text{if } a \in \mathbb{C},$$

$$B(\alpha, \varepsilon) := \{ z \in \mathbb{C} \| z | > \varepsilon^{-1} \} \cup \{ \infty \} \quad \text{if } \alpha = \infty.$$

We will denote by $\sigma_J(H)$ the subset of $\bar{\mathbb{C}}$ formed by all the eigenvalues of the pencil H and we will define the *ε-neighborhood* of $\sigma_J(H)$ to be $\sigma_\varepsilon(H) := \cup_{\alpha\in\sigma_J(H)} B(\alpha, \varepsilon)$ and if there is no possibility of confusion we will write σ_ε.

The partition of the r-numbers and the s-numbers of the pencil $H_{(k)}$ will be denoted by $r_{(K)} := (r_{(k)1}, r_{(k)2}, \cdots)$ and $s_{(k)} := (s_{(k)1}, s_{(k)2}, \cdots)$, respectively. Given $\alpha \in \bar{\mathbb{C}}$, the partition of the Weyr characteristic of the pencil $H_{(k)}$ for α (that can be null, as we have seen) will be denoted by $m_{(k)\alpha} := (m_{(k)\alpha 1}, m_{(k)\alpha 2}, \cdots)$.

Taking this notation into account, we can enunciate the following lemma.

LEMMA 1.1. *Let $H_{(k)}$ be a sequence of pencils of $\mathscr{P}_{m\times n}$ converging to the pencil $H \in \mathscr{P}_{m\times n}$ and let σ_ε be the ε-neighborhood of $\sigma_J(H)$. Then for all sufficiently large k there exists a nonnegative integer h_k such that the following conditions hold*:

(i) $r \prec\prec r_{(k)} + (h_k, h_k, \cdots)$;

(ii) $s \prec\prec s_{(k)} + (h_k, h_k, \cdots)$; *and*

(iii) $m_{(k)\alpha} \prec\prec m_\alpha + (h_k, h_k, \cdots)$ *for all* $\alpha \in (\bar{\mathbb{C}} - \sigma_\varepsilon) \cup \sigma_J(H)$.

Remark. We take h_k as many times as min $\{n, m\}$ and the other components equal to zero so that $(h_k, h_k, \cdots)$ is a partition and the three conditions may hold.

Proof. This lemma is Theorem 1 of [9, p. 104] where the notation employed has been substituted by that defined in this paper by means of the following equalities:

$$r_i(H) := r_i - r_{i+1} \quad \text{for} \quad i = 0, 1, 2, \cdots,$$

$$l_i(H) := s_i - s_{i+1} \quad \text{for} \quad i = 0, 1, 2, \cdots,$$

$$r(H) := r_0,$$

$$\sum_{i \geq j} d_i(\alpha, H) := m_{\alpha j} \quad \text{for} \quad j = 1, 2, \cdots.$$

Equalities of the same type hold for $r_i(H_{(k)})$, $r_{(k)i}$, $r_{(k)i+1}$, etc. Moreover, taking into account that $r_0 = n - \operatorname{rkn}(H)$, $r_{(k)0} = n - \operatorname{rkn}(H_{(k)})$, $s_0 = m - \operatorname{rkn}(H)$, and $s_{(k)0} = m - \operatorname{rkn}(H_{(k)})$, we have that for all k sufficiently large

$$(r_1, r_2, \cdots) \prec\prec (r_{(k)1}, r_{(k)2}, \cdots) + (\operatorname{rkn}(H_{(k)}) - \operatorname{rkn}(H), \operatorname{rkn}(H_{(k)}) - \operatorname{rkn}(H), \cdots),$$

$$(s_1, s_2, \cdots) \prec\prec (s_{(k)1}, s_{(k)2}, \cdots) + (\operatorname{rkn}(H_{(k)}) - \operatorname{rkn}(H), \operatorname{rkn}(H_{(k)}) - \operatorname{rkn}(H), \cdots),$$

$$(m_{(k)\alpha 1}, m_{(k)\alpha 2}, \cdots) \prec\prec (m_{\alpha 1}, m_{\alpha 2}, \cdots) + (\operatorname{rkn}(H_{(k)}) - \operatorname{rkn}(H), \operatorname{rkn}(H_{(k)}) - \operatorname{rkn}(H), \cdots)$$

for all $\alpha \in (\bar{\mathbb{C}} - \sigma_\varepsilon) \cup \sigma_J(H)$.

By the lower semicontinuity of the normal rank of a pencil and for all k sufficiently large $\operatorname{rkn}(H) \leqq \operatorname{rkn}(H_{(k)})$. Taking $h_k := \operatorname{rkn}(H_{(k)}) - \operatorname{rkn}(H)$, the lemma is proved. □

Remark. Condition (iii) means that there can exist $\alpha \in \bar{\mathbb{C}} - \sigma_\varepsilon$ such that $m_{(k)\alpha}$ is not the null partition for some k sufficiently large. That is to say, $H_{(k)}$ may have as eigenvalues elements of $\bar{\mathbb{C}}$, finite or infinite, that are not eigenvalues of H and whose distance from all the eigenvalues of H is greater than a predetermined ε in the metric of $\bar{\mathbb{C}}$.

If we suppose that $\operatorname{rkn}(H_{(k)}) = \operatorname{rkn}(H)$, for all k sufficiently large, we can say something more about the invariants of pencils $H_{(k)}$ and especially about the eigenvalues. This hypothesis is equivalent to $r_{(k)0} = r_0$ (and to $s_{(k)0} = s_0$) for all k sufficiently large, which means that from one k all the pencils of the sequence have the same number of column minimal indices as H (and the same number of row minimal indices as H).

LEMMA 1.2. *Let $H_{(k)} \to H$ and suppose that $\operatorname{rkn}(H_{(k)}) = \operatorname{rkn}(H)$ for all k sufficiently large. Let $\varepsilon > 0$ be a real number such that the sets $B(\alpha, \varepsilon)$, with $\alpha \in \sigma_J(H)$, are pairwise disjoint and let σ_ε be the ε-neighborhood of $\sigma_J(H)$. Then there exists a k_0 such that $k \geqq k_0$ implies*

(i) $r \prec\prec r_{(k)}$;

(ii) $s \prec\prec s_{(k)}$;

(iii) $\sigma_J(H_{(k)}) \subset \sigma_\varepsilon$; *and*

(iv) $\sum_{\beta \in B(\alpha,\varepsilon)} \sum_j m_{(k)\beta j} \leqq \sum_j m_{\alpha j}$ *for all* $\alpha \in \sigma_J(H)$.

Remark. Condition (iv) means that the sum of the multiplicities of the eigenvalues of $H_{(k)}$, which are in $B(\alpha, \varepsilon)$, is less than or equal to the multiplicity of α as an eigenvalue of H.

Proof. Items (i) and (ii) are consequences of (i) and (ii) of Lemma 1.1, and (iii) and (iv) can be obtained from Theorem 2 of [9, p. 105]. □

Remarks. (1) The hypothesis of Lemma 1.2 is fulfilled, for example, when the pencil H is regular because in this case $\operatorname{rkn}(H) = n = m$ (or equivalently, $r_0 = s_0 = 0$) and by the lower semicontinuity of the normal rank of a pencil, $\operatorname{rkn}(H_{(k)}) = n = m$ for

all k sufficiently large. Therefore, if a pencil H is regular and $H_{(k)} \to H$ from a k_0 all the pencils $H_{(k)}$ are regular and, besides, condition (iv) of Lemma 1.2 changes into an equality as is shown in the remark of [9, p. 107]. That is to say, for all $k \geqq k_0$, $H_{(k)}$ has its eigenvalues in the neighborhoods of the eigenvalues of H, which are pairwise disjoint and predetermined, and the *sum of the multiplicities of the eigenvalues of* $H_{(k)}$ *in the neighborhood of an eigenvalue* α *of* H, *coincide with the multiplicity of* α *as an eigenvalue of* H.

(2) The condition $\operatorname{rkn}(H_{(k)}) = \operatorname{rkn}(H)$, or equivalently $r_{(k)0} = r_0$, is true for all pencil $H_1 + \lambda H_2$ associated with a pair of matrices (A, B); thus the results of Lemma 1.2 hold. Moreover, if we only perturbate H_1, the matrix that does not take a λ, we prove the existence of a neighborhood of the matrix $[A, B]$, with elements in $\mathbb{R}$ or $\mathbb{C}$, where some additional general *necessary conditions* are verified (see [5, Thm. 4.7]).

In order to study the continuity of the Kronecker canonical form we are interested in knowing not only what restrictions are verified by the invariants of the pencils of converging sequences, which are sufficiently close to the given pencil, but also if there exists, in any neighborhood of the given pencil, a pencil whose invariants are the previously determined by some so-called *sufficient conditions*. In fact, these sufficient conditions exist but they do not coincide with the necessary ones and they are more restrictive.

The partitions of invariants of a given pencil H are denoted by r, s, and $m_\alpha (\alpha \in \bar{\mathbb{C}})$. If F is another pencil we will denote by r' and s' the partitions of the r-numbers and the s-numbers of F, respectively, and by $m'_\alpha (\alpha \in \bar{\mathbb{C}})$ the partition of the Weyr characteristic of F for α, or the null partition according to what corresponds to α. Finally, we define $h := \operatorname{rkn}(F) - \operatorname{rkn}(H)$.

LEMMA 1.3. *Let H and F be two pencils; there exist a sequence of pencils $H_{(k)} \to H$ such that the pencils $H_{(k)}$ are strictly equivalent to the pencil F if and only if the three conditions hold*:

(i) $r \prec\prec r' + (h, h, \cdots)$;

(ii) $s \prec\prec s' + (h, h, \cdots)$; *and*

(iii) $m'_\alpha \prec\prec m_\alpha + (h, h, \cdots)$ *for all* $\alpha \in \bar{\mathbb{C}}$.

Proof. For the proof see Theorem 3 of [9, p. 108], and take h as many times as $\min\{n, m\}$. □

Since pencils $H_{(k)}$ are strictly equivalent to F for all k we have

$$r_{(k)} = r', s_{(k)} = s', m_{(k)\alpha} = m'_\alpha \quad \text{for all} \quad \alpha \in \bar{\mathbb{C}}.$$

In the particular case of a pencil associated with a matrix pair (A, B) these conditions reduce to (i) and (iii), with $h = 0$. Some less strong conditions are sufficient to prove the existence of a matrix $[A', B']$, in any neighborhood of $[A, B]$, such that (A', B') has the r-numbers and the Weyr characteristic predetermined (see [5, Thm. 5.6]).

2. Points of continuity of the canonical form for the strict equivalence of matrix pencils. We consider the metric space defined in § 1.3 on $\mathscr{P}_{m\times n}$, the set of all the complex matrix pencils of size $m \times n$. We will study the continuity of the following map:

$$C: \mathscr{P}_{m\times n} \to \mathscr{P}_{m\times n}, \qquad H \mapsto C(H)$$

where $C(H)$ is the *Kronecker canonical form* associated with the pencil H, defined in a unique way by taking its blocks in the following order:

First. Blocks R_{ε_j} associated with the column minimal indices of H, ordered decreasingly;

Second. Blocks L_{η_j} associated with the row minimal indices of H, ordered decreasingly;

Third. Blocks $J_{n_{\infty j}}$ associated with the infinite elementary divisors of H, following the decreasing order of their exponents;

Fourth. Blocks $J_{n_{\alpha j}}$ associated with the finite elementary divisors of H, ordered in the following way. Let $\lambda_1 < \lambda_2 < \cdots < \lambda_u$ be the different finite eigenvalues of H ordered according to the lexicographic order in $\mathbb{C}$; for each eigenvalue λ_i we order the corresponding blocks following the decreasing order of the integers of the partition of the Segre characteristic of H for the eigenvalue λ_i.

The following properties are equivalent.

(a) C is continuous at H;

(b) For each $\varepsilon > 0$ there exists $\delta > 0$ such that if $F \in \mathscr{P}_{m\times n}$ and $\|H - F\| < \delta$, then $\|C(H) - C(F)\| < \varepsilon$;

(c) For each sequence of pencils $H_{(k)} \in \mathscr{P}_{m\times n}$ converging to H the sequence of pencils $C(H_{(k)})$ converges to $C(H)$.

Employing the definition of continuity (b) and the inequality between the norm of a product of matrices and the product of the norms of the same matrices, it is easy to prove the following lemma.

LEMMA 2.1. *Let $P \in GL_m(\mathbb{C})$ and $Q \in GL_n(\mathbb{C})$; then C is continuous at H if and only if C is continuous at PHQ.*

Therefore, the continuity of C at a pencil H is equivalent to the continuity of C at the pencil $C(H)$ or at any pencil strictly equivalent to H.

Now, we will study some particular cases of pencils where C is not continuous.

LEMMA 2.2. *If H has one, or more, infinite elementary divisors, then C is not continuous at H.*

Proof. It suffices to prove the lemma for a pencil F in canonical form that has a single block J_{n_∞}, with $n_\infty \geqq 1$, i.e.,

$$F = \begin{bmatrix} 1 & \lambda & 0 & \cdots & 0 \\ 0 & 1 & \lambda & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Let

$$F_{(k)} := \begin{bmatrix} 1+\lambda/k & \lambda & 0 & \cdots & 0 \\ 0 & 1 & \lambda & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

thus $F_{(k)} \to F$ and, notwithstanding

$$C(F_{(k)}) = \begin{bmatrix} k+\lambda & 0 & 0 & \cdots & 0 \\ 0 & 1 & \lambda & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad \text{for all } k.$$

Therefore $C(F_{(k)})$ does not converge to $C(F) = F$, i.e., C is not continuous at F. □

LEMMA 2.3. *If H has finite elementary divisors and column minimal indices, then C is not continuous at H.*

Proof. It is enough to show the lemma for a pencil H in canonical form that has only blocks of the types R_{ε_j} and $J_{n_{\infty j}}$. If H is $m \times n$, in this case $m < n$.

Let r' be a partition of m such that $r \prec\prec r'$ and $\sum_i r'_i = m$ and let $m'_\alpha := (0, 0, \cdots)$ for all $\alpha \in \bar{\mathbb{C}}$. By Lemma 1.3 there exists a sequence of pencils $H_{(k)}$ converging to H such that for all k, the pencil $H_{(k)}$ has, as only invariants, the column minimal indices corresponding to the conjugate partition r'. Thus the sequence of pencils $C(H_{(k)})$ does not converge to $C(H) = H$. □

LEMMA 2.4. *If H has finite elementary divisors and row minimal indices, then C is not continuous at H.*

Proof. To prove the lemma, take the transpose pencil of H, H^T, and apply Lemma 2.3. □

LEMMA 2.5. *If H has column and row minimal indices, then C is not continuous at H.*

Proof. As in the previous cases we consider a pencil H in canonical form with a single block R_{ε_1} and a single block L_{η_1}, with $\varepsilon_1 \geqq 0$ and $\eta_1 \geqq 0$. Then

$$H = \left[\begin{array}{cccccc|cccc} \lambda & 1 & 0 & \cdots & 0 & 0 & & & & \\ 0 & \lambda & 1 & \cdots & 0 & 0 & & & & \\ \cdot & \cdot & \cdot & & \cdot & \cdot & & & & \\ \cdot & \cdot & \cdot & & \cdot & \cdot & & & & \\ 0 & 0 & 0 & \cdots & \lambda & 1 & & & & \\ \hline & & & & & & \lambda & 0 & \cdots & 0 \\ & & & & & & 1 & \lambda & \cdots & 0 \\ & & & & & & 0 & 1 & \cdots & 0 \\ & & & & & & \cdot & \cdot & & \cdot \\ & & & & & & \cdot & \cdot & & \cdot \\ & & & & & & 0 & 0 & \cdots & \lambda \\ & & & & & & 0 & 0 & \cdots & 1 \end{array}\right].$$

has $\varepsilon_1 + n_1 + 1$ rows and columns.

Let $H_{(k)}$ be the pencil obtained by adding the number $1/k$ to the entry of H in position $(\varepsilon_1 + 1, 1)$. It is clear that $H_{(k)} \to H$ but for all k, the pencils $H_{(k)}$ have an only invariant that is the infinite elementary divisor $\mu^{\varepsilon_1 + \eta_1 + 1}$, i.e., the sequence of pencils $C(H_{(k)})$ does not converge to $C(H) = H$. □

A consequence of Lemmas 2.2–2.5 is the following necessary condition for the continuity of C at a pencil H.

COROLLARY 2.6. *If C is continuous at H then one of the following properties hold:*

(i) *H has only column minimal indices;*

(ii) *H has only row minimal indices;*

(iii) *H has only finite elementary divisors.*

We will analyze separately each of the three possible cases described in Corollary 2.6, because the relation between the number of rows m and the number of columns n of pencils of $\mathscr{P}_{m \times n}$ is different in each case: (i) occurs when $n > m$, (ii) when $m > n$, and (iii) when $n = m$.

Notation. If a and b are two positive integers, the equality $a = bc + d$ means that c and d are integers verifying

$$c \geqq 1 \text{ and } 0 \leqq d < b, \text{ if } a \geqq b \quad \text{or} \quad c = 0 \text{ and } d = a, \text{ if } a < b.$$

THEOREM 2.7. *Let* $C : \mathscr{P}_{m\times n} \to \mathscr{P}_{m\times n}$ *with* $n > m$. *Let* c *and* d *be nonnegative integers such that* $m = (n-m)c + d$. *Then* C *is continuous at* $H \in \mathscr{P}_{m\times n}$ *if and only if* H *has as only invariants the following column minimal indices*:

$$\varepsilon_1 = \cdots = \varepsilon_d = c+1, \varepsilon_{d+1} = \cdots = \varepsilon_{n-m} = c.$$

Remark. This condition for the column minimal indices is equivalent to the following condition for the r-numbers:

$$r_1 = \cdots = r_c = n-m \quad \text{and} \quad r_{c+1} = d,$$

that is to say, the partition of the r-numbers is maximal for the majorization.

Proof. First, we will show that if H is as given in Theorem 2.7, then C is continuous at H. In this case rkn $(H) = m$, i.e., it is maximum and we can apply Lemma 1.2. Thus for all sequences $H_{(k)} \to H$ and for all k sufficiently large, $r_{(k)} = r$ where r is the partition of the r-numbers of H, which appear in the remark above. Therefore for all k sufficiently large, $C(H_{(k)}) = C(H)$, i.e., C is continuous at H.

Now, if C is continuous at $H \in \mathscr{P}_{m\times n}$ with $n > m$, by Corollary 2.6 H has only column minimal indices. Suppose that they are not those enunciated, i.e., the partition of the r-numbers of H, r, is not maximal for the majorization. Let r' be the conjugate partition of

$$(c+1, \overset{(d)}{\cdots}, c+1, c, \overset{(n-m-d)}{\cdots\cdots}, c, 0, \cdots), \quad \text{i.e.,}$$

$r'_1 = \cdots = r'_c = n - m$ and $r'_{c+1} = d$. Then $r \prec r'$, which implies $r \prec\prec r'$.

By Lemma 1.3 there exists a sequence $H_{(k)} \to H$ such that the pencils $H_{(k)}$ have only column minimal indices and the partition of the r-numbers of $H_{(k)}$ is r'. Thus $C(H_{(k)})$ does not converge to $C(H)$, which contradicts the fact that C is continuous at H. □

THEOREM 2.8. *Let* $C : \mathscr{P}_{m\times n} \to \mathscr{P}_{m\times n}$ *with* $m > n$. *Let* c *and* b *be nonnegative integers such that* $n = (m-n)c + d$. *Then* C *is continuous at* $H \in \mathscr{P}_{m\times n}$ *if and only if* H *has as only invariants the following row minimal indices*:

$$\eta_1 = \cdots = \eta_d = c+1, \qquad \eta_{d+1} = \cdots = \eta_{m-n} = c.$$

Remark. As in Theorem 2.7, this condition is equivalent to the following one:

$$s_1 = \cdots = s_c = m-n \quad \text{and} \quad s_{c+1} = d.$$

Proof. Studying the continuity of C is equivalent to studying the continuity of the map

$$C^T : \mathscr{P}_{n\times m} \to \mathscr{P}_{n\times m}, \qquad F \mapsto C^T(F) := [C(F^T)]^T$$

because C is continuous at H if and only if C^T is continuous at H^T, the transpose pencil of H. Since $m > n$, C^T is defined for pencils that have more columns than rows, so that the continuity of C^T is studied in Theorem 2.7. □

If $\alpha \in \mathbb{C}$, we denote by Re (α) the *real part* of α.

THEOREM 2.9. *Let* $C : \mathscr{P}_{n\times n} \to \mathscr{P}_{n\times n}$; C *is continuous at* $H \in \mathscr{P}_{n\times n}$ *if and only if* H *has as only invariants* n *finite elementary divisors of the form*:

$$\lambda - \lambda_1, \cdots, \lambda - \lambda_n \quad \textit{with } \operatorname{Re}(\lambda_i) \neq \operatorname{Re}(\lambda_j) \quad \textit{for all } i \neq j.$$

Proof. If H has the indicated form, let $\lambda_1 < \cdots < \lambda_n$. Let $H_{(k)} \to H$ and let ε be any positive real number. We take

$$\varepsilon'' := \frac{1}{3}\min\{\operatorname{Re}(\lambda_{i+1}) - \operatorname{Re}(\lambda_i) \mid i = 1, \cdots, n-1\}, \quad \text{and}$$

$$\varepsilon' := \min\left\{\varepsilon'', \frac{\varepsilon}{2n}\right\}.$$

Let $\sigma_{\varepsilon'} := \cup_{i=1}^{n} B(\lambda_i, \varepsilon')$; then by Lemma 1.2 and by Remark (1), made after its proof, there exists a k_0 such that for all $k \geqq k_0$

$$\sigma_J(H_{(k)}) \cap B(\lambda_i, \varepsilon') = \{\alpha_{(k)i}\}$$

where $\alpha_{(k)1} < \cdots < \alpha_{(k)n}$ and $\alpha_{(k)i}$ has multiplicity equal to one for $i = 1, \cdots, n$.

Thus for all $k \geqq k_0$ $\|C(H_{(k)}) - C(H)\| < \varepsilon$, i.e., $C(H_{(k)}) \to C(H)$ and, therefore, C is continuous at H.

If C is continuous at $H \in \mathscr{P}_{n\times n}$, by Corollary 2.6, H has only finite elementary divisors. If they are not as indicated we consider the matrix pencil $-J + \lambda I$, where J is the $n \times n$ complex matrix in Jordan canonical form corresponding to the finite elementary divisors of H. Then J does not verify the necessary (and sufficient) conditions for the continuity of the Jordan canonical form at J [4, Thm. 2]. Thus there exists a positive real number ε such that we can find a matrix J', as close to J as we want, which verifies that the distance between J and the Jordan canonical form of J' is greater than ε. Therefore, there exists a matrix pencil $-J' + \lambda I$, as close to $-J + \lambda I$ as we want, such that the distance between the Kronecker canonical form of $-J + \lambda I$ and the Kronecker canonical form of $-J' + \lambda I$ is greater than ε. That is to say, C is not continuous at $-J + \lambda I$ and therefore C is not continuous at H, because H and $-J + \lambda I$ are strictly equivalent. □

Remark. If for H having only finite elementary divisors we define $C(H) = \lambda I - M$, where M is the matrix with blocks that are companion matrices of the invariant factors of H, then the necessary and sufficient condition for the continuity of $C: \mathscr{P}_{n\times n} \to \mathscr{P}_{n\times n}$ at H is that H has only an invariant factor different from one, that is to say, the corresponding matrix M (which is in rational canonical form) has a single block (i.e., M is nonderogatory).

3. Points of continuity of the canonical form for the equivalence of matrix quadruples. In this section we will consider *matrix quadruples* of the form (A, B, C, D) with $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, $C \in \mathbb{C}^{p\times n}$, and $D \in \mathbb{C}^{p\times m}$.

DEFINITION. Two matrix quadruples (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) are said to be *equivalent* if there exist matrices $P \in GL_n(\mathbb{C})$, $Q \in GL_p(\mathbb{C})$, $T \in GL_m(\mathbb{C})$, $R \in \mathbb{C}^{n\times p}$, and $S \in \mathbb{C}^{m\times n}$ such that

$$\begin{bmatrix} P & R \\ 0 & Q \end{bmatrix}\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}\begin{bmatrix} P^{-1} & 0 \\ S & T \end{bmatrix} = \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}.$$

Note that the matrices

$$\begin{bmatrix} P & R \\ 0 & Q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} P^{-1} & 0 \\ S & T \end{bmatrix}$$

are square and nonsingular.

PROPOSITION 3.1. *(A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) are equivalent if and only if (A_2, B_2, C_2, D_2) can be obtained from (A_1, B_1, C_1, D_1) by means of one, or more, of the following elementary transformations*:

(1) $(A_1, B_1, C_1, D_1) \to (A_2, B_2, C_2, D_2) = (PA_1P^{-1}, PB_1, C_1P^{-1}, D_1)$,

(2) $(A_1, B_1, C_1, D_1) \to (A_2, B_2, C_2, D_2) = (A_1 + RC_1, B_1 + RD_1, C_1, D_1)$,

(3) $(A_1, B_1, C_1, D_1) \to (A_2, B_2, C_2, D_2) = (A_1, B_1, QC_1, QD_1)$,

(4) $(A_1, B_1, C_1, D_1) \to (A_2, B_2, C_2, D_2) = (A_1 + B_1S, B_1, C_1 + D_1S, D_1)$,

(5) $(A_1, B_1, C_1, D_1) \to (A_2, B_2, C_2, D_2) = (A_1, B_1T, C_1, D_1T)$,

where P, Q, T, R, and S are matrices as those considered in the previous definition.

Proof. It suffices to take into account that

$$\begin{bmatrix} P & R \\ 0 & Q \end{bmatrix}\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}\begin{bmatrix} P^{-1} & 0 \\ S & T \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ 0 & Q \end{bmatrix}\begin{bmatrix} I & R \\ 0 & I \end{bmatrix}\begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}\begin{bmatrix} P^{-1} & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} I & 0 \\ S & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix},$$

with I being the identity matrices of adequate sizes. □

This equivalence relation defined for matrix quadruples is a generalization of the Γ-equivalence defined for matrix pairs and, at the same time, it corresponds to a particular case of strict equivalence of matrix pencils as we will now see.

Notation. Let $(A_i, B_i, C_i, D_i) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times m} \times \mathbb{C}^{p\times n} \times \mathbb{C}^{p\times m}$. We define

$$E := \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}, \qquad G_i := \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix},$$

where E and G_i are complex matrices of size $(n + p) \times (n + m)$, $i = 1, 2$.

PROPOSITION 3.2. *Two quadruples (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) are equivalent if and only if the pencils $G_1 + \lambda E$ and $G_2 + \lambda E$ are strictly equivalent.*

Proof. If the quadruples are equivalent by means of the nonsingular square matrices

$$\begin{bmatrix} P & R \\ 0 & Q \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} P^{-1} & 0 \\ S & T \end{bmatrix},$$

then these matrices make $G_1 + \lambda E$ and $G_2 + \lambda E$ be strictly equivalent.

Conversely, if $G_1 + \lambda E$ and $G_2 + \lambda E$ are strictly equivalent, there exist matrices $U \in GL_{n+p}(\mathbb{C})$ and $V \in GL_{n+m}(\mathbb{C})$ such that (i) $UG_1V = G_2$ and (ii) $UEV = E$.

If we take

$$U = \begin{bmatrix} U_1 & U_2 \\ U_3 & U_4 \end{bmatrix} \quad \text{and} \quad V = \begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix}$$

with U_1 and $V_1 \in \mathbb{C}^{n\times n}$, then from (i) and (ii) we deduce that $V_1 = U_1^{-1}$, $V_2 = 0$, and $V_3 = 0$. Then by (i) the quadruples are equivalent. □

Let $H(\lambda) = H_1 + \lambda H_2$ be a matrix pencil of size $(n + p) \times (n + m)$ such that $\operatorname{rk}(H_2) = n$; then there exist two nonsingular matrices P and Q, of adequate sizes, such that $PH_2Q = E$. If we take $G := PH_1Q$ it turns out that $H(\lambda)$ is strictly equivalent to $G + \lambda E$. If

$$G = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

with $A \in C^{n\times n}$ we will say that *the quadruple* (A, B, C, D) is *associated* with the pencil $H(\lambda)$.

A pencil may have different associated quadruples but all of them will be equivalent by Proposition 3.2. Moreover, if we consider two pencils and two quadruples, each one of them associated with each pencil, then the quadruples are equivalent if and only if the pencils are strictly equivalent. This is also deduced from Proposition 3.2. and from the definition of a quadruple associated with a pencil.

As a consequence, we have that the invariants and the canonical form for the equivalence of quadruples will be obtained by means of the invariants and the Kronecker canonical form of the pencils that have them as associated quadruples. So it is enough to study the pencils of the form $G + \lambda E$.

We consider matrices

$$G = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+m)}, \quad \text{i.e., } A \in \mathbb{C}^{n\times n}, B \in \mathbb{C}^{n\times m}, C \in \mathbb{C}^{p\times n}, \text{ and } D \in \mathbb{C}^{p\times m}.$$

We will employ the matrix norm $\|G\| = \sum_{i,j} |g_{ij}|$ for $G = (g_{ij}) \in \mathbb{C}^{(n+p)\times(n+m)}$.

Our aim is to study the continuity of the map

$$C_q : \mathbb{C}^{(n+p)\times(n+m)} \to \mathbb{C}^{(n+p)\times(n+m)},$$

which associates with each matrix

$$G = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

its canonical form

$$\begin{bmatrix} A_q & B_q \\ C_q & D_q \end{bmatrix},$$

for the equivalence of quadruples. This canonical form is defined, from the *Kronecker canonical form* for pencils, in the following unique way:

Let $H(\lambda) = G + \lambda E$ be the corresponding pencil of size $(n + p) \times (n + m)$ and let

(i) $\varepsilon_1 \geqq \cdots \geqq \varepsilon_{r_1} > \varepsilon_{r_1+1} = \cdots = \varepsilon_{r_0} = 0$ be the column minimal indices of $H(\lambda)$;

(ii) $\eta_1 \geqq \cdots \geqq \eta_{s_1} > \eta_{s_1+1} = \cdots = \eta_{s_0} = 0$ be the row minimal indices of $H(\lambda)$;

(iii) $n_{\infty 1} \geqq \cdots \geqq n_{\infty \nu_\infty} \geqq 1$ be the exponents of the infinite elementary divisors of $H(\lambda)$ and $d_j := n_{\infty j} - 1$ such that $d_1 \geqq \cdots \geqq d_{t_1} > d_{t_1+1} = \cdots = d_{t_0} = 0$, $t_0 := \nu_\infty$; and

(iv) $(\lambda + \lambda_1)^{n_{11}}, \cdots, (\lambda + \lambda_1)^{n_{1\nu_1}}, \cdots, (\lambda + \lambda_u)^{n_{u1}}, \cdots, (\lambda + \lambda_u)^{n_{u\nu_u}}$, with $n_{i1} \geqq \cdots \geqq n_{i\nu_i}$ for $i = 1, \cdots, u$, be the finite elementary divisors of $H(\lambda)$.

We define $A_q := \operatorname{diag}(A_\varepsilon, A_\eta, A_\infty, A_f) \in \mathbb{C}^{n\times n}$, where the square blocks A_ε, A_η, A_∞, and A_f are, respectively, of order $\sum_{j=1}^{r_1} \varepsilon_j$, $\sum_{j=1}^{s_1} \eta_j$, $\sum_{j=1}^{t_1} d_j$, and $\sum_{i=1}^{u} \sum_{j=1}^{\nu_i} n_{ij}$.

Moreover, if

$$M_i := \begin{bmatrix} 0 & I_{i-1} \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{i\times i} \quad \text{and} \quad N_i := \begin{bmatrix} 0 & 0 \\ I_{i-1} & 0 \end{bmatrix} \in \mathbb{C}^{i\times i}$$

then

$$A_\varepsilon := \operatorname{diag}(M_{\varepsilon_1}, \cdots, M_{\varepsilon_{r_1}}),$$

$$A_\eta := \operatorname{diag}(N_{\eta_1}, \cdots, N_{\eta_{s_1}}), \quad \text{and}$$

$$A_\infty := \operatorname{diag}(N_{d_1}, \cdots, N_{d_{t_1}}).$$

$$A_f := \operatorname{diag}(\lambda_1 I_{n_{11}} + M_{n_{11}}, \cdots, \lambda_1 I_{n_{1\nu_1}} + M_{n_{1\nu_1}}, \cdots, \lambda_u I_{n_{u1}} + M_{n_{u1}}, \cdots, \lambda_u I_{n_{u\nu_u}} + M_{n_{u\nu_u}})$$

where $\lambda_1 < \cdots < \lambda_u$ is the lexicographic order in $\mathbb{C}$. That is to say, the finite elementary divisors of A_f are $(\lambda - \lambda_1)^{n_{11}}, \cdots, (\lambda - \lambda_1)^{n_{1v_1}}, \cdots, (\lambda - \lambda_u)^{n_{u1}}, \cdots, (\lambda - \lambda_u)^{n_{uv_u}}$, with $n_{i1} \geqq \cdots \geqq n_{iv_i}$ for $i = 1, \cdots, u$.

We define

$$B_q := \begin{bmatrix} B_\varepsilon & 0 \\ 0 & 0 \\ 0 & B_\infty \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{n \times n}$$

where B_ε and B_∞ are blocks of sizes $[\sum_{j=1}^{r_1} \varepsilon_j] \times r_0$ and $[\sum_{j=1}^{t_1} d_j] \times t_0$, respectively; between B_ε and B_∞ there are $\sum_{j=1}^{s_1} \eta_j$ zero rows and below B_∞ there are $\sum_{i=j}^{u} \sum_{j=1}^{v_i} n_{ij}$ zero rows.

Moreover, if e_j is the row vector with an adequate number of components that are all zero except for the jth component that is equal to one, and we define

$$E_j := \begin{bmatrix} 0 \\ e_j \end{bmatrix} \in \mathbb{C}^{\varepsilon_j \times r_0} \quad \text{and} \quad F_j := \begin{bmatrix} e_j \\ 0 \end{bmatrix} \in \mathbb{C}^{d_j \times t_0},$$

then

$$B_\varepsilon := \begin{bmatrix} E_1 \\ \vdots \\ E_{r_1} \end{bmatrix} \quad \text{and} \quad B_\infty := \begin{bmatrix} F_1 \\ \vdots \\ F_{t_1} \end{bmatrix}.$$

We define

$$C_q := \begin{bmatrix} 0 & C_\eta & 0 & 0 \\ 0 & 0 & C_\infty & 0 \end{bmatrix} \in \mathbb{C}^{p \times n},$$

where C_η and C_∞ are blocks of sizes

$$s_0 \times \left[\sum_{j=1}^{s_1} \eta_j \right] \quad \text{and} \quad t_0 \times \left[\sum_{j=1}^{t_1} d_j \right],$$

respectively; the first $\sum_{j=1}^{r_1} \varepsilon_j$ columns are zero and the last $\sum_{i=1}^{u} \sum_{j=1}^{v_i} n_{ij}$ columns are also zero.

Moreover $C_\eta := [E_1^T, \cdots, E_{s_1}^T]$ with $E_j^T \in \mathbb{C}^{s_0 \times \eta_j}$ $(j = 1, \cdots, s_1)$, E_j^T being the transpose matrix of E_j defined above, but with the size now required. Analogously, $C_\infty := [E_1^T, \cdots, E_{t_1}^T]$ with $E_i^T \in \mathbb{C}^{t_0 \times d_j}$ $(j = 1, \cdots, t_1)$.

We define

$$D_q := \begin{bmatrix} 0 & 0 \\ 0 & I_{t_0 - t_1} \end{bmatrix} \in \mathbb{C}^{p \times m}.$$

Remarks. (1) Since the only transformations that may change the matrix D of a quadruple are those of types (3) and (5), where D is multiplied by nonsingular matrices, it turns out that rk (D) is an invariant of the equivalence of quadruples and it holds that

$$\text{rk}\,(D) = t_0 - t_1 = m_{\infty 1} - m_{\infty 2},$$

that is to say, rk (D) coincide with the number of infinite elementary divisors of exponent one.

(2) Also, it is easy to see that

(i) rkn $(G + \lambda E) = n + t_0$;

(ii) $m = r_0 + t_0$;

(iii) $p = s_0 + t_0$.

(3) The block A_f, which is in Jordan canonical form, might be taken in any other canonical form of A_f for the relation of similarity of matrices, as we noted for pencils. In each case the results of this study of the continuity would change in the corresponding way.

DEFINITION. From now on *quadruple* will mean (A, B, C, D) or the corresponding matrix G. We will call *column minimal indices*, *r-numbers*, *row minimal indices*, *s-numbers*, *Segre and Weyr characteristics* of G for the *eigenvalue infinite*, the column minimal indices, r-numbers, and so on, of the pencil $G + \lambda E$. Finally the *Segre* and *Weyr characteristics* of G for an *eigenvalue* $\alpha \in \mathbb{C}$ will be the Segre and Weyr characteristics of $G + \lambda E$ for the eigenvalue $-\alpha$.

Other ways of obtaining the invariants and a canonical form for the equivalence of matrix quadruples can be found in [7] and [10].

The results on perturbation of pencils that give necessary conditions that must be verified by the pencils converging to a given pencil, as, for example, Lemmas 1.1 and 1.2, are still true for quadruples; because making small perturbations on G is equivalent to making them on the matrix of pencil $G + \lambda E$ that does not go with λ. Nevertheless, Lemma 1.3 does not hold for quadruples because now we need to obtain a converging sequence of pencils by perturbing only the matrix that does not go with λ. But there is absolutely no problem, for we have theorems on perturbation of the Jordan canonical form and of the canonical form for the Γ-equivalence of matrix pairs. These forms are particular cases of the canonical form C_q defined for matrix quadruples.

As it happened for C it is true that C_q is continuous at a quadruple if and only if C_q is continuous at any equivalent quadruple.

Before characterizing the quadruples where C_q is continuous it is convenient to know some types of quadruples where C_q is not continuous.

LEMMA 3.3. *Let*

$$G = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+m)}.$$

Suppose that G satisfies at least one of the following properties:

(i) *G has infinite elementary divisors with exponent greater than one*;

(ii) *G has finite elementary divisors and column minimal indices*;

(iii) *G has finite elementary divisors and row minimal indices*;

(iv) *G has column and row minimal indices*.

Then C_q is not continuous at G.

Proof. (i) First suppose that G has only one elementary divisor μ^2 and that G is in canonical form, i.e.,

$$G = \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & 0 \end{array}\right] \in \mathbb{C}^{(1+1)\times(1+1)}.$$

Let

$$G_k := \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & \frac{1}{k} \end{array}\right] \in \mathbb{C}^{(1+1)\times(1+1)};$$

then

$$C_q(G_k) = \left[\begin{array}{c|c} -k & 0 \\ \hline 0 & 1 \end{array}\right],$$

which is at a distance greater than three from $C_q(G) = G$, i.e., $\|C_q(G_k) - C_q(G)\| > 3$. Thus C_q is not continuous at G.

If G has one or more infinite elementary divisors with exponent greater than two we act in the same way.

(ii) Let $G = C_q(G)$ with finite elementary divisors and column minimal indices. We take the submatrix of G,

$$\left[\begin{array}{cc|c} A_\varepsilon & 0 & B_\varepsilon \\ 0 & A_f & 0 \end{array}\right].$$

This matrix is associated with the pair of matrices

$$\left(\begin{bmatrix} A_\varepsilon & 0 \\ 0 & A_f \end{bmatrix}, \begin{bmatrix} B_\varepsilon \\ 0 \end{bmatrix}\right),$$

which is not completely controllable.

By Theorem 5.6 of [5] we can find, as close to

$$\left[\begin{array}{cc|c} A_\varepsilon & 0 & B_\varepsilon \\ 0 & A_f & 0 \end{array}\right]$$

as we want, a matrix of the same size such that the corresponding pair is completely controllable, i.e., it does not have finite elementary divisors. So we obtain quadruples, as close to G as we want, such that its image by C_q is at a distance greater than or equal to one from $C_q(G) = G$. See also Theorems 9 and 10 of [4].

(iii) In this case it is enough to consider the quadruple $G^T \in \mathbb{C}^{(n+m)\times(n+p)}$ and apply case (ii) for C_q defined and with values in $\mathbb{C}^{(n+m)\times(n+p)}$, because G^T is a quadruple having finite elementary divisors and column minimal indices.

(iv) If $C_q(G) = G$ and G has only column and row minimal indices it suffices to act as in Lemma 2.5, because there we perturbed only the matrix of the pencil that does not go with λ. □

COROLLARY 3.4. *If C_q is continuous at G then one of the following properties hold*:

(i) *G has only column minimal indices and infinite elementary divisors with exponent one*;

(ii) *G has only row minimal indices and infinite elementary divisors with exponent one*;

(iii) *G has only finite elementary divisors and infinite elementary divisors with exponent one*.

These three cases hold, respectively, when one of the following relations is true: (i) $m > p$, (ii) $p > m$, or (iii) $m = p$.

We will study them in the following three theorems.

THEOREM 3.5. *Let $C_q : \mathbb{C}^{(n+p)\times(n+m)} \to \mathbb{C}^{(n+p)\times(n+m)}$ with $m > p$. Let c and d be nonnegative integers such that $n = (m-p)c + d$. Then C_q is continuous at $G \in \mathbb{C}^{(n+p)\times(n+m)}$*

if and only if G has as only invariants p infinite elementary divisors with exponent one and the following column minimal indices:

$$\varepsilon_1 = \cdots = \varepsilon_d = c+1, \qquad \varepsilon_{d+1} = \cdots = \varepsilon_{m-p} = c.$$

Remark. As in Theorem 2.7, the column minimal indices are those indicated if and only if the partition of r-numbers is maximal for the majorization, i.e.,

$$r_1 = \cdots = r_c = m-p \quad \text{and} \quad r_{c+1} = d.$$

Proof. If G has the indicated form it turns out that $t_0 = p$, i.e., t_0 is maximum. Thus $\operatorname{rkn}(G + \lambda E)$ is maximum and we can apply Lemma 1.2. Since $\operatorname{rk}(D) = p$, i.e., is maximum and the r-numbers constitute a partition that is maximal for the majorization, we have that given any sequence $G_k \to G$ there exists a k_0 such that for all $k \geqq k_0$ (by the lower semicontinuity of the matrix rank) $\operatorname{rk}(D_k) = p$ and (by Lemma 1.2) $r_{(k)} = r$. That is to say, $C_q(G_k) = C(G)$ and thus C_q is continuous at G.

If C_q is continuous at G, since $m > p$, by Corollary 3.4, G has only column minimal indices and infinite elementary divisors with exponent one. If G has less than p infinite elementary divisors with exponent one, i.e., if $\operatorname{rk}(D) < p$ it is easy to find a sequence $G_k \to G$ such that $\operatorname{rk}(D_k) = p$. Thus, $\|C_q(G_k) - C_q(G)\| \geqq 1$ and C_q is not continuous at G.

Suppose now that G has p infinite elementary divisors with exponent one but its column minimal indices are not those enunciated. If G is in canonical form we have

$$G = \left[\begin{array}{c|c|c} A_\varepsilon & B_\varepsilon & 0 \\ \hline 0 & 0 & D_q \end{array}\right] \in \mathbb{C}^{(n+p)\times(n+m)},$$

where $D_q = I_p$ and $[A_\varepsilon, B_\varepsilon] \in \mathbb{C}^{n\times(n+m-p)}$ is a completely controllable pair with a partition of r-numbers that is not maximal for the majorization. Let r' be the conjugate partition of

$$(c+1, \overset{(d)}{\cdots}, c+1, c, \overset{(m-p-d)}{\cdots\cdots}, c, 0, \cdots), \quad \text{i.e.,}$$

$r'_1 = \cdots = r'_c = m - p$ and $r'_{c+1} = d$. Then $r \prec r'$ and by Theorem 5.3 of [5] there exists, in any neighborhood of $[A_\varepsilon, B_\varepsilon]$, a matrix $[A', B']$ such that r' is the partition of r-numbers of $[A', B']$. Thus, there exists, in any neighborhood of G, a matrix

$$G' = \left[\begin{array}{c|c|c} A' & B' & 0 \\ \hline 0 & 0 & I_p \end{array}\right]$$

such that r' is the partition of r-numbers of G', so $\|C_Q(G') - C_q(G)\| \geqq 1$ and C_q is not continuous at G. □

THEOREM 3.6. *Let* $C_q : \mathbb{C}^{(n+p)\times(n+m)} \to \mathbb{C}^{(n+p)\times(n+m)}$ *with* $p > m$. *Let c and d be nonnegative integers such that* $n = (p-m)c + d$. *Then* C_q *is continuous at* $G \in \mathbb{C}^{(n+p)\times(n+m)}$ *if and only if G has as only invariants m infinite elementary divisors with exponent one and the following row minimal indices*:

$$\eta_1 = \cdots = \eta_d = c+1, \qquad \eta_{d+1} = \cdots = \eta_{p-m} = c.$$

Remark. This last condition is equivalent to

$$s_1 = \cdots = s_c = p-m \quad \text{and} \quad s_{c+1} = d.$$

Proof. It suffices to deduce from Theorem 3.5 the points of continuity of the map C_q defined and with values in $C^{(n+m)\times(n+p)}$ with $p > m$. The transpose quadruples of those obtained in this way are the quadruples we are looking for. □

THEOREM 3.7. *Let* $C_q : \mathbb{C}^{(n+m)\times(n+m)} \to \mathbb{C}^{(n+m)\times(n+m)}$; C_q *is continuous at* $G \in \mathbb{C}^{(n+m)\times(n+m)}$ *if and only if* G *has as only invariants* m *infinite elementary divisors with exponent one and* n *finite elementary divisors of the form*

$$\lambda - \lambda_1, \cdots, \lambda - \lambda_n \quad \text{with } \operatorname{Re}(\lambda_i) \neq \operatorname{Re}(\lambda_j) \quad \text{for all } i \neq j.$$

Proof. If G has the indicated invariants we have that $\operatorname{rkn}(G + \lambda E) = n + m$ and $\operatorname{rk}(D) = m$ and both of them are maximum. Thus we can apply Lemma 1.2.

Given $G_k \to G$, by the lower semicontinuity of the matrix rank, we have that for all k sufficiently large $\operatorname{rk}(D_k) = m$, i.e., G_k has m infinite elementary divisors with exponent one. Now we make a reasoning analogous to that of the first part of Theorem 2.9, for the finite elementary divisors. So we have that if ε is any positive real number, for all k sufficiently large, G_k has exactly n finite eigenvalues that are simple and sufficiently close to the corresponding eigenvalue of G so as to have $\|C_q(G_k) - C_q(G)\| < \varepsilon$, i.e., $C_q(G_k) \to C_q(G)$. Therefore C_q is continuous at G.

If C_q is continuous at G by Corollary 3.4, G has only finite elementary divisors and infinite elementary divisors with exponent one. And G must have m infinite elementary divisors with exponent one because if G has less than m then D is not a full rank matrix and we can find a sequence $G_k \to G$ such that the matrices D_k are full rank and C_q will not be continuous at G.

Finally, suppose that G has m infinite elementary divisors with exponent one but G does not have the indicated finite elementary divisors. Let us consider that G is in canonical form, i.e.,

$$G = \left[\begin{array}{c|c} A_f & 0 \\ \hline 0 & I_m \end{array}\right] \in \mathbb{C}^{(n+m)\times(n+m)}.$$

From what we have supposed and by Theorem 2 of [4] we deduce that the Jordan canonical form is not continuous at A_f. Thus there exists an $\varepsilon > 0$ such that we can obtain A'_f, as close to A_f as we want, with a Jordan canonical form that is at a distance greater than the positive real number ε, from A_f. Therefore, C_q is not continuous at G. □

4. Points of continuity of the canonical form for the equivalence of matrix triples. In this last section we will study a particular case, which is that of *matrix triples* of the form (A, B, C) with $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, and $C \in \mathbb{C}^{p\times n}$. In fact, they consist of matrix quadruples such that $D = 0 \in \mathbb{C}^{p\times m}$ and we cannot change D in the definition of the equivalence relation of matrix triples when looking for matrix triples close to a given one. So we can apply to matrix triples the definitions and results obtained for quadruples, having in mind the restriction we have just mentioned.

DEFINITION. Two matrix triples (A_1, B_1, C_1) and (A_2, B_2, C_2) are said to be *equivalent* if there exist matrices $P \in GL_n(\mathbb{C})$, $Q \in GL_p(\mathbb{C})$, $T \in GL_m(\mathbb{C})$, $R \in \mathbb{C}^{n\times p}$, and $S \in \mathbb{C}^{m\times n}$ such that

$$\begin{bmatrix} P & R \\ 0 & Q \end{bmatrix}\begin{bmatrix} A_1 & B_1 \\ C_1 & 0 \end{bmatrix}\begin{bmatrix} P^{-1} & 0 \\ S & T \end{bmatrix} = \begin{bmatrix} A_2 & B_2 \\ C_2 & 0 \end{bmatrix}.$$

As a consequence of this definition and making proofs analogous to those of Propositions 3.1 and 3.2, we have the following results.

PROPOSITION 4.1. *(A_1, B_1, C_1) and (A_2, B_2, C_2) are equivalent if and only if (A_2, B_2, C_2) can be obtained from (A_1, B_1, C_1) by means of one, or more, of the following elementary transformations*:

(1) $(A_1, B_1, C_1) \to (A_2, B_2, C_2) = (PA_1P^{-1}, PB_1, C_1P^{-1})$,
(2) $(A_1, B_1, C_1) \to (A_2, B_2, C_2) = (A_1 + RC_1, B_1, C_1)$,
(3) $(A_1, B_1, C_1) \to (A_2, B_2, C_2) = (A_1, B_1, QC_1)$,
(4) $(A_1, B_1, C_1) \to (A_2, B_2, C_2) = (A_1 + B_1S, B_1, C_1)$,
(5) $(A_1, B_1, C_1) \to (A_2, B_2, C_2) = (A_1, B_1T, C_1)$

where P, Q, T, R, and S are matrices as those considered in the previous definition.

Notation. Let $(A_i, B_i, C_i) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times m} \times \mathbb{C}^{p\times n}$. We define

$$E := \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}, \qquad G_i := \begin{bmatrix} A_i & B_i \\ C_i & 0 \end{bmatrix}.$$

Thus E and G_i are complex matrices of size $(n + p) \times (n + m)$, $i = 1, 2$.

PROPOSITION 4.2. *Two triples (A_1, B_1, C_1) and (A_2, B_2, C_2) are equivalent if and only if the pencils $G_1 + \lambda E$ and $G_2 + \lambda E$ are strictly equivalent.*

This equivalence relation for matrix triples is a generalization of the Γ-equivalence for matrix pairs, and a particular case of the equivalence of quadruples and therefore a particular case of the strict equivalence of matrix pencils.

Let $H(\lambda) = H_1 + \lambda H_2$ be a matrix pencil of size $(n + p) \times (n + m)$ such that $\operatorname{rk}(H_2) = n$ and $H(\lambda)$ has no infinite elementary divisors with exponent one; then there exist nonsingular matrices P and Q such that $PH(\lambda)Q = G + \lambda E$ where

$$G = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}$$

and $A \in \mathbb{C}^{n\times n}$. In this case we will say that the *triple* (A, B, C) is *associated* with the pencil $H(\lambda)$.

It is also true that two triples associated with the same pencil are equivalent and that two triples, associated with different pencils, are equivalent if and only if the pencils, with which they are associated, are strictly equivalent.

As we did for quadruples, we will say that (A, B, C) or

$$G = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}$$

is a *triple*. We will call *column minimal indices*, *r-numbers*, *row minimal indices*, *s-numbers*, *Segre* and *Weyr characteristics* of G for the *eigenvalue infinite*, the column minimal indices, r-numbers, etc., of the pencil $G + \lambda E$. The *Segre* and *Weyr characteristics* of G for an *eigenvalue* $\alpha \in \mathbb{C}$ will be the Segre and Weyr characteristics of $G + \lambda E$ for the eigenvalue $-\alpha$.

Since $D = 0$, a triple will not have infinite elementary divisors with exponent one, as we have noted for pencils that have associated triples.

For the invariants of triples we will employ the same notation as for quadruples.

Let us consider

$$D_{n,m,p} := \left\{ \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+m)} \right\}$$

as a metric subspace of $\mathbb{C}^{(n+p)\times(n+m)}$, i.e., the matrix norm is that considered in § 3. Let

$$C_t : D_{n,m,p} \to D_{n,m,p}$$

be the map that associates with each matrix

$$G=\begin{bmatrix} A & B \\ C & 0 \end{bmatrix}$$

its canonical form for the equivalence of triples

$$\begin{bmatrix} A_t & B_t \\ C_t & 0 \end{bmatrix}$$

where

$$A_t := \operatorname{diag}(A_\varepsilon, A_\eta, A_\infty, A_f) \in \mathbb{C}^{n\times n},$$

$$B_t := \begin{bmatrix} B_\varepsilon & 0 \\ 0 & 0 \\ 0 & B_\infty \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{n\times n}, \quad \text{and} \quad C_t := \begin{bmatrix} 0 & C_\eta & 0 & 0 \\ 0 & 0 & C_\infty & 0 \end{bmatrix} \in \mathbb{C}^{p\times n},$$

according to the notation and definitions of § 3. That is to say, $(A_\varepsilon, B_\varepsilon)$ is a completely controllable pair in Brunovsky canonical form and its controllability indices are the column minimal indices of G; (A_η^T, C_η^T) is a completely controllable pair in Brunovsky canonical form and its controllability indices are the row minimal indices of G;

$$\begin{bmatrix} A_\infty & B_\infty \\ C_\infty & 0 \end{bmatrix}$$

is in canonical form corresponding to the infinite elementary divisors of G (which are of exponent greater than or equal to two); finally, A_f is a matrix in Jordan canonical form associated with the finite eigenvalues of G ordered according to the lexicographic order in $\mathbb{C}$, and taking the blocks, corresponding to each eigenvalue, in decreasing order of size.

Remarks. (1) Since in the elementary transformations (1), (3), and (5) the changes are due to nonsingular matrices we have that rk (B) and rk (C) are two invariants of the triple (A, B, C) for the equivalence.

(2) As there is no infinite elementary divisor of exponent one, we have that $m_{\infty 1} = m_{\infty 2}$, i.e., $t_0 = t_1$. Thus we can write

(i) rkn $(G + \lambda E) = n + t_1$;

(ii) $m = r_0 + t_1$ and $p = s_0 + t_1$.

The invariants and a canonical form for the equivalence of triples can also be obtained by the procedure followed in [8].

As it happened for quadruples, we can apply the results about perturbation of pencils, which give necessary conditions (Lemmas 1.1 and 1.2).

It remains true that C_t is continuous at a triple if and only if C_t is continuous at any equivalent triple.

LEMMA 4.3. *Let*

$$G=\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \in D_{n,m,p}.$$

Suppose that G satisfies at least one of the following properties:

(i) *G has infinite elementary divisors with exponent greater than two*;

(ii) *G has finite elementary divisors and column minimal indices*;

(iii) *G has finite elementary divisors and row minimal indices*;

(iv) *G has column and row minimal indices different from zero*;

(v) *G has column minimal indices equal to zero and* $\operatorname{rk}(B) < \min\{n, m\}$;

(vi) *G has row minimal indices equal to zero and* $\operatorname{rk}(C) < \min\{p, n\}$.

Then C_t *is not continuous at* G.

Proof. Suppose that G verifies (i) and that G is in canonical form. We will study the case where G has only one elementary divisor μ^3, and if G has infinite elementary divisors with exponent greater than three, then the procedure is the same:

$$G = \left[\begin{array}{cc|c} 0 & 0 & 1 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \end{array}\right] \in D_{2,1,1}.$$

Let

$$G_k := \left[\begin{array}{cc|c} 0 & 0 & 1 \\ 1 & 0 & \frac{1}{k} \\ \hline 0 & 1 & 0 \end{array}\right] \in D_{2,1,1}.$$

Then

$$C_t(G_k) = \left[\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & -k & 0 \\ \hline 1 & 0 & 0 \end{array}\right],$$

which is at a distance greater than three from $C_t(G) = G$. Thus C_t is not continuous at G.

If G verifies (ii), (iii), or (iv) we make a proof analogous to that of Lemma 3.3 in cases (ii), (iii), and (iv), respectively.

If (v) holds for G, in canonical form, it is enough to perturb some of the null columns of B, associated with the column minimal indices of G equal to zero, so as to obtain a matrix B' such that $\operatorname{rk}(B) < \operatorname{rk}(B')$. Thus we can find in any neighborhood of G a matrix G' such that $\|C_t(G') - C_t(G)\| \geqq 1$, i.e., C_t is not continuous at G.

In the case where (vi) holds, we proceed as we have just said by perturbing the matrix C of G instead of B. □

COROLLARY 4.4. *If* C_t *is continuous at* G, *then one of the following properties holds*:

(i) *G has only column minimal indices and infinite elementary divisors with exponent two*;

(ii) *G has only row minimal indices and infinite elementary divisors with exponent two*;

(iii) *G has only finite elementary divisors and infinite elementary divisors with exponent two*;

(iv) *G has only infinite elementary divisors with exponent two*;

(v) *G has only column and row minimal indices equal to zero and infinite elementary divisors of exponent two*.

Each one of these properties corresponds to a different relation between n, m, and p, as follows:

(i) $m > p$ and $n \geqq p$,

(ii) $p > m$ and $n \geqq m$,

(iii) $m = p$ and $n > m$,

(iv) $n = m = p$, and

(v) $n < \min\{p, m\}$.

Notation. $a = bc + d$ will have a meaning when $a = 0$ and $b > 0$ by taking $c = d = 0$.

THEOREM 4.5. *Let $C_t : D_{n,m,p} \to D_{n,m,p}$ with $m > p$ and $n \geqq p$. Let c and d be nonnegative integers such that $n - p = (m - p)c + d$. Then C_t is continuous at $G \in D_{n,m,p}$ if and only if G has as only invariants p infinite elementary divisors with exponent two and the following column minimal indices*:

$$\varepsilon_1 = \cdots = \varepsilon_d = c + 1, \qquad \varepsilon_{d+1} = \cdots = \varepsilon_{m-p} = c.$$

Remarks. This means that if $n = p$ the column minimal indices are $\varepsilon_1 = \cdots = \varepsilon_{m-p} = 0$, i.e., $r_1 = 0$. In other cases $r_1 = \cdots = r_c = m - p$ and $r_{c+1} = d$.

Proof. If G has the mentioned invariants, then rkn $(G + \lambda E)$ is maximum and we can apply Lemma 1.2. Moreover, C is a full rank matrix. So if $G_k \to G$, by the lower semicontinuity of the matrix rank, for all k sufficiently large we have that C_k is also full rank, i.e., G_k has the same infinite elementary divisors as G. By Lemma 1.2, they also have the same column minimal indices as G, for all sufficiently large k. Thus C_t is continuous at G.

If C_t is continuous at G, as $m > p$ and $n \geqq p$, by Corollary 4.4, G has only column minimal indices and infinite elementary divisors with exponent two. And G must have p infinite elementary divisors with exponent two because if G has less than p then C is not a full rank matrix and we can find a sequence $G_k \to G$ such that the matrices C_k are full rank and C_q will not be continuous at G.

If G has p infinite elementary divisors with exponent two but its column minimal indices are not those indicated, we take the submatrix $[A_\varepsilon, B_\varepsilon]$ of $C_t(G)$ and we apply Theorem 5.3 of [5] as we did in the last part of the proof of Theorem 3.5. □

THEOREM 4.6. *Let $C_t : D_{n,m,p} \to D_{n,m,p}$ with $p > m$ and $n \geqq m$. Let c and d be nonnegative integers such that $n - m = (p - m)c + d$. Then C_t is continuous at $G \in D_{n,m,p}$ if and only if G has as only invariants m infinite elementary divisors with exponent two and the following row minimal indices*:

$$\eta_1 = \cdots = \eta_d = c + 1, \qquad \eta_{d+1} = \cdots = \eta_{p-m} = c.$$

Remark. As in Theorem 4.5 if $n = m$ the row minimal indices are $\eta_1 = \cdots = \eta_{p-m} = 0$ (i.e., $r_1 = 0$). In other case $s_1 = \cdots = s_c = p - m$ and $s_{c+1} = d$.

Proof. As in other sections this case is solved by means of Theorem 4.5 if we consider the map $C_t : D_{n,p,m} \to D_{n,p,m}$ with $p > m$ and $n \geqq m$. □

THEOREM 4.7. *Let $C_t : D_{n,m,m} \to D_{n,m,m}$ with $n \geqq m$. Then C_t is continuous at $G \in D_{n,m,m}$ if and only if G has as only invariants m infinite elementary divisors with exponent two and $n - m$ finite elementary divisors of the form*

$$\lambda - \lambda_1, \cdots, \lambda - \lambda_{n-m} \quad \text{with } \operatorname{Re}(\lambda_i) \neq \operatorname{Re}(\lambda_j) \quad \text{for all } i \neq j.$$

Remark. If $n = m$ there are only m infinite elementary divisors of exponent two as we have anticipated in Corollary 4.4 and in the paragraph which follows it.

Proof. If G is as indicated, then rk (C) = rk $(B) = m$, i.e., C and B are full rank matrices. If $G_k \to G$, by the lower semicontinuity of the matrix rank, we have that C_k and B_k are full rank matrices for all sufficiently large k, i.e., G_k has m infinite elementary divisors with exponent two.

If $n = m$ we have proved that C_t is continuous at G. If $n > m$ we apply Lemma 1.2 (it is possible to do so because rkn $(G + \lambda E) = n + m$) as we did in Theorem 2.9 (see

also the proof of Theorem 3.7). So we obtain that $C_t(G_k) \to C_t(G)$ and thus C_t is continuous at G.

If C_t is continuous at G we deduce from Corollary 4.4 that G has only invariants of the indicated types. Moreover, G must have m infinite elementary divisors with exponent two because if G has less than m then C and B are not full rank matrices and C_t will not be continuous at G.

If $n = m$ we have proved the theorem. If $n > m$ and G has the indicated infinite invariant factors but G does not have the indicated finite invariant factors, it suffices to consider the submatrix A_f of G, in canonical form, and to apply Theorem 2 of [4] as we do in the proof of Theorem 3.7. Therefore, C_t is not continuous at G. □

THEOREM 4.8. *Let $C_t : D_{n,m,p} \to D_{n,m,p}$ with $n < \min\{p, m\}$. Then C_t is continuous at G if and only if G has as invariants n infinite elementary divisors with exponent two, $m - n$ column minimal indices equal to zero, and $p - n$ row minimal indices equal to zero.*

Proof. If G has the indicated form we have that B and C are full rank matrices. By the lower semicontinuity of the matrix rank, if $G_k \to G$ then for all sufficiently large k matrices B_k and C_k are also full rank, i.e., $C_t(G_k) = G$ for all sufficiently large k. Thus C_t is continuous at G.

Conversely, if C_t is continuous at G, by Corollary 4.4, G has invariants of the three enunciated types. If G has less than n infinite elementary divisors with exponent two it will be $\operatorname{rk}(B) < n$ and $\operatorname{rk}(C) < n$ and C_t will not be continuous at G. So G has exactly n infinite elementary divisors with exponent two, and then it is easy to deduce that the other invariants must be as indicated. □

5. Conclusion. The characterization of the continuity points of each one of the three canonical forms studied here depends on the relations among the sizes of the different matrices concerning each case. When these relations together with the continuity at one point (matrix pencil, quadruple, triple) allow a Jordan part, the eigenvalues must be simple and with different real parts. When there is no possibility of continuity at points with Jordan part, the invariants allowed are those corresponding to an equivalence class.

Acknowledgments. The author is grateful to Professor J. M. Gracia for suggesting this theme and for his guidance, and Professor I. Zaballa for many helpful discussions.

REFERENCES

[1] H. DEN BOER AND G. PH. A. THIJSSE, *Semi-stability of sums of partial multiplicities under additive perturbation*, Integral Equations Operator Theory, 3/1 (1980), pp. 23–42.

[2] F. R. GANTMACHER, *Théorie des matrices*, tome 2, Dunod, Paris, 1966.

[3] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Invariant Subspaces of Matrices with Applications*, John Wiley, New York, 1986.

[4] J. M. GRACIA AND I. DE HOYOS, *Puntos de continuidad de formas canónicas de matrices*, the Homage Book to Prof. Luis de Albuquerque of Coimbra, Coimbra, 1987.

[5] J. M. GRACIA, I. DE HOYOS, AND I. ZABALLA, *Perturbation of linear control systems*, Linear Algebra Appl., 121 (1989), pp. 353–383.

[6] A. S. MARKUS AND E. E. PARILIS, *The change of the Jordan structure of a matrix under small perturbations*, Linear Algebra Appl., 54 (1983), pp. 139–152.

[7] B. P. MOLINARI, *Structural invariants of linear multivariable systems*, Internat. J. Control, 28 (1978), pp. 493–510.

[8] A. S. MORSE, *Structural invariants of linear multivariable systems*, SIAM J. Control, 11 (1973), pp. 446–465.

[9] A. POKRZYWA, *On perturbations and the equivalence orbit of a matrix pencil*, Linear Algebra Appl., 82 (1986), pp. 99–121.

[10] J. S. THORP, *The singular pencil of a dynamical system*, Internat. J. Control, 18 (1973), pp. 577–596.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 301–311, April 1990

010

ON RUTISHAUSER'S APPROACH TO SELF-SIMILAR FLOWS *

D. S. WATKINS † AND L. ELSNER ‡

Abstract. Certain variants of the Toda flow are continuous analogues of the QR algorithm and other algorithms for calculating eigenvalues of matrices. This was a remarkable discovery of the early eighties. Until very recently contemporary researchers studying this circle of ideas have been unaware that continuous analogues of the quotient-difference and LR algorithms were already known to Rutishauser in the fifties. Rutishauser's continuous analogue of the quotient-difference algorithm contains the finite, nonperiodic Toda flow as a special case. A nice feature of Rutishauser's approach is that it leads from the (discrete) eigenvalue algorithm to the (continuous) flow by a limiting process. Thus the connection between the algorithm and the flow does not come as a surprise. In this paper it is shown how Rutishauser's approach can be generalized to yield large families of flows in a natural manner. The flows derived include continuous analogues of the LR, QR, SR, and HR algorithms.

Key words. Toda flow, self-similar flow, quotient-difference algorithm, LR algorithm, QR algorithm

AMS(MOS) subject classifications. 15A18, 15A23, 58F19, 58F25, 65F15

C.R. classification. G.1.3

1. The Toda flow and the quotient-difference algorithm.

In recent years there has been considerable interest in flows that are continuous analogues of the QR algorithm and other algorithms for calculating the eigenvalues of a matrix [2], [16], [18]. The present interest dates from Toda's study [17] of a dynamical system that came to be known as the Toda lattice. This is a system of infinitely many points of unit mass constrained to lie on a line, such that each point exerts an exponential repelling force on its two nearest neighbors. If the ith point has position q_i and momentum p_i, then

$$\dot{q}_i = p_i, \qquad \dot{p}_i = \exp(q_{i-1} - q_i) - \exp(q_i - q_{i+1}). \tag{1}$$

In addition Toda applied a periodicity condition $q_{n+i} = q_i + 2\pi l$, for all i. Here n and l are fixed positive numbers, n an integer. Toda's work was published in 1970. Subsequently many workers in dynamical system theory studied the Toda flow and numerous variants and generalizations. See, for example, [3], [5], [7], [8], [15], and the works cited above. (Additional works are cited in the bibliography of [19].) We will focus on a few of these. Moser [8] studied a variant with finitely many points and no periodicity condition. This finite, nonperiodic Toda lattice satisfies (1) for $i = 1, \cdots, n$ with $q_0 = -\infty$ and $q_{n+1} = \infty$. We will restrict our attention to this version of the Toda lattice. Flaschka [3] noticed that the change of variables

$$a_i = -\tfrac{1}{2}p_i, \qquad b_i = \tfrac{1}{2}\exp(\tfrac{1}{2}(q_i - q_{i+1}))$$

leads to the system

$$\dot{a}_i = 2(b_i^2 - b_{i-1}^2), \qquad i = 1, \cdots, n,$$

* Received by the editors November 14, 1988; accepted for publication (in revised form) June 1, 1989.

† Department of Pure and Applied Mathematics, Washington State University, Pullman, Washington 99164-2930 (watkins@wsumath.bitnet).

‡ Fakultät für Mathematik, Universität Bielefeld, Postfach 8640, D-4800 Bielefeld 1, Federal Republic of Germany (umatf105@dbuni11.bitnet).

$$\dot{b}_i = b_i(a_{i+1} - a_i), \qquad i = 1, \cdots, n-1, \tag{2}$$

$$b_0 = b_n = 0,$$

which can be expressed as a matrix differential equation

$$\dot{B} = B\rho(B) - \rho(B)B, \tag{3}$$

where B and $\rho(B)$ are the tridiagonal matrices

$$B = \begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & & & \\ & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & b_{n-1} \\ & & & b_{n-1} & a_n \end{bmatrix}, \qquad \rho(B) = \begin{bmatrix} 0 & -b_1 & & & \\ b_1 & 0 & & & \\ & \ddots & \ddots & \ddots & \\ & & & 0 & -b_{n-1} \\ & & & b_{n-1} & 0 \end{bmatrix}.$$

Note that B is symmetric and $\rho(B)$ is skew-symmetric. Given any symmetric, tridiagonal initial matrix $\hat{B}$, let $B(t)$ be the unique solution of (3) satisfying $B(0) = \hat{B}$. Then it is not hard to show that $B(t)$ is orthogonally similar to $\hat{B}$ for all t. Hence we say that the flow is *self-similar*. It is also called *isospectral* because the eigenvalues of $B(t)$ are invariant. Since the points of the lattice repel one another, we must eventually have $q_i - q_{i+1} \to -\infty$. Thus $b_i \to 0$ for $i = 1, \cdots, n-1$, and the a_i converge to the eigenvalues of $\hat{B}$. In a paper published in 1982, Symes [16] made the remarkable observation that the finite, nonperiodic Toda flow is a continuous analogue of the QR algorithm [22] for calculating the eigenvalues of a matrix. Starting from some initial matrix A_0, the QR algorithm produces a sequence (A_k) of matrices similar to A_0. Symes showed that the unshifted QR algorithm with initial matrix $A_0 = \exp(\hat{B})$ produces the sequence $(A_k) = (\exp(B(k)))$. This observation was generalized in various directions. Deift, Nanda, and Tomei [2] considered more general flows of the form

$$\dot{B} = B\rho(f(B)) - \rho(f(B))B$$

for suitable functions f. For a fixed f, the more general flow produces $B(t)$ such that the QR algorithm with starting matrix $A_0 = \exp(f(\hat{B}))$ produces the sequence

$$(A_k) = (\exp\{f(B(k))\}).$$

In particular, the choice $f(x) = \log x$ yields a flow for which $B(0), B(1), B(2), \cdots$ is exactly the sequence produced by the unshifted QR algorithm with starting matrix $A_0 = \hat{B}$. In other words, this flow interpolates the QR algorithm. Chu [1] extended the family of flows to include nonsymmetric, nontridiagonal B. We refer to this family of flows collectively as *QR flows*. In [18] Watkins introduced a family of LR flows (called LU flows in [18]) that are related to the unshifted LR algorithm [22] in exactly the same way.

All of this work was published after 1970, and all of it was done in ignorance of earlier work of Rutishauser [11], [14]. It is well known that Rutishauser invented the LR algorithm in the fifties [13], [14]. In 1958, in one of his early papers on the subject [14], he included a section entitled "A continuous analogue to the LR transformation," in which he developed the LR analogue of the Toda flow. It turns out that Rutishauser's flow is a member of the family of LR flows introduced by Watkins [18] much later.

A pleasing feature of Rutishauser's derivation is that it proceeds from the LR algorithm to the flow in a natural way, i.e., by taking a limit. Thus the connection does not come as a surprise, as it did in the case of Symes's discovery of the connection between the Toda flow and the QR algorithm. One might well wonder what led Rutishauser to this natural approach. The answer lies in the historical roots of the LR algorithm. The LR algorithm evolved from the quotient-difference (q-d) algorithm, which was also developed by Rutishauser [9], [10], [12]. The q-d algorithm started out as a method for finding the poles of a meromorphic function. For almost all choices of $x, y \in \mathbb{C}^n$, the function $f(\lambda) = y^T(\lambda I - A)^{-1}x$ has the eigenvalues of A as its poles, so the algorithm can also be used to find the eigenvalues of a matrix. As it was originally formulated, the q-d algorithm consisted of filling out a so-called q-d table, which resembles a table of differences, except that the rules for forming a q-d table are more complicated. For details see the original work of Rutishauser or Henrici's book [6]. The zeroth column of an ordinary difference table consists of the values of a smooth function at equally spaced points. As the spacing tends to zero, the first and higher order differences tend to zero as well. However, if the table is modified so that it contains divided differences instead of simple differences, the column of kth differences will tend to the kth derivative as the spacing tends to zero. Notice that if we let $g_k(t)$ denote the limit of the kth column, then $\dot{g}_k = g_{k+1}$ for $k = 0, 1, 2, \cdots$. Thus the columns are related by a simple system of differential equations. The entries in the zeroth column of a q-d table can also be viewed as values of a certain function at equally spaced points. It is therefore quite natural to ask what happens as the spacing converges to zero. It turns out that the limit is not very interesting. Certain columns (the quotients) tend to 1, while others (the differences) tend to zero. However, we would hope to be able to modify the table in the spirit of divided differences, so that an interesting limit is obtained. This turns out to be possible, but since the formation rules for a q-d table are more complicated than for a simple difference table, the columns (of the modified table) do not converge to simple derivatives of the original function. Instead, the limit satisfies a more complicated system of differential equations:

$$\begin{aligned} \dot{Q}_i &= E_i - E_{i-1}, \qquad i = 1, \cdots, n, \\ \dot{E}_i &= E_i(Q_{i+1} - Q_i), \qquad i = 1, \cdots, n-1, \\ E_0 &= 0 = E_n. \end{aligned} \tag{4}$$

$Q_i(t)$ is the limit of the ith column of (modified) quotients and $E_i(t)$ is the limit of the ith column of (modified) differences. This continuous analogue of the q-d algorithm was published by Rutishauser [11] in 1954. The equations (4) resemble Flaschka's form (2) of the finite, nonperiodic Toda equations. In fact, the change of variables

$$Q_i = 2a_i, \qquad E_i = 4b_i^2$$

transforms (4) into (2). Thus Rutishauser published a form of the Toda flow 16 years before Toda. The system (4) is actually more general than the Toda flow, since the Toda flow corresponds to the special case $E_i > 0$, $i = 1, \cdots, n-1$.

The original formulation of the quotient-difference algorithm was found to be unstable. A better approach is to fill in the q-d table from top to bottom, rather than from left to right. Rutishauser quickly recognized that the top-to-bottom procedure could be interpreted as a process of matrix factorization and recombination, and the LR algorithm was born. The q-d algorithm is just the LR algorithm applied to a tridiagonal matrix with 1's on the superdiagonal. Once the algorithm assumed

this new guise, it became easy to forget the q-d table and its infinitesimal limit. But Rutishauser did not forget. Generalizing from the q-d algorithm, he obtained a continuous analogue of the LR algorithm [14], which he published in 1958.

Given that the Toda flow is a continuous analogue of the QR algorithm, whereas Rutishauser's flow (4) is associated with the LR algorithm, it might seem surprising that (4) should include the Toda flow as a special case. Actually, this need not be such a surprise. Suppose the LR algorithm, or, equivalently, the q-d algorithm, is applied to a symmetric, positive definite, tridiagonal matrix. The symmetry is not preserved by the algorithm, but a trivial rescaling transforms the LR algorithm to the Cholesky LR algorithm [22], which does preserve symmetry. The outputs of the two algorithms differ by diagonal similarity transformations, so we can think of them as the same, at least in principle. It is well known [22] that two steps of the Cholesky LR algorithm are equivalent to one step of the (symmetric, unshifted) QR algorithm. Thus, in a sense, the q-d algorithm includes as a special case the QR algorithm for symmetric, positive definite, tridiagonal matrices. The same must be true of the continuous analogues.

In the remainder of the paper we will show how to construct flows by Rutishauser's method. Our construction will be based on Rutishauser's LR flow, not the q-d flow; the former is more general than the latter. We will present a generalization of Rutishauser's construction that produces QR, SR, HR, and other flows as well. We begin by introducing a generic eigenvalue algorithm, the *FG algorithm.* We then derive a continuous analogue, a generic *FG flow.* In §3 the construction is generalized to yield a whole family of FG flows associated with each FG algorithm. This is exactly the family of autonomous FG flows discussed in [19]. The contribution of the present paper is not to develop new flows, but to show how Rutishauser's construction can be generalized to produce known flows in a very natural manner. An additional contribution is that our development is rigorous. By contrast, Rutishauser's development was sketchy and omitted numerous details.

The approach developed here can also be used to derive families of flows associated with algorithms for the generalized eigenvalue problem $\hat{A}x = \lambda\hat{B}x$. These are exactly the autonomous FGZ flows of [20]. The same approach can also be used to derive the autonomous flows associated with the singular value decomposition discussed in [21]. The constructions are straightforward, and we omit them.

2. Construction of flows by Rutishauser's approach. In order to achieve the desired level of generality, we will make use of some notions from elementary Lie theory. The reader who would rather not learn about Lie theory at this time should skim the next two paragraphs lightly, then have a close look at Examples 2.1L and 2.1Q. The reader can then read the rest of the paper easily by substituting either QR or LR for FG.

Let $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, and let $GL_n(\mathbb{F})$ denote the general linear group of nonsingular $n \times n$ matrices over $\mathbb{F}$. Given a closed subgroup $\mathcal{G}$ of $GL_n(\mathbb{F})$, let $\Lambda(\mathcal{G}) \subset \mathbb{F}^{n\times n}$ denote the Lie algebra associated with $\mathcal{G}$. The basic facts about Lie algebras of matrices are stated in [19]. For more complete information about Lie groups and algebras see [4], for example. The Lie algebra $\Lambda(\mathcal{G})$ is most easily viewed as the tangent space of the manifold $\mathcal{G}$ at the identity element. Thus it can be thought of as a subspace of $\mathbb{F}^{n\times n}$. Let $\mathcal{F}$ and $\mathcal{G}$ be two closed subgroups of $GL_n(\mathbb{F})$ such that

$$\Lambda(\mathcal{F}) \oplus \Lambda(\mathcal{G}) = \mathbb{F}^{n\times n}, \tag{5}$$

and $\Lambda(\mathcal{G})$ contains the identity matrix. This last assumption implies that $\Lambda(\mathcal{G})$ con-

tains the Lie algebra of all real multiples of I, which is equivalent to the condition that $\mathcal{G}$ contains the Lie group of all positive multiples of I. We could equally well require that $\Lambda(\mathcal{F})$, rather than $\Lambda(\mathcal{G})$, contain the identity matrix. However, as we shall later see, neither of these assumptions is really necessary. The assumption (5) means that every $X \in \mathbb{F}^{n\times n}$ can be decomposed in exactly one way as

$$X = \rho(X) + \sigma(X), \qquad \rho(X) \in \Lambda(\mathcal{F}),\ \sigma(X) \in \Lambda(\mathcal{G}). \tag{6}$$

This equation defines linear transformations ρ and σ, which are complementary projectors of $\mathbb{F}^{n\times n}$ onto $\Lambda(\mathcal{F})$ and $\Lambda(\mathcal{G})$, respectively. The existence of the additive decomposition (5) implies the existence of a multiplicative decomposition: There is a neighborhood $\mathcal{V}$ of I in $GL_n(\mathbb{F})$ such that every $A \in \mathcal{V}$ has a unique *FG decomposition*; that is, A can be expressed uniquely as a product $A = FG$, where $F \in \mathcal{F}$ and $G \in \mathcal{G}$ [4], [19].

Let $A(\epsilon)$ be an analytic function of ϵ with $A(0) = I$. Then for sufficiently small ϵ, $A(\epsilon)$ has an FG decomposition $A(\epsilon) = F(\epsilon)G(\epsilon)$, and the factors $F(\epsilon) \in \mathcal{F}$ and $G(\epsilon) \in \mathcal{G}$ are also analytic functions satisfying $F(0) = G(0) = I$. Expanding each in a Taylor series we have

$$F(\epsilon) = I + \epsilon X + \epsilon^2 M + O(\epsilon^3), \qquad G(\epsilon) = I + \epsilon Y + \epsilon^2 N + O(\epsilon^3),$$

where $X = F'(0) \in \Lambda(\mathcal{F})$ and $Y = G'(0) \in \Lambda(\mathcal{G})$. We will need to use expansions of this type to derive the FG flows.

Example 2.1L. Rutishauser considered the special case in which the FG decomposition is the LR decomposition. In this case $\mathcal{F}$ is the group of unit lower triangular matrices, and $\mathcal{G}$ is the group of nonsingular upper triangular matrices. ($\mathbb{F}$ can be either $\mathbb{R}$ or $\mathbb{C}$.) Thus $\Lambda(\mathcal{F})$ and $\Lambda(\mathcal{G})$ are the Lie algebras of strictly lower triangular and upper triangular matrices, respectively. Clearly (5) holds, and $\Lambda(\mathcal{G})$ contains I. Given $X \in \mathbb{F}^{n\times n}$, we obtain $\sigma(X)$ by setting the lower triangular entries of X to zero. Then $\rho(X) = X - \sigma(X)$.

Example 2.1Q. Let $\mathbb{F} = \mathbb{C}$. If we take $\mathcal{F}$ to be the unitary group and $\mathcal{G}$ the group of upper triangular matrices with real, positive, main diagonal entries, then the FG decomposition is just the QR decomposition. The Lie algebras $\Lambda(\mathcal{F})$ and $\Lambda(\mathcal{G})$ are just the skew-Hermitian matrices and the upper triangular matrices with real main diagonal entries, respectively. Obviously $\Lambda(\mathcal{G})$ contains I. It is easy to show that (5) holds. Every $X \in \mathbb{F}^{n\times n}$ can be expressed uniquely as a sum $X = L + D_r + D_i + U$, where L is strictly lower triangular, D_r is diagonal and real, D_i is diagonal and imaginary, and U is strictly upper triangular. We have $\rho(X) = L + D_i - L^*$ and $\sigma(X) = D_r + U + L^*$. There is also a real QR decomposition, which we obtain by taking $\mathcal{F}$ to be the group of real, orthogonal matrices and $\mathcal{G}$ the group of real, upper triangular matrices with positive entries on the main diagonal.

Two other examples, the SR and HR decompositions, are discussed in [19].

Associated with each FG decomposition is an FG *algorithm* for calculating eigenvalues of matrices. The *shifted* FG algorithm associated with $\mathcal{F}$ and $\mathcal{G}$ begins with a matrix $\hat{B} \in GL_n(\mathbb{F})$ and produces a sequence (B_k) by setting $B_0 = \hat{B}$, and then defining B_k, for $k = 1, 2, 3, \cdots$, by the equations

$$B_{k-1} - \sigma_k I = \bar{F}_k \bar{G}_k, \qquad \bar{G}_k \bar{F}_k + \sigma_k I = B_k, \tag{7}$$

where $\bar{F}_k \in \mathcal{F}$, $\bar{G}_k \in \mathcal{G}$, and the shift σ_k is chosen so that $B_{k-1} - \sigma_k I$ has an FG decomposition. The meaning of (7) is that a shift is subtracted from B_{k-1}, an FG

decomposition of the shifted matrix is performed, the factors of the decomposition are multiplied back together in reverse order, then the shift is added back on, giving B_k. It is easy to show that the B_k so produced are all similar to $\hat{B}$, so they have the same eigenvalues. Under certain conditions on $\hat{B}$, $\mathcal{F}$, $\mathcal{G}$, and (σ_k), the sequence (B_k) can be shown to converge to triangular or quasi-triangular form, yielding the eigenvalues of $\hat{B}$. The shifts are generally chosen with an eye to accelerating convergence. Rutishauser used shifts for a different purpose, namely, to pass to a continuous limit. Following Rutishauser we consider a constant shift $\sigma_k = -\mu$, where μ is positive and large. (We plan to take a limit in which $\mu \to \infty$.) Then the sequence (B_k) is generated by

$$B_{k-1} + \mu I = \bar{F}_k(\mu \bar{G}_k), \qquad (\mu \bar{G}_k)\bar{F}_k - \mu I = B_k. \tag{8}$$

We have factored the scalar μ out of $\bar{G}_k$ for convenience. Because of the assumption that $\mathcal{G}$ contains all positive multiples of the identity matrix, we have $\bar{G}_k \in \mathcal{G}$ if and only if $\mu \bar{G}_k \in \mathcal{G}$.

Notice that this choice of shifts actually slows convergence. This is so because the rate of convergence (when it occurs at all) is determined, at least in part, by ratios of eigenvalues of $\hat{B} + \mu I$. As μ is made larger, the ratios of the eigenvalues approach one, indicating progressively slower convergence.

It is easy to show that

$$B_k = {\bar{F}_k}^{-1} B_{k-1} \bar{F}_k = \bar{G}_k B_{k-1} {\bar{G}_k}^{-1}.$$

Letting

$$F_k = \bar{F}_1 \bar{F}_2 \cdots \bar{F}_k, \qquad G_k = \bar{G}_k \cdots \bar{G}_2 \bar{G}_1,$$

we have

$$B_k = {F_k}^{-1} \hat{B} F_k = G_k \hat{B} {G_k}^{-1}. \tag{9}$$

We can also show easily by induction that

$$(\hat{B} + \mu I)^k = \mu^k F_k G_k. \tag{10}$$

We prefer to work with a small parameter rather than the large parameter μ, so let $\epsilon = 1/\mu$. Then (8) and (10) can be rewritten as

$$I + \epsilon B_{k-1} = \bar{F}_k \bar{G}_k, \qquad \bar{G}_k \bar{F}_k = I + \epsilon B_k. \tag{11}$$

$$(I + \epsilon \hat{B})^k = F_k G_k. \tag{12}$$

We used the assumption that $\Lambda(\mathcal{G})$ contains the identity matrix to write the shifted FG algorithm in the form (8), which we then rewrote in the equivalent form (11). This assumption will not be used anywhere else. If we use (11) as our point of departure instead of the shifted FG algorithm, we can drop the assumption.

It is useful to view the FG algorithm (11) as a discrete-time dynamical system governed by the difference equation

$$B_k = B_{k-1} + \frac{1}{\epsilon}(\bar{G}_k \bar{F}_k - \bar{F}_k \bar{G}_k). \tag{13}$$

We will view each step forward as a time step of length ϵ. Thus the elapsed time after k steps is $k\epsilon$. If we let $\epsilon \to 0$ and $k \to \infty$, holding $t = k\epsilon$ fixed, the difference equation (13) is transformed into a differential equation, a continuous analogue of the FG algorithm.

In order to carry out the limiting process rigorously, we need to know that certain limits exist. The matrices B_k, F_k, G_k, $\bar{F}_k$, and $\bar{G}_k$ are all functions of ϵ as well as k, and we will write $B_k = B(k, \epsilon)$, for example, when we want to emphasize this fact. From (12) it is clear that $F(k, \epsilon)$ and $G(k, \epsilon)$ are well defined for all complex k and sufficiently small complex ϵ, and they are analytic in both variables. From (9) and (11) we see that the same is true of $B(k, \epsilon)$, $\bar{F}(k, \epsilon)$, and $\bar{G}(k, \epsilon)$ as well. Since we intend to hold $t = k\epsilon$ fixed as we pass to the limit, it is useful to write $F(k, \epsilon) = F(t/\epsilon, \epsilon)$, for example. With this notation we can rewrite (12) as

$$(I + \tfrac{\epsilon}{t}(t\hat{B}))^{t/\epsilon} = F(t/\epsilon, \epsilon)G(t/\epsilon, \epsilon). \tag{14}$$

The limit of the left-hand side as $\epsilon \to 0$ is $\exp(t\hat{B})$. Suppose $\exp(t\hat{B})$ has an FG decomposition. (This will certainly be the case if t is sufficiently small.) Define $F(t) \in \mathcal{F}$ and $G(t) \in \mathcal{G}$ to be the FG factors of $\exp(t\hat{B})$; that is,

$$\exp(t\hat{B}) = F(t)G(t). \tag{15}$$

Since the FG decomposition is analytic, it is certainly continuous. Thus (14) and (15) imply that

$$\lim_{\epsilon \to 0} F(t/\epsilon, \epsilon) = F(t), \qquad \lim_{\epsilon \to 0} G(t/\epsilon, \epsilon) = G(t).$$

For fixed t the left-hand side of (14) is an analytic function of ϵ in a deleted neighborhood of zero, with a removable singularity at $\epsilon = 0$. Therefore $F(t/\epsilon, \epsilon)$ and $G(t/\epsilon, \epsilon)$ are also analytic functions of ϵ with removable singularities at zero, provided $\exp(t\hat{B})$ has an FG decomposition. Define another analytic function $B(t)$ by

$$B(t) = F(t)^{-1}\hat{B}F(t) = G(t)\hat{B}G(t)^{-1}. \tag{16}$$

The equations (9) can be rewritten as

$$B(t/\epsilon, \epsilon) = F(t/\epsilon, \epsilon)^{-1}\hat{B}F(t/\epsilon, \epsilon) = G(t/\epsilon, \epsilon)\hat{B}G(t/\epsilon, \epsilon)^{-1}.$$

Therefore $B(t/\epsilon, \epsilon)$ is an analytic function of ϵ in a neighborhood of zero. Taking the limit as $\epsilon \to 0$, we find that

$$\lim_{\epsilon \to 0} B(t/\epsilon, \epsilon) = B(t).$$

The function $B(t)$ is, in fact, our continuous analogue of the sequence (B_k).

We now have in hand the tools to prove the following interpolation result: Let (A_k) be the output of the FG algorithm with zero shifts, starting with $A_0 = \hat{A} = \exp(\hat{B})$. Then

$$A_k = \exp(B(k)), \qquad k = 0, 1, 2, 3, \cdots.$$

The main tools for proving this are (15) and its discrete analogue $\hat{A}^k = F_k G_k$, which holds for the unshifted FG algorithm. See [19] for a proof. Rutishauser stated the

LR case of (15), but he did not arrive at it in the same manner as we have here. He may have been unaware of the interpolation result, as he did not mention it in [14].

We will now derive the continuous analogue of the FG algorithm, i.e., the differential equation that $B(t)$ satisfies. The usual approach is just to differentiate (15) and (16). This yields differential equations for $F(t)$ and $G(t)$, as well as $B(t)$. Now let us see how Rutishauser obtained them by passing to a limit. For this we need Taylor expansions of the quantities $\bar{F}_k = \bar{F}(t/\epsilon, \epsilon)$ and $\bar{G}_k = \bar{G}(t/\epsilon, \epsilon)$, which appear in (13). The first equation in (11) can be written as

$$I + \epsilon B((t-\epsilon)/\epsilon, \epsilon) = \bar{F}(t/\epsilon, \epsilon)\bar{G}(t/\epsilon, \epsilon). \tag{17}$$

Letting $A(\epsilon) = I + \epsilon B((t-\epsilon)/\epsilon, \epsilon)$ we see that $A(\epsilon)$ is analytic, and $\lim_{\epsilon \to 0} A(\epsilon) = I$. Thus $\bar{F}(t/\epsilon, \epsilon)$ and $\bar{G}(t/\epsilon, \epsilon)$ have Taylor expansions

$$\begin{aligned} \bar{F}(t/\epsilon, \epsilon) &= I + \epsilon X(t) + \epsilon^2 M(t) + O(\epsilon^3), \\ \bar{G}(t/\epsilon, \epsilon) &= I + \epsilon Y(t) + \epsilon^2 N(t) + O(\epsilon^3), \end{aligned} \tag{18}$$

where $X(t) \in \Lambda(\mathcal{F})$ and $Y(t) \in \Lambda(\mathcal{G})$. Substituting the expansions (18) into (17), we find that

$$B((t-\epsilon)/\epsilon, \epsilon) = X(t) + Y(t) + O(\epsilon).$$

Letting $\epsilon \to 0$, we obtain

$$B(t) = X(t) + Y(t).$$

Since $X(t) \in \Lambda(\mathcal{F})$ and $Y(t) \in \Lambda(\mathcal{G})$, it follows that

$$X(t) = \rho(B(t)) \quad \text{and} \quad Y(t) = \sigma(B(t)),$$

where ρ and σ are defined by (6). We are finally ready to pass to the limit. Following Rutishauser we substitute the expansions (18) into (13), which can then be rewritten as

$$\frac{B(t/\epsilon, \epsilon) - B((t-\epsilon)/\epsilon, \epsilon)}{\epsilon} = [Y(t), X(t)] + O(\epsilon),$$

where $[Y, X] = YX - XY$. Taking the limit as $\epsilon \to 0$, we obtain

$$\dot{B}(t) = [\sigma(B(t)), \rho(B(t))]. \tag{19}$$

This is our continuous analogue of the FG algorithm. Since $[\sigma(B), \rho(B)] = [B, \rho(B)] = [\sigma(B), B]$, (19) also has the forms

$$\dot{B} = [B, \rho(B)] \quad \text{and} \quad \dot{B} = [\sigma(B), B]. \tag{20}$$

This shows that this flow is a member of the family of FG flows introduced in [19].

The differential equations for F and G are also easily obtained.

$$\frac{F_k - F_{k-1}}{\epsilon} = F_{k-1}\frac{(\bar{F}_k - I)}{\epsilon} = F_{k-1}\{\rho(B(t)) + O(\epsilon)\}.$$

Taking the limit, we have

$$\dot{F} = F\rho(B) = F\rho(F^{-1}\hat{B}F). \tag{21}$$

Similarly,

$$\dot{G} = \sigma(B)G = \sigma(G\hat{B}G^{-1})G. \tag{22}$$

These equations are familiar from [19]. They were also stated by Rutishauser [14] for the LR case.

A second way to obtain the differential equation for $B(t)$ is to use the equation

$$B_k = \bar{F}_k{}^{-1} B_{k-1} \bar{F}_k. \tag{23}$$

From the first expansion in (18) it is obvious that

$$\bar{F}_k{}^{-1} = \bar{F}(t/\epsilon, \epsilon)^{-1} = I - \epsilon X(t) + O(\epsilon^2).$$

Substituting this expansion and the first expansion of (18) into (23), we find that

$$B_k = B_{k-1} + \epsilon[B_{k-1}, X(t)] + O(\epsilon^2). \tag{24}$$

Thus

$$\frac{B(t/\epsilon, \epsilon) - B((t-\epsilon)/\epsilon, \epsilon)}{\epsilon} = [B((t-\epsilon)/\epsilon, \epsilon), X(t)] + O(\epsilon). \tag{25}$$

Taking the limit as $\epsilon \to 0$ we obtain

$$\dot{B}(t) = [B(t), \rho(B(t))],$$

the first equation of (20). We could equally well have started with the equation

$$B_k = \bar{G}_k B_{k-1} \bar{G}_k{}^{-1}. \tag{26}$$

This gives

$$B_k = B_{k-1} + \epsilon[Y(t), B_{k-1}] + O(\epsilon^2),$$

which leads to $\dot{B} = [\sigma(B), B]$, the second equation of (20). The nicest feature of this approach is that it can be generalized. We will carry out the generalization in the next section.

In order to carry out the construction, we have had to assume that t is such that $\exp(t\hat{B})$ has an FG decomposition. We have already shown in [19] that the points at which $\exp(t\hat{B})$ does not have an FG decomposition are exactly the points at which the flow has singularities.

3. Carrying the generalization further. So far we have derived FG flows of the form $\dot{B} = [B, \rho(B)]$. This is a special case of a more general family of autonomous FG flows of the form $\dot{B} = [B, \rho(f(B))]$, which we studied in [19]. Here f is any locally analytic function defined on the spectrum of $\hat{B}$. In the present section we will show how to derive this entire family of flows by taking limits.

We will make use of the following generalization of the FG algorithm. Instead of choosing a sequence of shifts (σ_k), we choose a sequence (p_k) of analytic functions defined on the spectrum of $\hat{B}$. Then, starting with $B_0 = \hat{B}$, we define, for $k = 1, 2, 3, \cdots$

$$\begin{cases} B_k = \bar{F}_k{}^{-1} B_{k-1} \bar{F}_k = \bar{G}_k B_{k-1} \bar{G}_k{}^{-1}, \\ \text{where} \quad p_k(B_{k-1}) = \bar{F}_k \bar{G}_k, \quad \bar{F}_k \in \mathcal{F}, \ \bar{G}_k \in \mathcal{G}. \end{cases} \tag{27}$$

If we choose $p_k(x) = x - \sigma_k$, (27) reduces to the shifted FG algorithm introduced in the previous section. The choice $p_k(x) = (x - \sigma_k)(x - \tau_k)$ gives the *double-step* FG algorithm. In actual implementations the p_k would be chosen with the intent of accelerating convergence, but for our purposes we will choose $p_k(x) = 1 + \epsilon f(x)$, $k = 1, 2, 3, \cdots$, where f is a fixed analytic function defined on the spectrum of $\hat{B}$. Then

$$I + \epsilon f(B_{k-1}) = \bar{F}_k \bar{G}_k. \tag{28}$$

Defining $F_k = \bar{F}_1 \cdots \bar{F}_k$ and $G_k = \bar{G}_k \cdots \bar{G}_1$, we have

$$B_k = {F_k}^{-1} \hat{B} F_k = G_k \hat{B} {G_k}^{-1}$$

and

$$(I + \epsilon f(\hat{B}))^k = F_k G_k. \tag{29}$$

In the case $f(x) = x$, (28) and (29) reduce to (11, first equation) and (12), respectively. Letting $t = k\epsilon$ and using the same notational conventions as before, we can rewrite (29) as

$$(I + \tfrac{\epsilon}{t}(tf(\hat{B})))^{t/\epsilon} = F(t/\epsilon, \epsilon) G(t/\epsilon, \epsilon),$$

which is analogous to (14). Obviously the limit of the left-hand side as $\epsilon \to 0$ is $\exp(tf(\hat{B}))$. The entire development of the previous section can be generalized in a straightforward manner. Now $F(t)$ and $G(t)$ are defined by the FG decomposition

$$\exp(tf(\hat{B})) = F(t)G(t).$$

The Taylor expansions

$$\bar{F}(t/\epsilon, \epsilon) = I + \epsilon X(t) + O(\epsilon^2),$$

$$\bar{G}(t/\epsilon, \epsilon) = I + \epsilon Y(t) + O(\epsilon^2)$$

continue to be valid, but now

$$X(t) + Y(t) = f(B(t)),$$

so

$$X(t) = \rho(f(B(t))) \qquad \text{and} \qquad Y(t) = \sigma(f(B(t))).$$

Equations (23), (24), and (25) continue to hold, except that now $X(t) = \rho(f(B(t)))$. Taking the limit as $\epsilon \to 0$ in (25), we obtain

$$\dot{B} = [B, \rho(f(B))],$$

as desired. Alternatively we can start from (26) and obtain the form $\dot{B} = [\sigma(f(B)), B]$. Finally, in analogy with (21) and (22) we find that

$$\dot{F} = F\rho(F^{-1} f(\hat{B}) F), \qquad \dot{G} = \sigma(G f(\hat{B}) G^{-1}) G.$$

This flow has the interpolation property $\exp\{f(B(k))\} = A_k$, where (A_k) is the output of the FG algorithm with zero shifts, starting with $A_0 = \exp\{f(\hat{B})\}$. In particular, the choice $f(x) = \log x$ yields a flow that interpolates the FG algorithm.

REFERENCES

[1] M. CHU, *The generalized Toda flow, the QR algorithm, and the centre manifold theory,* SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 187–201.

[2] P. DEIFT, T. NANDA, AND C. TOMEI, *Differential equations for the symmetric eigenvalue problem,* SIAM J. Numer. Anal., 20 (1983), pp. 1–22.

[3] H. FLASCHKA, *The Toda lattice,* II, *existence of integrals,* Phys. Rev. B, 9 (1974), pp. 1924–1925.

[4] S. HELGASON, *Differential Geometry, Lie Groups and Symmetric Spaces,* Academic Press, New York, 1978.

[5] M. HÉNON, *Integrals of the Toda lattice,* Phys. Rev. B, 9 (1974), pp. 1921–1923.

[6] P. HENRICI, *Applied and Computational Complex Analysis, Vol.* I, John Wiley, New York, 1974.

[7] J. MOSER, *Dynamical Systems, Theory and Applications,* Springer-Verlag, Berlin, New York, 1975.

[8] ———, *Finitely many mass points on the line under the influence of an exponential potential—An integrable system,* in Dynamical Systems, Theory and Applications, Springer-Verlag, Berlin, New York, 1975, pp. 467–497 in [7].

[9] H. RUTISHAUSER, *Der Quotienten-Differenzen-Algorithmus,* Z. Angew. Math. Phys., 5 (1954), pp. 233–251.

[10] ———, *Anwendungen des Quotienten-Differenzen-Algorithmus,* Z. Angew. Math. Phys., 5 (1954), pp. 496–508.

[11] ———, *Ein infinitesimales Analogon zum Quotienten-Differenzen-Algorithmus,* Arch. Math., 5 (1954), pp. 132–137.

[12] ———, *Der Quotienten-Differenzen-Algorithmus,* Mitt. Inst. Angew. Math., No. 7, ETH, Zürich, 1957. MR 19–686. (This report gathers the material of [9], [10], [11], and parts of [14] into a single volume.)

[13] ———, *Une méthode pour la détermination des valeurs propres d'une matrice,* Comptes Rendus Acad. Sci. Paris, 240 (1955), pp. 34–36.

[14] ———, *Solution of eigenvalue problems with the LR-transformation,* National Bureau of Standards Applied Mathematics Series, 49 (1958), pp. 47–81.

[15] M. SHUB AND A. T. VASQUEZ, *Some linearly induced Morse-Smale systems, the QR algorithm and the Toda lattice,* Contemp. Math., 64 (1987), pp. 181–194.

[16] W. W. SYMES, *The QR algorithm and scattering for the finite nonperiodic Toda lattice,* Physica, 4D (1982), pp. 275–280.

[17] M. TODA, *Waves in nonlinear lattice,* Prog. Theoret. Phys. (Supp.), 45 (1970), pp. 174–200.

[18] D. S. WATKINS, *Isospectral flows,* SIAM Rev., 26 (1984), pp. 379–391.

[19] D. S. WATKINS AND L. ELSNER, *Self-similar flows,* Linear Algebra Appl., 110 (1988), pp. 213–242.

[20] ———, *Self-equivalent flows associated with the generalized eigenvalue problem,* Linear Algebra Appl., 118 (1989), pp. 107–127.

[21] ———, *Self-equivalent flows associated with the singular value decomposition,* SIAM J. Matrix Anal. Appl., 10 (1989), pp. 244–258.

[22] J. H. WILKINSON, *The Algebraic Eigenvalue Problem,* Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 312–322, April 1990

INCREMENTAL CONDITION ESTIMATION*

CHRISTIAN H. BISCHOF†

Abstract. This paper introduces a new technique for estimating the smallest singular value, and hence the condition number, of a dense triangular matrix as it is generated one row or column at a time. It is also shown how this condition estimator can be interpreted as trying to approximate the secular equation with a simpler rational function. While one can construct examples where this estimator fails, numerical experiments demonstrate that despite its small computational cost, it produces reliable estimates. Also given is an example that shows the advantage of incorporating the incremental condition estimation strategy into the QR factorization algorithm with column pivoting to guard against near rank deficiency going unnoticed.

1. Introduction. Let $A = [a_1, \cdots, a_n]$ be an $m \times n$ matrix and let $\sigma_1 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$ be the singular values of A. The smallest singular value

$$\sigma_{\min} \equiv \sigma_{\min(m,n)}$$

of A is important in that it measures how close A is to a rank-deficient matrix [10, p. 19]. The condition number

$$\kappa_2(A) \equiv \frac{\sigma_1}{\sigma_{\min}}$$

which determines the sensitivity of equation systems involving A [10], [19], also depends crucially on $\sigma_{\min}$. For most practical purposes an order-of-magnitude estimate of $\sigma_{\min}$ or $\kappa_2(A)$ is sufficient. Most of the schemes for estimating $\sigma_{\min}$ and $\kappa_2(A)$ apply to triangular matrices, since in common applications A will be factored into a product of matrices involving a triangular matrix. An excellent survey of those so-called condition estimation techniques for triangular matrices as well as their applications is given by Higham [12].

All of these condition estimators do, however, estimate the smallest singular value of a triangular matrix *after* it has been factored and cannot be used to monitor the factorization of an upper triangular matrix as it is generated one column at a time (or a lower triangular matrix as it is generated one row at a time). Since a matrix and its transpose have the same singular values, we assume without loss of generality that we are generating a lower triangular matrix L one row at a time. The advantage of the incremental condition estimator we present in this paper is that it does allow us to *update estimates of* $\sigma_{\min}(L)$ *and* $\kappa_2(L)$ *cheaply as* L *is generated* one row at a time. In particular, if we are given an $n \times n$ lower triangular matrix L, an approximate singular vector x such that $\sigma_{\min}(L) \approx 1/\|x\|_2$, and a new row (v^T, γ) of L, we are able to obtain a new approximate singular vector of

$$L' \equiv \begin{pmatrix} L & 0 \\ v^T & \gamma \end{pmatrix}$$

* Received by the editors January 6, 1989; accepted for publication September 6, 1989. This work was partially supported by the U.S. Army Research Office through the Mathematical Science Institute of Cornell University, by the Office of Naval Research under contract N00014-83-K-0640, by National Science Foundation under contract CCR 86-02310, and the Applied Mathematical Sciences subprogram of the Office of Energy Research, U. S. Department of Energy under contract W-31-109-Eng-38. Computations were performed in part at the facilities of the Cornell Computational Optimization Project, which is supported by the National Science Foundation under contract DMS-87-06133.

† Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Ave., Argonne, Illinois 60439 (bischof@mcs.anl.gov).

such that $\sigma_{\min}(L') \approx 1/\|y\|_2$ with $3n$ flops[1] and *without accessing L again.* In this fashion, incremental condition estimation makes it possible to monitor the condition number of L as it is generated.

In the next section we motivate the idea behind incremental condition estimation and describe the algorithm. Section 3 explains why incremental condition estimation works and explores its limitations. Section 4 presents numerical results showing the robustness of the proposed scheme. In § 5 we give an example of the usefulness of incremental condition estimation in the context of the QR factorization with column pivoting. Lastly, we summarize our contributions and outline directions for further research.

2. Estimating the smallest singular value of a triangular matrix. A common idea underlying condition estimators [7], [8], [11] is to exploit the implication

$$Lx = d \implies \frac{1}{\sigma_{\min}(L)} = \|L^{-1}\|_2 \geq \frac{\|L^{-1}d\|_2}{\|d\|_2} = \frac{\|x\|_2}{\|d\|_2}$$

by generating a large norm solution x to a moderately sized right-hand side d and then to use

$$\hat{\sigma}_{\min}(L) := \frac{\|d\|_2}{\|x\|_2}$$

as an estimate for $\sigma_{\min}(L)$. We hope that x will be an approximate singular vector corresponding to the smallest singular value and that as a consequence $\hat{\sigma}_{\min}(L)$ will not be too much of an overestimate of $\sigma_{\min}(L)$.

For our incremental condition estimator we want to monitor $\sigma_{\min}(L)$ as L is generated one row at a time. As a consequence, it is not feasible to reaccess L, since that would require $O(n^2)$ flops at every updating step, which is too expensive. To be more precise, given a good estimate $\hat{\sigma}_{\min}(L)$ defined by a large norm solution x to $Lx = d$ and a new row (v^T, γ) of L, we want to obtain a large norm solution y to

$$L'y = \begin{pmatrix} L & 0 \\ v^T & \gamma \end{pmatrix} y = d'$$

without accessing L again. None of the condition estimators surveyed by Higham [12] has that property.

We achieve this objective by choosing the new right-hand side d' in a fashion that allows us to reuse the previous approximate singular vector x. The idea is as follows.

Given x such that $Lx = d$ with $\|d\|_2 = 1$ and $\sigma_{\min}(L) \approx 1/\|x\|_2$, find $s := \sin\varphi$ and $c := \cos\varphi$ such that $\|y\|_2$ is maximized where y solves

$$\begin{pmatrix} L & 0 \\ v^T & \gamma \end{pmatrix} y = \begin{pmatrix} sd \\ c \end{pmatrix}. \tag{1}$$

By setting up the problem in this way we obtain immediately

$$y = \begin{pmatrix} sx \\ \frac{c - s\alpha}{\gamma} \end{pmatrix} \tag{2}$$

[1] Either an addition or a multiplication of two floating-point numbers is counted as a *flop.*

where

$$\alpha = v^T x$$

and $\|d'\|_2 = \|d\|_2 = 1$. The proper (s, c) is found using (2) and expressing

$$\|y\|_2^2 = \frac{1}{\gamma^2} \begin{pmatrix} s & c \end{pmatrix} B \begin{pmatrix} s \\ c \end{pmatrix} \tag{3}$$

where

$$B = \begin{pmatrix} 1+\beta & -\alpha \\ -\alpha & 1 \end{pmatrix} \tag{4}$$

with

$$\beta = \gamma^2 x^T x + \alpha^2 - 1. \tag{5}$$

Assuming that $\gamma \neq 0$ (otherwise L' is singular and $\sigma_{\min}(L') = 0$), the optimal $\begin{pmatrix} s \\ c \end{pmatrix}$ is the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ of B. In the case $\alpha \neq 0$ we define

$$\eta = \frac{\beta}{2\alpha} \quad \text{and} \quad \mu = \eta + \operatorname{sign}(\alpha)\sqrt{\eta^2 + 1} \tag{6}$$

and obtain

$$\lambda_{\max} = \alpha\mu + 1 \tag{7}$$

and

$$\begin{pmatrix} s \\ c \end{pmatrix} = \frac{1}{\sqrt{\mu^2 + 1}} \begin{pmatrix} \mu \\ -1 \end{pmatrix}. \tag{8}$$

For the special case $\alpha = 0$ we obtain

$$B = \begin{pmatrix} \gamma^2 x^T x & 0 \\ 0 & 1 \end{pmatrix}$$

in (4) and choose

$$\begin{pmatrix} s \\ c \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{if } |\gamma| \, \|x\|_2 > 1, \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \text{otherwise.} \end{cases}$$

Having computed the optimal $\begin{pmatrix} s \\ c \end{pmatrix}$, we compute a new approximate singular vector y as defined by (2) and the resulting estimate for the smallest singular value of $\sigma_{\min}(L')$ of L' is

$$\hat{\sigma}_{\min}(L') = \frac{1}{\|y\|_2}. \tag{9}$$

Given L, we need save only the current approximate singular vector x to compute an estimate for the smallest singular value of L'. Furthermore, the calculation is inexpensive. Since

$$\|y\|_2 = \frac{\sqrt{\lambda_{\max}}}{|\gamma|} \tag{10}$$

we need only $3k$ flops (a dot product and a scaling) to arrive at an estimate for $\sigma_{\min}(L')$. In particular, it costs only $\frac{3}{2}n^2$ flops to run this condition estimator alongside the generation of an $n \times n$ triangular matrix.

The incremental condition estimation idea can also be used to obtain good estimates for $\sigma_{\max}$, the largest singular value. Traditionally, the norm of the largest column, i.e.,

$$r_{\max} := \max_{1 \le i \le n} \|a_i\|_2 \quad (\le \sigma_{\max}) \tag{11}$$

is used as an estimate for $\sigma_{\max}(A)$. It is easy to show that

$$r_{\max}\sqrt{n} \ge \sigma_{\max}$$

and hence $r_{\max}$ underestimates $\sigma_{\max}$ at most by a factor of $\sqrt{n}$. For large n this could be a substantial underestimate. Furthermore, in the estimate

$$\hat{\kappa}_2(L) = \frac{r_{\max}}{\hat{\sigma}_{\min}} \quad (\le \kappa_2(L)), \tag{12}$$

the errors in both $r_{\max}$ and $\hat{\sigma}_{\min}$ multiply.

To obtain a better estimate for $\sigma_{\max}$, observe that an approximate singular vector x for $\sigma_{\max}$ is a *small norm solution* to $Lx = d, \|d\|_2 = 1$ for a suitably chosen vector d. Note that this vector d is different from the one involved in estimating the smallest singular value. Given such a vector x for $Lx = d$, we compute (s, c) such that $\|y\|_2$ is *minimized* where y solves (1). The optimal (s, c) is then computed as in (5) through (8), except that we define

$$\mu = \eta - \operatorname{sign}(\alpha)\sqrt{\eta^2 + 1}.$$

We mention that the incremental condition estimation scheme is related to the two-norm condition estimator suggested by Cline, Conn, and Van Loan [7], [18]. While their scheme looks backward and forward in a matrix, our scheme looks only backward to allow for the estimator to proceed in an incremental fashion.

3. Limitations. Given that there are counterexamples for computationally more expensive condition estimation schemes [12], it is not surprising that incremental condition estimation can fail as well. To understand why incremental condition estimation works, consider the singular value decomposition

$$L = U\Sigma V^T, U = [u_1, \cdots, u_n], V = [v_1, \cdots, v_n]$$

of L and assume that incremental condition estimation was exact and computed

$$x = \frac{1}{\sigma_{\min}(L)} v_n. \tag{13}$$

x is the largest possible solution to an equation system $Lx = d$, where $\|d\|_2 = 1$. Letting

$$L' = \begin{pmatrix} L & 0 \\ v^T & \gamma \end{pmatrix}$$

consider the so-called *secular equations* [4], [5] for the singular values of L'. The roots

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n+1} \geq 0$$

of

$$f(\lambda) := \sum_{i=1}^{n} \frac{(v^T v_i)^2}{\sigma_i^2(L) - \lambda} - \frac{\gamma^2}{\lambda} + 1 \tag{14}$$

determine the singular values of L' in that

$$\sigma_i^2(L') = \lambda_i.$$

Now define

$$\tilde{f}(\lambda) := \frac{(v^T v_n)^2}{\sigma_{\min}^2(L) - \lambda} - \frac{\gamma^2}{\lambda} + 1. \tag{15}$$

The smallest root $\tilde{\lambda}$ of $\tilde{f}$ is

$$\tilde{\lambda} = \frac{1}{2}\left(\tau - \sqrt{\tau^2 - 4\gamma^2\sigma_{\min}^2(L)}\right)$$

where

$$\tau = \sigma_{\min}^2(L) + (v^T v_n)^2 + \gamma^2.$$

f and $\tilde{f}$ are identical if

$$| v^T v_n | = \|v\|_2$$

and then

$$\tilde{\lambda} = \lambda_{n+1} = \sigma_{\min}^2(L').$$

In this special case it is also easy to verify that $\lambda_{\max}$, as defined in (7), is

$$\lambda_{\max} = \frac{1}{2\sigma_{\min}^2(L)}\left(\tau + \sqrt{\tau^2 - 4\gamma^2\sigma_{\min}^2(L)}\right)$$

and the resulting estimate for $\hat{\sigma}_{\min}(L')$ defined by (9) and (10) satisfies

$$\hat{\sigma}_{\min}^2(L') = \sqrt{\tilde{\lambda}}$$

and hence is exact.

This analysis shows that incremental condition estimation can be viewed as trying to approximate the secular equation (14) with the simpler rational function (15). This is a reasonable strategy, since the smallest root $\tilde{\lambda}$ of $\tilde{f}$ is a good approximation to the

smallest root λ_{n+1} of f when the roots of f are reasonably well separated and $v^T v_n$ is not too small in comparison with $\|v\|_2$.

In the general case, $\tilde{\lambda} \neq \sigma_{\min}(L')^2$, but instead $\tilde{\lambda} = \sigma_{\min}(\tilde{L})^2$, where

$$\tilde{L} = \begin{pmatrix} L & 0 \\ \tilde{v} & \gamma \end{pmatrix}$$

with

$$\tilde{v} := \frac{v^T x}{\|x\|_2^2} x = (v^T v_n) v_n$$

being the projection of v onto $\operatorname{span}(v_n) = \operatorname{span}(x)$. Hence, incremental condition estimation approximates $\sigma_{\min}(L')$ by $\sigma_{\min}(\tilde{L})$.

This analysis shows that incremental condition estimation can be unreliable when

$$| v^T x | \ll \|x\|_2 \|v\|_2. \tag{16}$$

Then the singular values of L' and $\tilde{L}$ can differ by an arbitrary amount. It should be noted, however, that even in the case where v and x are orthogonal, incremental condition estimation need not necessarily fail. If γ is small enough, the contribution of v to the change of the smallest singular values is negligible and incremental condition estimation still works. This is the reason why in the QR factorization with column pivoting [6], [9], [10] the smallest diagonal element usually works as a predictor of $\sigma_{\min}$.

We also point out that the previous analysis leads to a counterexample for the two-norm condition estimator suggested by Cline, Conn, and Van Loan [7], [18], since the nth step of their condition estimator is identical to incremental condition estimation. No counterexample for this condition estimator had been known thus far.

4. Numerical experiments. To assess the reliability of incremental condition estimation, we performed the test suite suggested by Higham [12] using PRO-MATLAB [16]. Three different types of test matrices are employed. In each test, upper triangular matrices R were generated by computing the Householder QR factorization of various $n \times n$ matrices A for $n = 25, 50, 75, 100$ both with and without column pivoting.

Test 1. The elements of A were chosen as random numbers from the uniform distribution on $[-1, 1]$. Fifty matrices were generated for each n. As observed by Higham, this type of matrix usually is well conditioned. Over the whole test the minimum, maximum, and average values of the two-norm condition number $\kappa_2(A) = \sigma_1/\sigma_n$ were $24, 1.5 \cdot 10^4$, and $2.4 \cdot 10^6$, respectively.

Tests 2 *and* 3. In these tests we used random matrices A with preassigned singular value distributions $\{\sigma_i\}$. Random orthogonal matrices U and V were generated using the method of Stewart [17] and then A was formed as $A = U\Sigma V^T$. For each value of n and each singular value distribution, fifty matrices were generated by choosing different matrices U and V. For Test 2 we chose the exponential distribution

$$\sigma_i = \alpha^i, \ 1 \leq i \leq n$$

where $\alpha < 1$ is determined by $\kappa_2(A)$. For Test 3, we chose the sharp-break distribution

$$1 = \sigma_1 = \cdots = \sigma_{n-1} > \sigma_n = \frac{1}{\kappa_2(A)}.$$

TABLE 1
Max/avg values of $\hat{\sigma}_{\min}(R)/\sigma_{\min}(R)$.

Test 1: Uniform Distribution of Singular Values		
n	No pivoting	Column pivoting
25	6.6/2.4	2.9/2.0
50	7.0/3.1	5.6/2.6
75	8.5/3.5	4.9/3.2
100	9.6/4.1	5.5/3.4

Test 2: Exponential Distribution of Singular Values						
	No pivoting			Column pivoting		
n	$\kappa_2 = 10$	$\kappa_2 = 10^6$	$\kappa_2 = 10^{12}$	$\kappa_2 = 10$	$\kappa_2 = 10^6$	$\kappa_2 = 10^{12}$
25	1.6/1.3	4.2/2.5	11/3.6	1.7/1.4	3.3/2.2	4.1/2.1
50	1.7/1.4	6.1/2.9	6.9/3.8	1.6/1.4	3.4/2.4	4.1/3.0
75	1.5/1.4	4.0/3.0	6.1/3.9	1.6/1.4	3.5/2.7	5.1/3.1
100	1.5/1.4	4.3/3.0	7.3/4.1	1.6/1.4	3.4/2.8	5.0/3.5

TABLE 2
Max/avg values of $\kappa_2(R)/\hat{\kappa}_2(R)$.

Test 1: Uniform Distribution of Singular Values		
n	No pivoting	Column pivoting
25	6.8/2.4	2.9/2.0
50	7.0/3.1	5.6/2.6
75	8.6/3.5	4.9/3.2
100	9.7/4.1	5.5/3.4

Test 2: Exponential Distribution of Singular Values						
	No pivoting			Column pivoting		
n	$\kappa_2 = 10$	$\kappa_2 = 10^6$	$\kappa_2 = 10^{12}$	$\kappa_2 = 10$	$\kappa_2 = 10^6$	$\kappa_2 = 10^{12}$
25	2.2/1.6	6.8/3.0	21/4.1	2.1/1.5	3.3/2.2	4.1/2.1
50	2.1/1.6	7.1/3.5	9.7/4.5	1.8/1.6	3.5/2.6	4.2/3.0
75	1.8/1.6	7.0/3.6	9.7/4.8	1.8/1.6	4.4/2.9	5.9/3.3
100	1.8/1.6	5.2/3.8	8.2/5.1	1.9/1.6	3.9/3.1	5.4/3.7

The figures given in Table 1 are the ratios

$$\hat{\sigma}_{\min}(R)/\sigma_{\min}(R) \geq 1 .$$

The figures in Table 2 are the ratios

$$\kappa_2(R)/\hat{\kappa}_2(R) \geq 1$$

where

$$\hat{\kappa}_2(R) = \frac{\hat{\sigma}_{\max}}{\hat{\sigma}_{\min}}$$

and both $\hat{\sigma}_{\max}$ and $\hat{\sigma}_{\min}$ have been computed using incremental condition estimation. The first number in each pair is the maximum ratio over the fifty matrices, and the second is the average ratio. All results were rounded to two significant digits. For Test 3 we observed a ratio of 1.0 (i.e., the estimate had at least two correct figures) in all cases. These results show that our condition estimator is reliable in producing good estimates. We overestimate $\sigma_{\min}(R)$ (or underestimate $\kappa_2(R)$) only by a small factor and the results vary only little with condition number, matrix size, and singular value distribution. Since

$$\frac{\kappa_2}{\hat{\kappa}_2} = \left(\frac{\hat{\sigma}_{\min}}{\sigma_{\min}}\right)\left(\frac{\sigma_{\max}}{\hat{\sigma}_{\max}}\right) ,$$

the comparison of Tables 1 and 2 shows that

$$\frac{\sigma_{\max}}{\hat{\sigma}_{\max}} \approx 1$$

in most cases, and hence the estimate for the largest singular value is very good.

We also note that although pivoting somewhat increases the accuracy of the condition estimator, it is not needed to obtain reliable estimates. This is in contrast to the estimator for $\sigma_{\min}$ arrived at choosing

$$\hat{\sigma}_{\min}(L) = \min_{1 \leq i \leq n} | l_{ii} | .$$

This estimator also works in an incremental fashion, but is unreliable when the matrix is not graded (as it is, for example, in the QR factorization with column pivoting). In particular, this condition estimator fails on matrices produced by the QR or Cholesky factorizations without column pivoting.

5. Guarding the QR factorization with column pivoting. A well-known strategy for extracting a set of reasonably independent columns of a given matrix A and for computing an orthonormal basis for the span of A is the QR factorization with column pivoting [1], [6], [15]. Viewed geometrically [10, p. 168, P.6.4-5] this strategy chooses at every step that column of A that is farthest away (in the two-norm sense) from the subspace spanned by the columns that were selected before. In matrix terms we are computing a QR factorization

$$AP = QR \tag{17}$$

where P is an $n \times n$ permutation matrix determined by the pivoting strategy, Q is an $m \times m$ orthogonal matrix, and R is an $m \times n$ upper triangular matrix. If A is dense, Q is typically generated as a sequence of Householder transformations [10, p. 37].

We hope that the rank of A will reveal itself by a small trailing subblock of R: if we partition R into

$$\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \tag{18}$$

with an $r \times r$ lower right-hand block R_{22}, then it is easy to show [10, p. 19] that

$$\sigma_{n-r+1}(A) \leq \|R_{22}\|_2 .$$

Hence, if R_{22} is small, A can be considered to have numerical rank $n-r$, and the first $n-r$ columns of Q form an orthonormal basis for the range space of A.

This strategy works well in practice, but there are counterexamples where it fails without giving any indication of failure. A well-known example (originally suggested by Kahan) is

$$A_n = \operatorname{diag}(1, s, s^2, \cdots, s^{n-1}) \begin{pmatrix} 1 & -c & \cdots & \cdots & -c \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & -c \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} + \Delta \tag{19}$$

where $\Delta = \operatorname{diag}(n\epsilon, (n-1)\epsilon, \cdots, \epsilon)$, $c^2 + s^2 = 1$, and ϵ is the machine precision. A_n is very ill conditioned, but although each leading principal submatrix A_k ($k \leq n$) is also ill conditioned, there is a well-defined gap between σ_n and σ_{n-1}. As an example, for $n = 50$ and $c = 0.5$ we have $\sigma_{49} = 1.2 \cdot 10^{-3}$ and $\sigma_{50} = 3.7 \cdot 10^{-12}$. Even in floating-point arithmetic the matrix is its own QR factorization with pivoting but no trailing block of R is small enough to reveal its ill conditioning.

However, the incremental condition estimator integrated into the QR factorization algorithm detects the ill conditioning of the leading principal submatrices A_k and, in fact, we observed that in this particular example it never overestimates the smallest singular value of A_k, $k = 1, \cdots, 50$ by a factor of more than 1.5. So while the column pivoting scheme fails to detect rank deficiency, the incremental condition estimator prevents this failure from going unnoticed. Given its negligible cost compared to the QR factorization, this suggests the usefulness of incorporating the incremental condition estimator into the traditional column pivoting scheme. In the same spirit we believe incremental condition estimation to be useful in monitoring Gaussian elimination and Cholesky factorization. The traditional "after-the-fact" condition estimation schemes, on the other hand, would only indicate that there had been problems at some unspecified point in the factorization process.

6. Conclusions. We introduced a technique that allowed us to estimate the smallest singular value of a dense triangular matrix R as it was generated one row (or column) at a time. This strategy required only $O(n)$ flops per step and the storage of $O(n)$ words between successive steps. In particular, it was not necessary to reaccess the previously generated R when a new row or column was added to R. We also showed how this strategy is related to the approximation of the secular equations. While one can construct examples where this strategy fails, numerical experiments indicate that the suggested scheme is reliable despite its small computational cost.

In the context of the QR factorization with column pivoting we gave an example that showed the usefulness of integrating the incremental condition estimator within the factorization process. Another use of incremental condition estimation is finding those columns of a matrix that are responsible for its ill conditioning. The Householder QR factorization algorithm with column pivoting usually achieves this goal, but the traditional pivoting strategy may conflict with other desirable features such as the sparsity structure of a matrix or locality of memory reference in the program.

Incremental condition estimation performs well on all kinds of triangular matrices, whereas the (also incremental) estimate of taking the smallest diagonal entry works only on the graded matrices produced by the traditional pivoting strategy. Incremental condition estimation allows us to restrict pivoting without sacrificing numerical reliability, and hence one can tailor the pivoting strategy to conform to other requirements. An example of such an application in the context of a rank-identifying QR factorization on a distributed-memory MIMD machine is given in [3]. Other applications we are currently exploring are block QR factorization schemes for rank-deficient matrices and algorithms for rank-deficient sparse matrices.

We also mention that it would be preferable to also have a *lower* bound for $\sigma_{\min}(R)$ instead of the upper bound that the incremental condition estimator is computing. To that end we experimented with the lower bounds derived from comparison matrices [2], [13], [14] that can also be updated in an incremental fashion. We found, however, that these bounds in most cases underestimated the smallest singular value by several orders of magnitude (this is consistent with Higham's [12] results) and as a result were not of practical use.

REFERENCES

[1] Å. BJÖRCK, *Least squares methods*, in Handbook of Numerical Analysis, Vol. 1, P. Ciarlet and J. Lions, eds., North Holland, Amsterdam, 1989, to appear.

[2] N. ANDERSON AND I. KARASALO, *On computing bounds for the least singular value of a triangular matrix*, BIT, 15 (1975), pp. 1–4.

[3] C. H. BISCHOF, *A parallel* QR *factorization algorithm with controlled local pivoting*, Tech. Report ANL/MCS–P21–1088, Argonne National Laboratory, Mathematics and Computer Sciences Division, Argonne, IL, 1988.

[4] J. R. BUNCH AND C. R. NIELSEN, *Updating the singular value decomposition*, Numer. Math., 31 (1978), pp. 111–129.

[5] J. R. BUNCH, C. R. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[6] P. A. BUSINGER AND G. H. GOLUB, *Linear least squares solution by Householder transformation*, Numer. Math., 7 (1965), pp. 269–276.

[7] A. K. CLINE, A. R. CONN, AND C. F. VAN LOAN, *Generalizing the* LINPACK *Condition Estimator*, Lecture Notes in Mathematics 909, Springer-Verlag, Berlin, New York, 1982, pp. 73–83.

[8] A. K. CLINE, C. B. MOLER, G. W. STEWART, AND J. H. WILKINSON, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal., 16 (1979), pp. 368–375.

[9] G. H. GOLUB, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, London, 1983.

[11] N. J. HIGHAM, *Efficient algorithms for computing the condition number of a tridiagonal matrix*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 150–165.

[12] ———, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29 (1987), pp. 575–596.

[13] ———, *Upper bounds for the condition number of a triangular matrix*, Numerical Analysis Report No. 86, University of Manchester, England, 1983.

[14] I. KARASALO, *A criterion for truncation of the* QR*-decomposition algorithm for the singular linear least squares problem*, BIT, 14 (1974), pp. 156–166.
[15] C. L. LAWSON AND R. J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[16] C. MOLER, J. LITTLE, AND S. BANGERT, PRO-MATLAB *User's Guide*, The Mathworks, Sherborn, MA, 1987.
[17] 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.
[18] C. F. VAN LOAN, *On estimating the condition of eigenvalues and eigenvectors*, Linear Algebra Appl., 88/89 (1987), pp. 715–732.
[19] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 2, pp. 323–334, April 1990

012

A NEW ALGORITHM FOR FINDING A PSEUDOPERIPHERAL NODE IN A GRAPH*

ROGER G. GRIMES†, DANIEL J. PIERCE†, AND HORST D. SIMON‡

Abstract. A new algorithm for the computation of a pseudoperipheral node of a graph is presented, and the application of this algorithm to reordering algorithms for the solution of sparse linear systems is discussed. Numerical tests on large sparse matrix problems show the efficiency of the new algorithm. When used for some of the reordering algorithms for reducing the profile and bandwidth of a sparse matrix, the results obtained with the pseudoperipheral nodes of the new algorithm are comparable to the results obtained with the pseudoperipheral nodes produced by the SPARSPAK version of the Gibbs–Poole–Stockmeyer algorithm. The advantage of the new algorithm is that it accesses the adjacency structure of the sparse matrix in a regular pattern. Thus this algorithm is much more suitable both for a parallel and for an out-of-core implementation of the ordering phase for sparse matrix problems.

Key words. sparse matrices, reordering algorithms, bandwidth reduction, reverse Cuthill–McKee algorithm, Gibbs–Poole–Stockmeyer algorithm, eigenvalues of graphs

AMS(MOS) subject classifications. 65F05, 05C99, 94A20

1. Introduction.

Algorithms for the numerical solution of sparse linear systems of equations usually start out with reordering the coefficient matrix in order to reduce the fill-in during Gaussian elimination. Several reordering algorithms for sparse matrices require as a first step the determination of a pseudoperipheral node of the graph associated with the adjacency matrix of the problem. For example, the reverse Cuthill–McKee [3] algorithm, the automated nested dissection algorithm, the refined quotient-tree algorithm, and the one-way-dissection algorithm in SPARSPAK [7] all require the determination of a peripheral (or at least pseudoperipheral) node in the associated graph. A widely used algorithm for this purpose is due to Gibbs, Poole, and Stockmeyer [8], and was improved by George and Liu [7], and by Lewis [10]. Other related algorithms have been investigated by Smyth [15]. These heuristic algorithms do not guarantee finding a peripheral node. However, the pseudoperipheral node computed by these algorithms is usually well suited for the purposes of reordering the sparse matrix.

The idea common to all these algorithms is the concept of a rooted level structure of the graph. All these algorithms make direct use of the level structure in performing some type of search heuristic. Here we consider a new and quite different algorithm for determining a pseudoperipheral node. This new algorithm is based on the dominant eigenvector of the adjacency matrix of the graph. Even though our new algorithm does not yield a significant improvement in the performance of the reordering algorithms for sparse linear systems, there are two reasons for writing this detailed investigation of the new algorithm. First, it is indeed remarkable that an algebraic quantity such as an eigenvector can be used in the solution of a discrete graph problem. Eigenvalues of graphs have been studied extensively [4]. Aspvall and Gilbert [2] have used

* Received by the editors February 17, 1988; accepted for publication (in revised form) May 1, 1989.

† Scientific Computing and Analysis Division, Boeing Computer Services, M/S 7L-21, Seattle, Washington 98124.

‡ Numerical Aerodynamic Simulation (NAS) Systems Division, National Aeronautics and Space Administration (NASA) Ames Research Center, Mail Stop 258-5, Moffett Field, California 94035. (The author is an employee of the Scientific Computing and Analysis (SCA) Division of Boeing Computer Services.)

eigenvectors of the adjacency matrix for the graph coloring problem. Our algorithm, however, appears to be the first application of spectral properties of graphs to sparse matrix reordering problems.

Furthermore, the new algorithm is more suitable for an out-of-core or a parallel implementation. Its key computational requirement is a matrix-vector multiplication, which can be easily implemented, both out-of-core and on a parallel machine. This is in contrast to the algorithms based on a rooted level structure. The generation of a rooted level structure requires repeated access to the adjacency structure of the graph (or the sparse matrix). This involves a large number of random input/output accesses, which make programming an out-of-core version of these algorithms difficult, and their performance inefficient.

The current study of an alternative approach was motivated by the need for an out-of-core reordering algorithm for sparse matrices arising in structural analysis. Since the new reordering capability needed to be implemented in the context of an existing structural analysis package, it was bound by severe core memory limitations. These limitations were imposed rather by the structure of the package, than by actual physical limitations. Details of the implementation are reported by Grimes and Pierce in [9].

In a connected graph with n vertices and m edges an exact peripheral node can be found in $O(nm)$ time by an obvious algorithm. For sparse matrix applications, what is wanted is an almost peripheral node in $O(m)$ time. In this paper "pseudoperipheral" means "approximately peripheral," i.e., a heuristic approximation to a peripheral node. In some other contexts [7] a pair of nodes are defined to be pseudoperipheral if they both have eccentricity equal to the distance between them. The SPARSPAK algorithm finds such a pair of nodes, usually in $O(m)$ time in practice, although there are examples that can make it run for at least $O(m\sqrt{n})$ time, and perhaps more. A different algorithm gets $O(m\sqrt{n})$ time in the worst case but is not practical [12].

The current report summarizes some of the initial investigations into an alternative algorithm for determining a pseudoperipheral node. Most of the material is based on an earlier report [14]. In §2 we collect some definitions, and in §3 we present the heuristic algorithm. Section 4 presents some bounds on the dominant eigenvector of a graph, which give additional (albeit weak) justification for the heuristic algorithm. Computational issues and numerical results are discussed in §§5 and 6.

2. Definitions. Here we consider an undirected, connected graph $G = (X, E)$, where X is the set of nodes, and E is the set of edges. The elements a_{ij} of the adjacency matrix A of G are defined by

$$a_{ij} = \begin{cases} 1 & \text{if node } i \text{ and } j \text{ are adjacent, or if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

This definition differs from the common definition of an adjacency matrix for a graph (e.g., in [4]) in that we also set $a_{ii} = 1$, whereas usually the diagonal elements are set to be zero. If G is the ordered graph of a symmetric positive definite matrix M, this definition proves to be more useful for our purposes. In this case the a_{ij} could be defined directly by

$$a_{ij} = \begin{cases} 1 & \text{if} \quad m_{ij} \neq 0 \\ 0 & \text{if} \quad m_{ij} = 0, \end{cases} \tag{2}$$

i.e., the adjacency matrix reflects directly the zero-nonzero structure of a given matrix and is therefore the appropriate tool for sparse matrix computations.

Since we assumed G to be connected, the matrix A is irreducible. By the Perron-Frobenius theorem, A has a simple, positive eigenvalue λ. The corresponding eigenvector $v = (v_1, v_2, \cdots, v_n)^T$ has all components $v_i > 0$, for $i = 1, \cdots, n$. Here $n = |X|$. Therefore v can be normalized such that $\sum_{i=1}^{n} v_i = 1$. In the following we will only deal with λ and v, such that

$$Av = \lambda v, \qquad \sum_{i=1}^{n} v_i = 1, \qquad v_i > 0 \quad \text{for } i = 1, \cdots, n. \tag{3}$$

No confusion with other eigenvalues and vectors is possible. Since G is connected, every row sum of A is at least 2, for $n > 1$. Hence $\lambda \geq 2$.

We will also use the notation $A_1 > A_2$, implying that all elements of the matrix A_1 are larger than the corresponding elements of A_2. Similarly, $A > \alpha$ for $\alpha \in R$ means that all elements of A are larger than the scalar α. We will use the same notation for the componentwise comparison of vectors.

The *distance* of two nodes x_i and x_j, i.e., the length of the shortest path connecting x_i and x_j, is denoted by $d(x_i, x_j)$, or for short by d_{ij}. The *eccentricity* of a node x_i is the quantity

$$e(x_i) = \max_{j=1,\cdots,n} d(x_i, x_j). \tag{4}$$

The *diameter* of G is then defined by

$$\delta(G) = \max_{i=1,\cdots,n} e(x_i). \tag{5}$$

A node $x_i \in X$ is said to be *peripheral* if its eccentricity is equal to the graph's diameter, i.e., if $\delta(G) = e(x_i)$.

For a subset $Y \subseteq X$, the adjacency set of Y, denoted by $Adj(Y)$, is

$$Adj(Y) = \{x_i \in X - Y \mid \{x_i, x_j\} \in E \text{ for some } x_j \in Y\}. \tag{6}$$

For a node $x \in X$, the *level structure rooted at* x is the partitioning $L(x)$ of X satisfying

$$\begin{aligned} L(x) &= \{L_0(x), L_1(x), \cdots, L_{e(x)}(x)\}, \\ L_0(x) &= \{x\}, \quad L_1(x) = Adj(L_0(x)), \\ L_i(x) &= Adj(L_{i-1}(x)) - L_{i-2}(x) \qquad \text{for } i = 2, 3, \cdots, e(x). \end{aligned} \tag{7}$$

3. A heuristic algorithm for finding peripheral nodes. We are trying to find a peripheral node of the graph G, i.e., a node with maximum eccentricity. Such a node seems likely to have the greatest average distance from all other nodes. Consider now the matrix A^k. Its (i, j)th entry denotes the number of different paths (or walks) of length k leading from x_i to x_j, where paths are included, which "stay for a while" at a node, because of $a_{ii} = 1$. Now let $u = (1, 1, \cdots, 1)^T$. Then the ith component of $A^k u$ is equal to the number of paths of length k, beginning at an arbitrary node and ending in x_i. If a node x_i is "peripheral," this number will be smaller and if a node x_i lies in the "center" of the graph, this number will be larger. So for $k \to \infty$ one should obtain some average number, which indicates how many paths go "on average" through a node. But with some suitable normalization, $A^k u$ converges to the largest

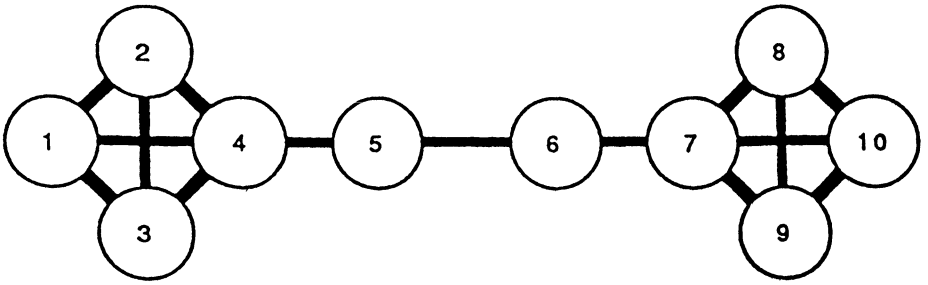


FIG. 1. *Counterexample.*

eigenvector v of A, unless u were orthogonal to this eigenvector. But this cannot happen; since $u = (1, \cdots, 1)^T$, we have

$$u^T v = \sum_{i=1}^{n} v_i = 1 \neq 0. \tag{8}$$

A similar argument has been used in [16] to determine the center of a graph for an application in geography. We use the same method for a different, but closely related application. These arguments suggest the following very simple algorithm for finding pseudoperipheral nodes of a graph:

(1) Find v, the dominant eigenvector of the adjacency matrix A.

(2) The node corresponding to the smallest component in v is a pseudoperipheral node.

This algorithm will only determine pseudoperipheral nodes and not necessarily a peripheral node. As a counterexample, consider the graph in Fig. 1. Clearly all the nodes in the two cliques at the end (x_1, x_2, x_3 and x_8, x_9, x_{10}) are peripheral. The vector v, however, is given by $v \approx$ (0.1073, 0.1073, 0.1073, 0.1211, 0.0569, 0.0569, 0.1211, 0.1073, 0.1073, 0.1073)^T. The smallest components of v are just corresponding to the "interior" nodes x_5 and x_6. It is interesting to note that the graph in Fig. 1 also serves as the standard counterexample for a perfect elimination graph for which the minimum degree algorithm does not find a perfect elimination order (see [5, p. 130]).

The proposed method will also fail if the graph is regular, that is, all vertices have the same degree. In this case the components of the dominant eigenvector are all equal. Regular graphs are not common in practice, but it is easy to construct regular graphs in which eccentricities vary widely.

4. Bounds for the dominant eigenvector. Although the example above shows that the heuristic algorithm from §3 will not always produce peripheral nodes, we are able to obtain lower bounds on the components of the dominant eigenvector. These bounds indicate that there is a certain inverse relationship between the components of the eigenvector and the eccentricity of the corresponding node.

PROPOSITION 1. *For $n > 1$ the components v_i of the dominant eigenvector v satisfy*

$$v_i \geq \frac{1}{e(x_i)(\lambda - 1)^{e(x_i)} + 1} \tag{9}$$

for $i = 1, 2, \cdots, n$.

Proof. Let $Av = \lambda v$ and let $L(x_i) = \{L_0(x_i), L_1(x_i), \cdots, L_{e(x_i)}\}$ be the level structure rooted at x_i. Furthermore, for brevity let

$$\sum_{Adj(x_i)} v_j \tag{10}$$

denote the sum of all v_j over all indices j, such that $x_j \in Adj(x_i)$, and similarly $\sum_{L_k(x_i)} v_j$, etc.

Now $Av = \lambda v$ implies that (for $n > 1$)

$$v_i = \frac{1}{\lambda - 1} \sum_{Adj(x_i)} v_j = \frac{1}{\lambda - 1} \sum_{L_1(x_i)} v_j, \quad i = 1, \cdots, n. \tag{11}$$

Substituting (11) into itself and taking into account that $x_i \in Adj(x_j)$ for $x_j \in L_1(x_i)$, we obtain

$$v_i \geq \frac{1}{(\lambda - 1)^2} \sum_{L_2(x_i)} v_j, \qquad i = 1, \cdots, n. \tag{12}$$

This process can be repeated $e(x_i)$ times so that we obtain

$$v_i \geq \frac{1}{(\lambda - 1)^k} \sum_{L_k(x_i)} v_j \qquad \text{for } i = 1, 2, \cdots, n \qquad \text{and } k = 1, 2, \cdots, e(x_i). \tag{13}$$

Summing up the $e(x_i)$ inequalities (13), it follows that

$$e(x_i)v_i \geq \sum_{k=1}^{e(x_i)} \frac{1}{(\lambda - 1)^k} \sum_{L_k(x_i)} v_j \geq \frac{1}{(\lambda - 1)^{e(x_i)}} \sum_{j=1; j \neq i}^{n} v_j = \frac{1 - v_i}{(\lambda - 1)^{e(x_i)}}. \tag{14}$$

Here $\lambda \geq 2$ was used, which is correct for $n > 1$, as mentioned after formula (3). Therefore

$$v_i \geq \frac{1}{e(x_i)(\lambda - 1)^{e(x_i)} + 1} \qquad \text{for } i = 1, 2, \cdots, n. \tag{15}$$

This is also correct for $n = 1$. □

PROPOSITION 2. *Let δ be the diameter of the graph. Then*

$$\lambda \geq 1 + \sqrt[\delta]{\frac{n-1}{\delta}}. \tag{16}$$

Proof. From (9) it follows that

$$v_i \geq \frac{1}{\delta(\lambda-1)^\delta + 1} \qquad \text{for } i = 1, 2, \cdots, n. \tag{17}$$

Summing up for $i = 1, 2, \cdots, n$ and rearranging yields the result. □

For the proof of Proposition 3, the following lemma is needed.

LEMMA 4.1. *Let $a_{ij}^{(k)}$ be the (i,j)th entry of the matrix $A^k, k = 1, 2, 3, \cdots$ and let $n > 1$. Then it holds that*

$$a_{ij}^{(k)} \geq 1 \qquad \text{for all } i, j \text{ with } d(x_i, x_j) = k \tag{18}$$

$$a_{ij}^{(k)} \geq k \qquad \text{for all } i, j \text{ with } d(x_i, x_j) < k. \tag{19}$$

Proof. Let $d(x_i, x_j) = p \leq k$. Now $a_{ij}^{(k)}$ counts the number of paths of length at most k steps from x_i to x_j. If we follow the shortest path and make exactly k steps, of which p go forward and $k - p$ stay at the same node (go around self-loops), there are $\binom{k}{p}$ possibilities for the choice of p forward steps. But $\binom{k}{p} \geq k$ if $1 \leq p < k$ giving (19) if $i \neq j$; the case $i = j$ is treated similarly, and $\binom{k}{p} = 1$ if $p = k$ giving (18). □

PROPOSITION 3.

$$v_i \geq \frac{1}{\lambda^{e(x_i)} + 1} \qquad \text{for } i = 1, \cdots, n. \tag{20}$$

Proof. Let $a_{ij}^{(k)}$ be the (i,j)th entry of A^k as before, and let D be the distance matrix of the graph, i.e., $D = (d_{ij})$, where $d_{ij} = d(x_i, x_j)$. Then the following statements about $a_{ij}^{(k)}$ and d_{ij} can be made for $k = 1, 2, \cdots$ using (18) and (19):

$$\left.\begin{matrix} a_{ij}^{(k)} \geq k \\ d_{ij} \geq 1 \end{matrix}\right\} \qquad \begin{matrix} \text{for all } i, j \text{ with } d_{ij} < k \text{ except} \\ \text{for the diagonal elements where } d_{ii} = 0 \end{matrix} \tag{21}$$

$$\left.\begin{matrix} a_{ij}^{(k)} \geq 1 \\ d_{ij} = k \end{matrix}\right\} \qquad \text{for all } i, j \text{ with } d_{ij} = k \tag{22}$$

$$\left.\begin{matrix} a_{ij}^{(k)} = 0 \\ d_{ij} \geq k+1 \end{matrix}\right\} \qquad \text{for all } i, j \text{ with } d_{ij} \geq k+1. \tag{23}$$

Taking (21) – (23) together in matrix form it holds that every element of the matrix $I + A^k + D$ is greater or equal to $k + 1$, where I is the $n \times n$ identity matrix. Let J be the $n \times n$ matrix with all entries equal to one. Then this fact can be written as

$$I + A^k + D \geq (k+1)J. \tag{24}$$

Therefore

$$v + A^k v + Dv \geq (k+1)Jv = (k+1)u, \tag{25}$$

where $u = (1, 1, \cdots, 1)^T$. The ith component of Dv can be bounded as follows:

$$(Dv)_i = \sum_{l=1}^{n} d_{il} v_l \leq e(x_i) \sum_{l=1}^{n} v_l = e(x_i). \tag{26}$$

TABLE 1
Eigenvector bounds for example graph.

i	v_i	$e(x_i)$	Lower bound from Prop. 1	Lower bound from Prop. 3
1	0.0364	4	0.00235	0.00317
2	0.1111	3	0.00998	0.01322
3	0.0756	4	0.00235	0.00317
4	0.0414	4	0.00235	0.00317
5	0.1144	3	0.00998	0.01322
6	0.1315	3	0.00998	0.01322
7	0.1330	3	0.00998	0.01322
8	0.1232	3	0.00998	0.01322
9	0.0871	3	0.00998	0.01322
10	0.1480	3	0.00998	0.01322

Using (25) and $A^k v = \lambda^k v$, one obtains for the components in (24)

$$v_i + \lambda^k v_i + e(x_i) \geq k + 1 \qquad \text{for } i = 1, 2, \cdots, n \quad \text{and} \quad k = 1, 2, 3, \cdots. \tag{27}$$

If k is chosen to be $e(x_i)$, then (20) follows. □

Note that the choice $k = e(x_i)$ in (27) makes the bounds the best possible, since $k < e(x_i)$ yields trivial bounds and $k > e(x_i)$ yields in general some worse bounds because of the rapidly growing denominator.

PROPOSITION 4.

$$\lambda \geq \sqrt[\delta]{n-1}. \tag{28}$$

Proof. Set $k = \delta$ in (27). Then

$$v_i \geq \frac{\delta + 1 - e(x_i)}{\lambda^\delta + 1} \geq \frac{1}{\lambda^\delta + 1} \qquad \text{for } i = 1, \cdots, n. \tag{29}$$

Summing over i and rearranging yields (28). □

All the bounds in the propositions above are rather weak. But this is to be expected, since they were proven for general graphs without any further assumptions. The bounds of Proposition 1 are better for some smaller graphs, whereas the bounds of Proposition 3 are better for larger graphs (for larger $e(x_i)$). It should also be noted that the bounds of Proposition 3 are almost sharp, if the graph is a clique. Therefore, there is not much hope to improve these bounds in all generality. However, all bounds on the components of the eigenvector show that there is an inverse relationship between eccentricity $e(x_i)$ and the corresponding v_i.

The weakness of the bounds can be seen in the following example. The graph is taken from [7], where it is also used to illustrate several algorithms and concepts. For this graph the figures in Table 1 were obtained. Clearly the bounds from Propositions 1 and 3 are an order of magnitude smaller than the corresponding components of the eigenvector.

Here $\delta = 4$ and $\lambda \approx 4.21$. The bounds of Propositions 2 and 4 yield

$$\begin{aligned} \lambda &\geq 1 + \sqrt[4]{\tfrac{9}{4}} \approx 2.225 \\ \lambda &\geq \sqrt[4]{9} \approx 1.732. \end{aligned} \tag{30}$$

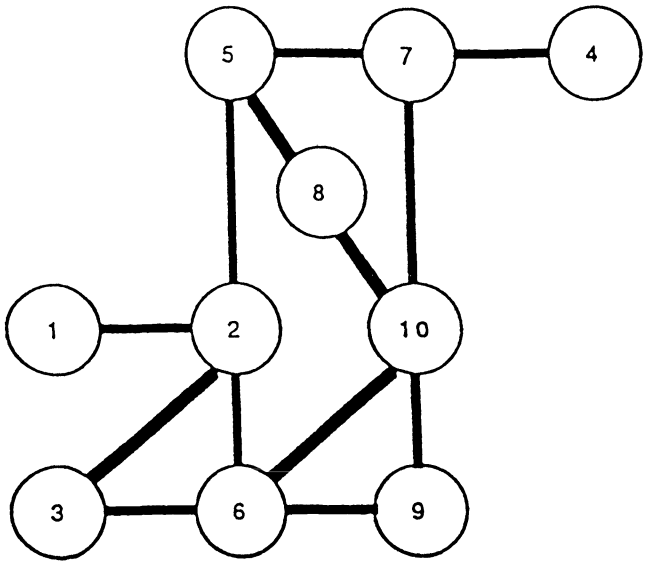


FIG. 2. *Example.*

5. Application to bandwidth and profile reduction for sparse matrices. In §2 we proposed a new algorithm for computing a pseudoperipheral node in a graph. Since this algorithm is only a heuristic, and since the term "pseudoperipheral" is only defined in the context of this heuristic algorithm, there is only one way to assess the efficacy of such an algorithm: to compare it to other algorithms in an application to a practical problem. The application of the new algorithm that we are most interested in is sparse matrix computations.

The solution of sparse linear equations of the form

$$Mx = b \tag{31}$$

by direct methods has been an area of intensive research during the past 15 years. For symmetric positive definite matrices most of the effort has been directed toward a combination of Gaussian elimination with some reordering of the equations and unknowns in (31). The goal is to obtain a permutation such that the solution of the permuted system incurs less fill-in than the solution of the original system. The actual numerical entries of M are irrelevant for this reordering phase, because if M is positive definite, then so is the permuted system, and a Cholesky factorization can always be computed. Thus the reordering can be based on the structural information for the matrix, i.e., the graph of the adjacency matrix alone. For a detailed discussion of the topic see the book by George and Liu [7].

Several reordering heuristics discussed in [7] ideally require the computation of a peripheral node. The practical Fortran implementation of these algorithms, however, relies on the Gibbs-Poole-Stockmeyer (GPS) algorithm, which computes a pseudoperipheral node. For most practical applications, this node is a good starting node for the reordering algorithms in SPARSPAK. In order to test the new pseudoperipheral

node finder, we replaced the subroutine FNROOT in SPARSPAK by a new subroutine which computed the dominant eigenvector of the adjacency matrix using the power method. Then the node corresponding to the component with the smallest entry in the eigenvector was used as a pseudoperipheral node. We could have used a more powerful algorithm such as the Lanczos algorithm, as is argued in [13]. But for our purposes here a few steps of the power method were sufficient, as our results in the next section will show.

The application of the power method is straightforward. The only question that remains to be discussed is a suitable stopping criterion. We are interested only in the location of the smallest entry of the dominant vector, possibly only in the location of a small, but not necessarily the smallest entry. The numerical results indicate that four steps of the power method were sufficient to obtain a pseudoperipheral node which was efficient for our sparse matrix applications. This number of iteration steps was also chosen in the implementation discussed in [9].

The new algorithm can be applied in the context of profile and bandwidth reduction algorithms for the reordering of sparse matrices. Recent research results [1], [11] indicate that sparse Gaussian elimination based on profile and bandwidth is no longer competitive with general sparse and multifrontal methods. However, band and envelope methods are widely used in applications in structural engineering, and are used in many software packages for engineers. For these applications the new algorithm is an alternative, since it does not require a general redesign of the package based on a new data structure for the sparse matrices. As in the case of [9] only one extra subroutine is required. Another potential application of the new algorithm is in the context of general sparse schemes, which sometimes require pseudoperipheral nodes as well, e.g., the automated nested dissection algorithm [7].

6. Numerical results. In Table 2 we summarize some characteristics of the sparse matrix test problems, which we used to evaluate the new heuristic algorithm. All test problems are available in the Boeing–Harwell sparse matrix collection and are described in [6]. Table 2 lists the problems, the number of equations (nodes), the number of nonzeros in the matrix (edges in the graph), and both profile and bandwidth of the unordered matrix. The first two examples are electric power networks. These are planar graphs, which correspond probably most closely to the model we had in mind, when developing the new algorithm. Problems 3 – 7 are finite-element models of three-dimensional structures. They are probably distinguished by the existence of many cliques. These examples are typical for the type of matrices encountered in structural engineering. The last three examples are finite-difference approximations to problems defined in very regular two-dimensional domains.

The matrices in Table 2 were first reordered with the reverse Cuthill–McKee (RCM) algorithm as implemented in [7] , and then reordered using the new eigenvector algorithm. In order to evaluate the change in efficiency in the reordering, we computed the smallest component of the iteration vector in the power method for each of the first 25 iterations of the power method, and then at each iteration step the resulting RCM ordering. In Table 3 we list the results of this numerical experiment. We give the best result obtained with the eigenvector method, and the number of iterations required to obtain this result. In most (but not all) cases more iterations of the power method did not change the results in Table 3.

Table 3 demonstrates that the node corresponding to the smallest component of the iteration vector in the power method is a suitable alternative as a pseudoperipheral node. The RCM method yields about the same reduction in profile and bandwidth

TABLE 2
Test matrices.

	Title	Equations	Nonzeros	Profile	Bandwidth
1	Western US Power Network	1,723	6,511	472,515	1,663
2	Entire US Power Network	5,300	21,842	6,122,200	5,189
3	TV Studio	1,074	12,960	240,161	590
4	Fluid Flow – Stiffness Matrix	2,003	83,883	434,798	1,250
5	Geodesic Dome	2,132	14,872	188,488	1,805
6	Cannes Matrix	1,072	12,444	277,248	1,048
7	Connection Table	2,680	25,026	587,863	2,499
8	9-Point Operator on 40 × 40 Grid	1,600	13,924	63,960	41
9	9-Point Operator on 80 × 80 Grid	6,400	56,644	511,920	81
10	George's L-shaped Problem	3,466	23,896	363,844	3,434

TABLE 3
Comparison with SPARSPAK RCM *for envelope reduction.*

	SPARSPAK RCM		Best power with RCM		Iter.	Time	Time
	Profile	Bandw.	Profile	Bandw.		RCM	power
1	79,260	133	74,251	130	4	0.18	0.44
2	667,245	285	626,863	274	9	0.66	3.10
3	282,999	704	246,776	640	1	0.22	0.20
4	502,907	546	522,640	411	2	1.08	1.86
5	171,437	105	172,712	101	8	0.30	1.94
6	56,438	178	75,409	248	1	0.24	0.14
7	102,983	69	105,058	69	15	0.50	5.96
8	81,497	79	81,497	79	1	0.55	0.46
9	666,997	159	666,997	159	1	2.25	1.86
10	158,546	62	158,546	62	1	0.48	0.32

TABLE 4
Comparison with SPARSPAK RCM *as a pseudoperipheral node finder.*

	Diameter	Periph. Nodes	SPARSPAK RCM		Best power with RCM	
			Node	Eccen.	Node	Eccen.
1	38	5	418	38	224	38
2	50	6	1436	50	92	48
3	9	4	1063	9	1	9
4	12	90	659	12	34	11
5	35	20	633	35	192	34
6	13	24	203	13	46	12
7	76	7	243	76	240	73
8	40	156	40	40	1	40
9	80	316	80	80	1	80
10	91	2	16	91	16	91

with either the SPARSPAK pseudoperipheral node as starting node or with the node delivered by our algorithm. The execution times (in seconds) for these numerical tests were obtained on a Sun 3/260 with a floating-point accelerator. The new method does require somewhat higher execution times; however, this additional overhead is insignificant when compared to actual numerical factorization times for these types of matrices (cf. [1], [11]).

Table 3 demonstrates that the eigenvector method is suitable for the intended sparse matrix application. The effectiveness of the eigenvector method for finding pseudoperipheral nodes is demonstrated in Table 4. For the graphs corresponding to the matrices in Table 2 we list the diameter, the number of peripheral nodes, and the nodes found by SPARSPAK RCM and the eigenvector method together with their eccentricity. The numbering of the nodes refers to the original ordering of the matrices as given in the sparse matrix test collection [6].

In Table 5 we summarize the reduction in profile obtained by using the SPARSPAK pseudoperipheral node, the node corresponding to the smallest component of the power method iteration vector after 4 steps, and the node corresponding to the smallest component of the dominant eigenvector. In addition, we list the envelope reduction obtained from the GPS algorithm and from the Gibbs–King (GK) algorithm as implemented by Lewis in [10]. Generally the Gibbs–King is known to obtain the best reduction in envelope size, usually at the cost of increasing the bandwidth.

TABLE 5
Profile reduction using SPARSPAK RCM, GK, GPS, *four iterations of the power method* (POW4), *and dominant eigenvector* (EIG).

	RCM	GK	GPS	POW4	EIG
1	0.17	0.14	0.15	0.16	0.18
2	0.11	0.09	0.09	0.16	0.13
3	1.18	0.80	0.87	1.27	1.27
4	1.16	0.97	1.07	1.20	1.30
5	0.91	0.89	0.92	0.94	0.92
6	0.20	0.18	0.27	0.36	0.20
7	0.18	0.16	0.17	0.25	0.18
8	1.27	1.00	1.00	1.27	1.27
9	1.30	1.00	1.00	1.30	1.30
10	0.43	0.43	0.43	0.43	0.43

Four iterations of the power method were used in [9], and Table 5 demonstrates that this is a reasonable choice. The node thus selected delivers a profile reduction comparable to the GPS node, at a cost which is slightly higher. Note that Table 5 lists the reduction in profile obtained, normalized so that the profile of the original matrix as given in [6] is one. Apparently Problems 3 – 5 are given in a reduced profile form already, since we are not able to obtain any improvements. All algorithms fail in the same way on the regular grid problems. If there is no reduction in the envelope size, GPS and GK are returning the original ordering.

These results demonstrate that the new pseudoperipheral node finder based on the dominant eigenvector, or the computationally more efficient algorithm based on a few steps of the power method, is an alternative to the GPS, GK, and SPARSPAK RCM algorithms. The figures in Table 5 indicate the better performance of GPS and GK on this test set. These are results with the unmodified versions of these

algorithms. We did not merge our eigenvector algorithm with GPS and GK in the same way as we combined it with SPARSPAK RCM. These tests were not carried out, since we expect to see very similar results.

Because of its simplicity the above algorithm has been implemented as an out-of-core alternative in a software package for solving linear systems arising in structural analysis [9]. The advantages of the new algorithm for a parallel implementation are clear, but have not yet been pursued by the authors. More fundamentally, we were able to exploit the algebraic properties of the adjacency matrix of a graph for computational purposes. That it is possible at all to utilize this information in order to uncover structural properties of the graph and the corresponding sparse matrix came as a surprise to us. We believe that spectral properties of the adjacency matrix have more potential use in sparse matrix computations beyond the ideas discussed here.

Acknowledgment. We would like to thank John Lewis for making several valuable suggestions for improving the manuscript, as well as for providing the numerical test results with the GPS and GK algorithms.

REFERENCES

[1] C. ASHCRAFT, R. GRIMES, J. LEWIS, B. PEYTON, AND H. SIMON, *Recent progress in sparse matrix methods for large linear systems*, Internat. J. Supercomput. Appl., 1 (1987), pp. 10 – 30.

[2] B. ASPVALL AND J. GILBERT, *Graph coloring using eigenvalue decomposition*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 526 – 538.

[3] E. CUTHILL AND J. MCKEE, *Reducing the bandwidth of sparse symmetric matrices*, in Proc. 24th National Conference of the Association of Computing Machinery, ACM Publications, 1969, p. P69.

[4] D. CVETKOVIC, M. DOOB, AND H. SACHS, *Spectra of Graphs*, Academic Press, New York, 1980.

[5] I. DUFF, A. ERISMAN, AND J. REID, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.

[6] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM TOMS, 15 (1989), pp. 1 – 14.

[7] A. GEORGE AND J. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[8] N. GIBBS, W. POOLE, AND P. STOCKMEYER, *An algorithm for reducing the bandwidth and profile of a sparse matrix*, SIAM J. Numer. Anal., 13 (1976), pp. 236 – 249.

[9] R. GRIMES AND D. PIERCE, *The implementation of three resequencing algorithms for* MSC/NASTRAN, Tech. Report ETA-TR-65, Boeing Computer Services, Seattle, WA, 1987.

[10] J. LEWIS, *Implementations of the Gibbs–Poole–Stockmeyer and Gibbs–King algorithms*, ACM Trans. Math. Software, 8 (1982), pp. 180 – 189.

[11] J. LEWIS AND H. SIMON, *The impact of hardware gather/scatter on sparse Gaussian elimination*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 304 – 311.

[12] J. K. PACHL, *Finding pseudoperipheral nodes in graphs*, J. Comput. System Sci., 29 (1984), pp. 48 – 53.

[13] B. PARLETT, H. SIMON, AND L. STRINGER, *Estimating the largest eigenvalue with the Lanczos algorithm*, Math. Comp., 38 (1982), pp. 153 – 165.

[14] H. D. SIMON, *Bounds for the dominant eigenvector of a graph*, Tech. Report, Dept. of Appl. Math., State University of New York, Stony Brook, NY, 1982.

[15] W. F. SMYTH, *Algorithms for the reduction of matrix bandwidth and profile*, J. Comp. Appl. Math., 12/13 (1985), pp. 551 – 561.

[16] P. D. STRAFFIN, *Linear algebra in geography: Eigenvector networks*, Math. Mag., 53 (1980), pp. 269 – 276.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 335–360, July 1990

AVERAGE-CASE STABILITY OF GAUSSIAN ELIMINATION*

LLOYD N. TREFETHEN† AND ROBERT S. SCHREIBER‡

Dedicated to the memory of Jim Wilkinson.

Abstract. Gaussian elimination with partial pivoting is unstable in the worst case: the "growth factor" can be as large as 2^{n-1}, where n is the matrix dimension, resulting in a loss of $n-1$ bits of precision. It is proposed that an average-case analysis can help explain why it is nevertheless stable in practice. The results presented begin with the observation that for many distributions of matrices, the matrix elements after the first few steps of elimination are approximately normally distributed. From here, with the aid of estimates from extreme value statistics, reasonably accurate predictions of the average magnitudes of elements, pivots, multipliers, and growth factors are derived. For various distributions of matrices with dimensions $n \leqq 1024$, the average growth factor (normalized by the standard deviation of the initial matrix elements) is within a few percent of $n^{2/3}$ for partial pivoting and approximately $n^{1/2}$ for complete pivoting. The average maximum element of the residual with both kinds of pivoting appears to be of magnitude $O(n)$, as compared with $O(n^{1/2})$ for QR factorization.

The experiments and analysis presented show that small multipliers alone are not enough to explain the average-case stability of Gaussian elimination; it is also important that the correction introduced in the remaining matrix at each elimination step is of rank 1. Because of this low-rank property, the signs of the elements and multipliers in Gaussian elimination are not independent, but are interrelated in such a way as to retard growth. By contrast, alternative pivoting strategies involving high-rank corrections are sometimes unstable even though the multipliers are small.

Key words. Gaussian elimination, stability, pivoting, growth factor, extreme values

AMS(MOS) subject classifications. 65F05, 65G05

Notation.

A	matrix in $\mathbb{R}^{n\times n}$,
σ_A	standard deviation of elements of A,
$A^{(k)}$	modified matrix before step k of elimination,
$\hat{A}^{(k)}$	modified matrix at step k after pivoting but before row operations,
$U = A^{(n)}$	final upper-triangular matrix,
$u_{kk} = \hat{a}_{kk}^{(k)}$	kth pivot,
m	$n+1-k$ (partial pivoting), $(n+1-k)^2$ (complete pivoting),
σ_k	standard deviation of elements $a_{ij}^{(k)}$ $(k \leqq i, j \leqq n)$,
π_k	average absolute value of pivots u_{kk},
μ_k	standard deviation of multipliers $\hat{a}_{ik}^{(k)}/\hat{a}_{kk}^{(k)}$ $(k < i \leqq n)$,
$\rho, \tilde{\rho}$	growth factor, growth factor normalized by σ_A,
$W(m)$	extreme value or "winner" function for normal random variables,
N	sample size,
$\langle\cdot\rangle$	expected value.

0. Introduction. At the beginning of the computer era, it was feared that Gaussian elimination would be an ineffective method for solving systems of linear equations. A paper by Hotelling in 1943 [19] predicted that in the solution of $n \times n$ systems of the form $A^TAx = b$, errors might be amplified by as much as 4^{n-1}, so that a "78-rowed matrix would need to be carried to no less than 46 places to insure even an approximate

* Received by the editors May 23, 1988; accepted for publication (in revised form) July 13, 1989.

† Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 (lnt@math.mit.edu). The research of this author was supported by an IBM Faculty Development Award and a National Science Foundation Presidential Young Investigator Award.

‡ Research Institute for Advanced Computer Science, Moffett Field, California 94035 (schreiber@riacs.edu). The research of this author was supported by Office of Naval Research contract N00014-86-K-0610, by U.S. Army Research Office grant DAAL03-86-K-0112, and by the Saxpy Computer Corporation.

accuracy in the first decimal place." Another paper by Bargmann, Montgomery, and von Neumann in 1946 [1] stated that "very little is known about the stability of the methods so far described, [but] what information there is tends to indicate that these methods are unstable and that rounding errors accumulate so seriously that the methods are impractical for large values of n."

By the early 1950s, computational experience had revealed that these fears were groundless, and Gaussian elimination with partial pivoting rapidly became the universal algorithm for solving general dense systems of linear equations. Progress was also made on the theoretical side by Turing [30], von Neumann and Goldstine [31], and especially Wilkinson [32], [33], whose elegant arguments based on condition numbers and backward error analysis shed light on every aspect of the elimination process. The result of these developments is that a widespread view among numerical analysts nowadays, thirty years later, is roughly that "Wilkinson proved that Hotelling's prediction was too pessimistic."

This view is not entirely accurate, however, for a fundamental gap in our understanding remains. When Gaussian elimination with partial pivoting is performed on an $n \times n$ matrix A, the result is a factorization $PA = LU$, where P is a permutation matrix, L is unit lower triangular, and U is upper triangular. Let $\bar{x}$ denote the solution of a linear system $Ax = b$ computed in floating-point arithmetic. Wilkinson proved that under reasonable assumptions, the relative error in $\bar{x}$ satisfies

$$\frac{\|\bar{x}-x\|_\infty}{\|x\|_\infty} \leqq 4n^2 \kappa_\infty(A) \rho \varepsilon, \tag{0.1}$$

where ε is the machine precision, $\kappa_\infty(A)$ is the condition number of A in the supremum norm, and ρ is the **growth factor**,

$$\rho = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}, \tag{0.2}$$

with $a_{ij}^{(k)}$ denoting the i, j element before the kth step of elimination [33]. (Results like (0.1) appear in various forms, with different definitions of ρ, norms, and polynomial factors; we have picked a representative one.) Unfortunately, ρ may be as large as 2^{n-1} (though no larger), as is proved by the simple example shown here for $n = 5$:

(0.3)

$$\underset{A}{\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}} = \underset{L}{\begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ -1 & -1 & -1 & 1 & \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}} \underset{U}{\begin{bmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & 1 & 8 \\ & & & & 16 \end{bmatrix}}.$$

It follows that unless (0.1) is highly pessimistic, Gaussian elimination will be useless for certain matrices. And so it is.[1]

Thus Gaussian elimination is unstable in the worst case; the improvement from Hotelling to Wilkinson is merely from 4^{n-1} to 2^{n-1}. Why, then, is it successful in practice? Indeed, partial pivoting is so reliable that most of the software in use today—including

[1] Thanks to the integer entries and unit diagonal elements, experiments with this matrix A sometimes reveal no instability. To be sure of seeing it, choose a right-hand side with negative as well as positive entries, or perturb the elements of A slightly in such a way that the pivot sequence is preserved.

LINPACK [8]—does not even bother to monitor pivot growth, although that would be a fail-safe method of guarding against instability.

We propose that a partial answer would be obtained if we could show that Gaussian elimination is stable *on average*. Average-case analysis has not been popular in numerical linear algebra, partly because of the obvious fact that the matrices encountered in practical problems are by no means random. Indeed, some researchers have expressed the opinion that Gaussian elimination is stable in practice precisely because the matrices that occur in practice are better behaved than if they were random.[2] The purpose of this paper is to argue the opposite opinion. We believe that Gaussian elimination is stable because the matrices encountered in practice *are* random, to a sufficient degree, and that the essential reason examples such as (0.3) do not cause trouble is that they occupy a negligible proportion of the space of matrices.

We began this project with the optimistic conjecture that Gaussian elimination is stable on average for a combination of two reasons:

(1) The magnitudes of the multipliers are on average much less than 1;

(2) The signs of the multipliers and elements are effectively random and tend to cancel.

Both of these hypotheses are readily translated into quantitative predictions, but when carried out, it was quickly found that the two of them, taken together, are not enough to explain experimental observations. In actuality, as will be discussed in § 6, average growth factors in Gaussian elimination exhibit a mild $n^{2/3}$ dependence on n, at least for $n \leqq 1024$, whereas (1) and (2) lead to a prediction on the order of $e^{n/4 \log n}$ (see eq. (5.4)). This paper can be viewed as an exploration of how (1) and (2) can be made precise, and modified where necessary, to explain this behavior. To summarize, §§ 2–4 show that hypothesis (1) is valid: simple estimates based on extreme value statistics give good predictions of observed multipliers, which are indeed on average small. The trouble lies in hypothesis (2), which must be corrected as follows:

(2′) The signs of the multipliers and elements are "better than random" from the point of view of cancellation.

For many distributions of matrices the multipliers and elements are *uncorrelated* in the sense that their covariances are zero—this follows from simple sign considerations—but they are not independent. On the contrary, there are relationships among them that conspire to retard growth. A tentative explanation of this phenomenon, together with a quantitative model of it, are proposed in § 5.

For a quick demonstration that the numbers produced by Gaussian elimination with pivoting are highly dependent, factor a random matrix A into $PA = LU$, i.e., $U = L^{-1}PA$, and you will find that $\|L^{-1}\|$ is reasonably small—33.2 in one experiment with $n = 256$. Now, randomize the signs of the elements of L and compute $\|L^{-1}\|$ again. It will be dramatically larger—in the same experiment, 2.7×10^8.

Our statistical arguments and numerical experiments indicate that for matrices that are random in various senses, both growth factors and computed residuals tend to be no

[2] For example, Gaussian elimination is particularly stable for ill-conditioned matrices, and some have suggested, with discretization of partial differential equations in mind, that its stability in practice comes about because most matrices arising in practice tend to be exceptionally ill-conditioned. However, this is not true; the average $n \times n$ matrix has condition number $O(n)$ or larger [11], [12], while the condition number for the standard discretization of Poisson's equation is $O(n)$ in two space dimensions and only $O(n^{2/3})$ in three dimensions. (See a similar remark on p. 460 of [1].) Even if it were true, this kind of argument could not explain the success of Gaussian elimination. If examples such as (0.3) were typical in the space of matrices, it would not be enough for *most* matrices arising in practice to be exceptionally well behaved; essentially *all* of them would have to be exceptional, which is highly implausible.

larger than $O(n)$ on average. This is true for matrices with elements drawn from a normal distribution and for various other classes of matrices too; in fact, the growth factors and residuals often depend only on the standard deviation of the initial matrix elements. Although the initial matrix elements may be far from normally distributed, a few steps of Gaussian elimination typically bring them toward that form. A more systematic summary of our results can be found in the final section.

An analogous problem of average- versus worst-case behavior—concerning speed rather than stability—appears in linear programming. The simplex method was invented in 1947, and it was soon recognized that the number of steps to convergence is usually small in practice, even though the worst-case behavior is exponential [20]. The problem of obtaining an average-case convergence result became well publicized beginning in 1963 [6, p. 160], and in recent years has been solved in various senses by Borgwardt, Smale, and others [3], [25], [26], [24].

The problem of stability of Gaussian elimination is an embarrassing theoretical gap at the heart of numerical analysis. We believe that it is also of practical importance. One reason is that pivoting conflicts with both sparsity preservation and parallelization, so that less stringent strategies such as threshold pivoting [10] and pairwise pivoting [27] are attracting increasing attention (see § 8). We can hardly assess these variants fully while our understanding of classical Gaussian elimination remains incomplete. A more basic reason is that as computers grow more powerful, n is getting bigger. Traditionally, polynomial factors like the n^2 term in (0.1) have been ignored as moderate in size and in any case generally pessimistic, but as n increases from 10^2 (Wilkinson?) to 10^3 (LINPACK?) to 10^4 (supercomputers?) to 10^6 (the year 2000?) and beyond, the need for a more quantitative understanding of stability will grow. Average-case modeling of error propagation is already a well-established tool, for example, in the study of fast Fourier transforms for digital signal processing [22].

We wish to acknowledge several previous experimental studies of the behavior of Gaussian elimination for random matrices and related matters: by Goodman and Moler [16] (reported also in the LINPACK manual [8]), by Birkhoff and Gulati [2], and by MacLeod [21], [34] who presents detailed statistics from Gaussian elimination applied to random matrices of dimensions $n \leqq 100$ with sample sizes 10,000. Higham and Higham have investigated general classes of matrices with large growth factors [18]. Many theoretical questions concerning eigenvalues and condition numbers of random matrices have recently been settled by Edelman [11], [12].

1. Preliminaries. Throughout this paper A denotes a real $n \times n$ matrix, and $A^{(k)}$, $1 \leqq k \leqq n$, is the modified matrix, with zeros below the diagonal in the first $k-1$ columns, that remains before the kth step of Gaussian elimination. The end result is an upper-triangular matrix $U = A^{(n)}$. We denote by $\hat{A}^{(k)}$ the intermediate matrix obtained after pivoting but before elimination at step k; thus the kth elimination step has the form

$$\text{Step } k: \quad A^{(k)} \to \hat{A}^{(k)} \to A^{(k+1)} \qquad (1 \leqq k \leqq n-1).$$

The i, j entries of $A^{(k)}$ and $\hat{A}^{(k)}$ are denoted by $a_{ij}^{(k)}$ and $\hat{a}_{ij}^{(k)}$, respectively, and $\hat{a}_{kk}^{(k)} = a_{kk}^{(n)} = u_{kk}$ is the kth pivot element.

The growth factor ρ of (0.2) is intimately connected with the pivots u_{kk}: for complete pivoting (rows and columns) $\rho = \max_k |u_{kk}|/|u_{11}|$ exactly, and for partial pivoting (rows only) the details are more complicated but large growth is again usually associated with large pivots. On the other hand, a constraint on the size of the pivots is provided by Hadamard's inequality,

$$\prod_{k=1}^{n} |u_{kk}| = |\det A| \leqq \prod_{k=1}^{n} \|a_k\|, \tag{1.1}$$

where $\|a_k\|$ is the 2-norm of the kth column of A. If A is $\sqrt{n}$ times an orthogonal matrix (the factor $\sqrt{n}$ being introduced to make the standard deviation of the elements equal to 1), then (1.1) becomes

$$\prod_{k=1}^{n} |u_{kk}| = (\sqrt{n})^n. \tag{1.2}$$

Similarly, if A is a random matrix with independent elements drawn from the standard normal distribution of mean 0 and standard deviation 1, a known result on expected determinants [15] gives

$$\left\langle \prod_{k=1}^{n} |u_{kk}| \right\rangle = \sqrt{n!} \sim (2\pi n)^{1/4} (\sqrt{n/e})^n. \tag{1.3}$$

(Here and throughout the paper, $\langle \cdot \rangle$ denotes the expected value.) These observations imply that so long as the pivots are reasonably uniform in magnitude, they must be of a modest size, comparable to $\sqrt{n}$. Large pivots can occur only if the pivots are highly nonuniform, as in (0.3).

Sections 2–5 of this paper are devoted to investigating, by statistical arguments and numerical experiments, the dependence on n and k of the following quantities:

$$\sigma_k = \langle (a_{ij}^{(k)})^2 \rangle^{1/2} \quad \text{standard deviation of elements } (k \leqq i,j \leqq n), \tag{1.4}$$

$$\pi_k = \langle |\hat{a}_{kk}^{(k)}| \rangle = \langle |u_{kk}| \rangle \quad \text{average absolute value of pivots,} \tag{1.5}$$

$$\mu_k = \langle (\hat{a}_{ik}^{(k)} / \hat{a}_{kk}^{(k)})^2 \rangle^{1/2} \quad \text{standard deviation of multipliers } (k < i \leqq n). \tag{1.6}$$

(In the definitions of σ_k and μ_k, i and j are any integers in the ranges indicated; for most distributions of matrices A of practical interest, symmetry implies that the statistics are independent of these indices.) We shall argue that for many distributions of matrices, σ_k and π_k grow slowly and steadily with k, never attaining very large values. Section 6 applies these results to investigate average growth factors and § 7 reports numerical experiments concerning average residuals.

Our experiments are based on eight classes of matrices:

normal	standard normal distribution of mean 0, variance 1,
[−1, 1]	uniform distribution on [−1, 1],
[0, 1]	uniform distribution on [0, 1],
{−1, 1}	discrete distribution with $p(-1) = p(1) = \frac{1}{2}$,
{0, 1}	discrete distribution with $p(0) = p(1) = \frac{1}{2}$,
symm.	symmetric matrices with elements from the standard normal dist.,
Toep.	Toeplitz matrices with elements from the standard normal dist.,
orth.	orthogonal matrices distributed by Haar measure.

In the first five cases, the elements of an individual matrix are independent samples from the distributions indicated, while the final three cases have dependent elements. The random orthogonal matrices are calculated by a sequence of Householder reflections as proposed by Stewart [28]; Haar measure is a name for the isotropic distribution of orthogonal matrices in which each column or row is uniformly distributed on the unit $(n-1)$-sphere.

In each of our experiments, matrices A of one or more dimensions n are selected at random from one of these classes, with the sample size N diminishing with n to keep the computing time within reasonable bounds. A typical set of dimensions and sample sizes are listed below, although for some of our experiments the samples were larger.

dimension n	2	4	8	16	32	64	128	256	512	1024
sample size N	4096	2048	1024	512	256	128	64	32	20	10

Our calculations have been carried out in single precision Fortran 77 on SUN workstations, IBM-compatible personal computers, a CRAY-2, and an Ardent Titan; no machine dependences were observed. Most of our experiments, but not all, have made use of the shuffled random number generators RAN1 and GASDEV in [23]. Plotting, data analysis, and hundreds of supporting tests of every kind were carried out with the superb matrix "workbench" program MATLAB, without whose powerful assistance a project of this kind would have been difficult indeed.

2. Elements. Our arguments begin with a fundamental observation: for many classes of matrices, the elements $a_{ij}^{(k)}$ at the kth step of Gaussian elimination tend to be normally distributed with mean 0. This statement is not exactly valid for $k > 1$, even if the elements of the initial matrix $A = A^{(1)}$ are themselves normally distributed, nor is it asymptotically valid in any limit such as k, $n \to \infty$, so far as we know, since the conditions of the central limit theorem are not satisfied by Gaussian elimination. Nevertheless, the hypothesis of normally distributed elements is often an excellent approximation, after the first few steps of elimination, even when the elements of A are not normally distributed.

Figure 2.1 provides evidence for this claim. For each half of the figure—partial and complete pivoting—1280 matrices of dimension 64 with normally distributed elements have been factored, and the elements $a_{ij}^{(k)}$ ($k \leqq i, j \leqq n$) in columns $k = 1, 9, 17, \cdots, 57$ accumulated in bins. The data are plotted as asterisks after being rescaled to have

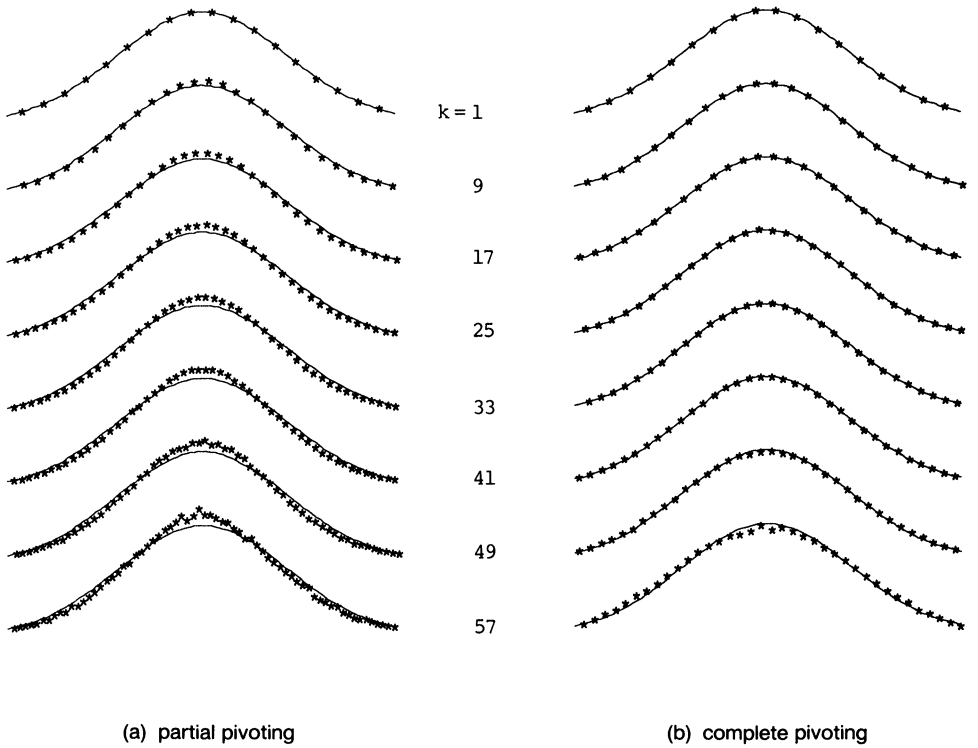


FIG. 2.1. *Distributions of elements* $a_{ij}^{(k)}$ *rescaled to have variance* 1, *for* $n = 64$: *observed* (*) *and normal distribution* (——).

standard deviation 1, and the solid curves show the normal distribution for comparison. The agreement of the two is excellent. (The noise toward the end results from the smaller numbers of elements in the samples.) It is not perfect, however; evidently partial pivoting leads to a distribution that is slightly more peaked in the center than the normal distribution. Similar plots are obtained for other values of n and k.

Although the shape of the element distribution is roughly independent of k, its standard deviation σ_k grows considerably. This is visible in the increasing density of asterisks in Fig. 2.1 as $k \to n$, especially for partial pivoting. (The bins holding the raw data before rescaling were equally spaced.) This dependence of σ_k on k is essentially the growth that is the subject of this paper. We shall model it in § 5.

When matrices from nonnormal distributions are investigated, the results are often surprisingly similar to those of Fig. 2.1. As a modest example, Fig. 2.2 details the initial steps of Gaussian elimination for matrices with elements from the uniform $[-1, 1]$ distribution. At $k = 1$, the asterisks reveal the initial square wave, but by $k = 8$, the distributions have become very close to normal, and for higher k (not shown) they are barely distinguishable from those of Fig. 2.1. The same phenomenon occurs with most of the classes of matrices listed in the last section. The exception is orthogonal matrices, for which the element distribution is approximately normal for much of the elimination but changes to a pronounced bimodal form toward the end.

From now on, then, we shall assume that at every step of elimination, the elements $a_{ij}^{(k)}$ are normally distributed with mean 0; only the standard deviation σ_k depends on k. For most of the argument, we shall further assume that the elements are independent, until we are forced to abandon that assumption at (5.5).

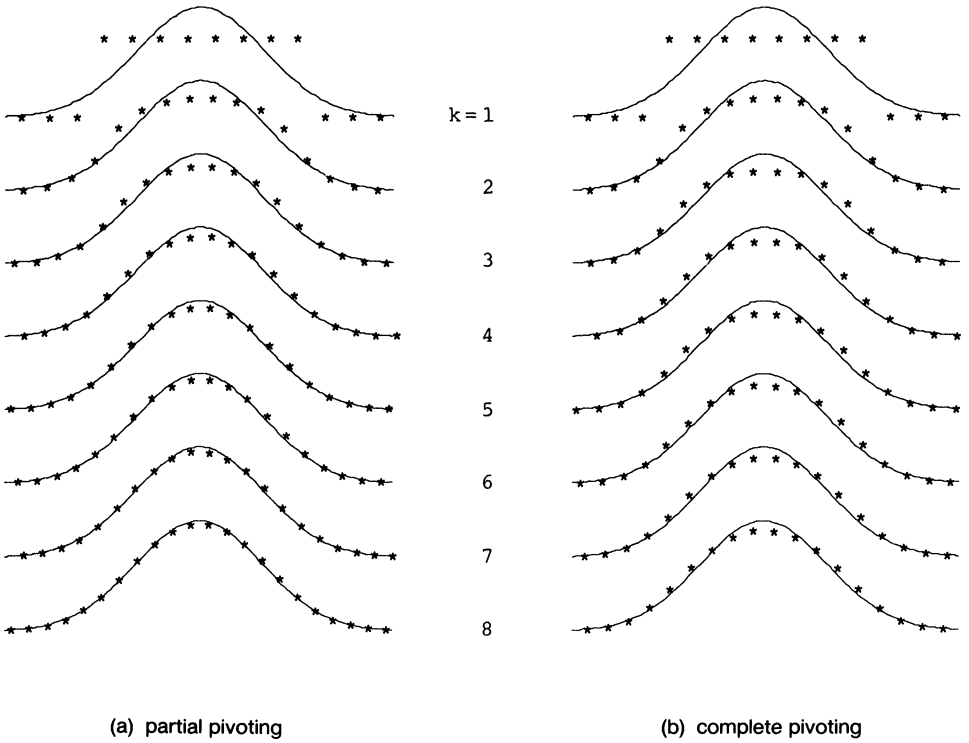


FIG. 2.2. *Similar to Fig.* 2.1, *but for matrices with elements uniformly distributed in* $[-1, 1]$. *Only the initial steps* $1 \leqq k \leqq 8$ *are shown.*

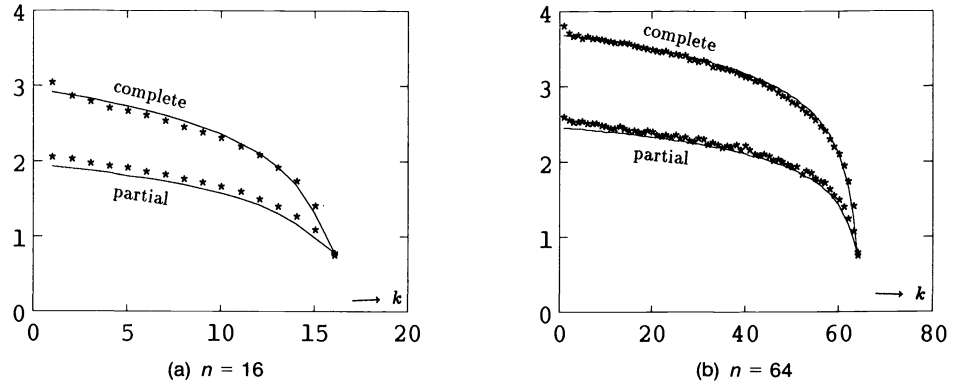


FIG. 3.1. *Average ratios* π_k/σ_k *of pivots to elements: observed* (*) *and predicted* (——).

Except where otherwise indicated, the experiments reported in the remainder of this paper are based on matrices A with elements from the standard normal distribution.

3. Pivots. Even without knowing σ_k, the standard deviation of the elements at the kth step of elimination, we can predict π_k/σ_k, the size of the average pivot relative to σ_k. The pivot element $u_{kk} = \hat{a}_{kk}^{(k)}$ is the largest in absolute value among m contestants, where

$$m = \begin{cases} n+1-k & \text{(partial pivoting)}, \\ (n+1-k)^2 & \text{(complete pivoting)}. \end{cases} \tag{3.1}$$

If the elements are normally distributed with standard deviation σ_k, the distribution of the pivots is a standard result from the field of the statistics of extreme values, going back to Tippett and Fisher in the 1920's [13], [17]. Let $W(m)$ (the "winner function") be defined as the mode of the distribution of the largest absolute value among m numbers taken from a normal distribution of mean 0, variance 1.[3] From equations 4.2.3(11, 15) of Gumbel [17], $W(m)$ is asymptotic to

$$\alpha := \sqrt{2 \log (m\sqrt{2/\pi})} \tag{3.2}$$

as $m \to \infty$, and a more accurate estimate is[4]

$$W(m) \approx \alpha\left(1 - \frac{2 \log \alpha}{1+\alpha^2}\right)^{1/2} + O\left(\frac{1}{\log m}\right). \tag{3.3}$$

(We define $W(1) = \sqrt{2/\pi}$, the expected absolute value of a single normal variate.) Thus our model of Gaussian elimination makes the prediction

$$\pi_k \approx \sigma_k W(m). \tag{3.4}$$

[3] We have chosen to work with the mode (the most frequent value) rather than the mean, although the two are asymptotic as $m \to \infty$. The reason is that the extreme value distribution is far from symmetric: for practical values of m the mode is several percent smaller than the median, which is several percent smaller than the mean. We shall be dividing by $W(m)$ to compute multipliers in the next section, and the mode is a convenient statistic that is relatively insensitive to this inversion.

[4] Gumbel has $m/2$ instead of m in (3.2), since he is concerned with the largest element in *signed* magnitude.

In the calculations and plots to follow we shall assume that each pivot element is exactly equal to $\pm\sigma_k W(m)$, although in actuality it is, of course, a random variable.

Figure 3.1 provides experimental confirmation of this prediction for matrices with normally distributed elements. For $n = 16$ and 64 and both partial and complete pivoting, the figure compares experimentally obtained ratios π_k/σ_k with the prediction $W(m)$, as a function of k. The agreement is not perfect, but it is quite good. Similar agreement is obtained with most of the other matrix distributions listed in § 1.

4. Multipliers. The previous two sections lead readily to a prediction of the distribution of multipliers at step k and of their standard deviation μ_k. First the pivot element $\hat{a}_{kk}^{(k)}$ is chosen and the rows and possibly columns permuted accordingly; we have assumed $\hat{a}_{kk}^{(k)}$ is equal to $\pm\sigma_k W(m)$. The multipliers are then the numbers $\hat{a}_{ik}^{(k)}/\hat{a}_{kk}^{(k)}$, and what we know about $\hat{a}_{ik}^{(k)}$ is that it comes from the normal distribution of mean 0, standard deviation σ_k, except with the tails beyond $\pm\sigma_k W(m)$ deleted and the total probability renormalized to compensate. That distribution has probability density function

$$\text{p.d.f.}\,(\hat{a}_{ik}^{(k)}) \approx \begin{cases} \dfrac{(1/\sqrt{2\pi})e^{-(x/\sigma_k)^2/2}}{\sigma_k \operatorname{erf}(W(m)/\sqrt{2})} & \text{for } |x| \leqq \sigma_k W(m), \\ 0 \quad \text{for } |x| > \sigma_k W(m), \end{cases}$$

where erf is the error function. (By (3.3) and standard estimates we have $1 - \operatorname{erf}(W(m)/\sqrt{2}) \sim W(m)/2m$ as $m \to \infty$.) The division by $\hat{a}_{kk}^{(k)} = \pm\sigma_k W(m)$ now gives the following approximate density distribution for the multipliers:

$$\text{p.d.f.}\left(\frac{\hat{a}_{ik}^{(k)}}{\hat{a}_{kk}^{(k)}}\right) \approx \begin{cases} \dfrac{(1/\sqrt{2\pi})W(m)e^{-(xW(m))^2/2}}{\operatorname{erf}(W(m)/\sqrt{2})} & \text{for } |x| \leqq 1, \\ 0 \quad \text{for } |x| > 1. \end{cases} \tag{4.1}$$

This is a rather remarkable formula, for it asserts that the multiplier distribution is independent of everything except the length of the column on and below the diagonal, $n + 1 - k$ (which determines m by (3.1)). From (4.1), by integration by parts, we can further derive an approximation for the variance of the multipliers at step k:

$$\mu_k^2 \approx \frac{1}{W(m)^2}\left(1 - \frac{\sqrt{2/\pi}W(m)e^{-W(m)^2/2}}{\operatorname{erf}(W(m)/\sqrt{2})}\right) \tag{4.2}$$

$$\sim \frac{1}{2\log(m\sqrt{2/\pi})}. \tag{4.3}$$

For experimental confirmation of these predictions, Fig. 4.1 is patterned after Fig. 2.1, but shows both $n = 16$ and $n = 128$. This time the solid reference curves are not simply rescaled Gaussians, but the predicted multiplier distributions (4.1). The agreement with predictions is excellent for partial pivoting and reasonably good for complete pivoting. Note that the multipliers are smaller for large n and for complete pivoting.

5. Dependence on k. Sections 2–4 have proposed models of the behavior of elements, pivots, and multipliers at each step k, but did not consider how the scale of these quantities—σ_k and π_k—changes with k. We turn now to this question.

The first half of step k is the interchange of rows and possibly columns $A^{(k)} \to \hat{A}^{(k)}$, which moves some large element $a_{ij}^{(k)}$ to the pivot position $\hat{a}_{kk}^{(k)}$. In the case of complete pivoting, this repeated removal of the largest element from the submatrix $k \leqq i, j \leqq n$ has a pronounced retarding effect on element growth, especially toward the end of the

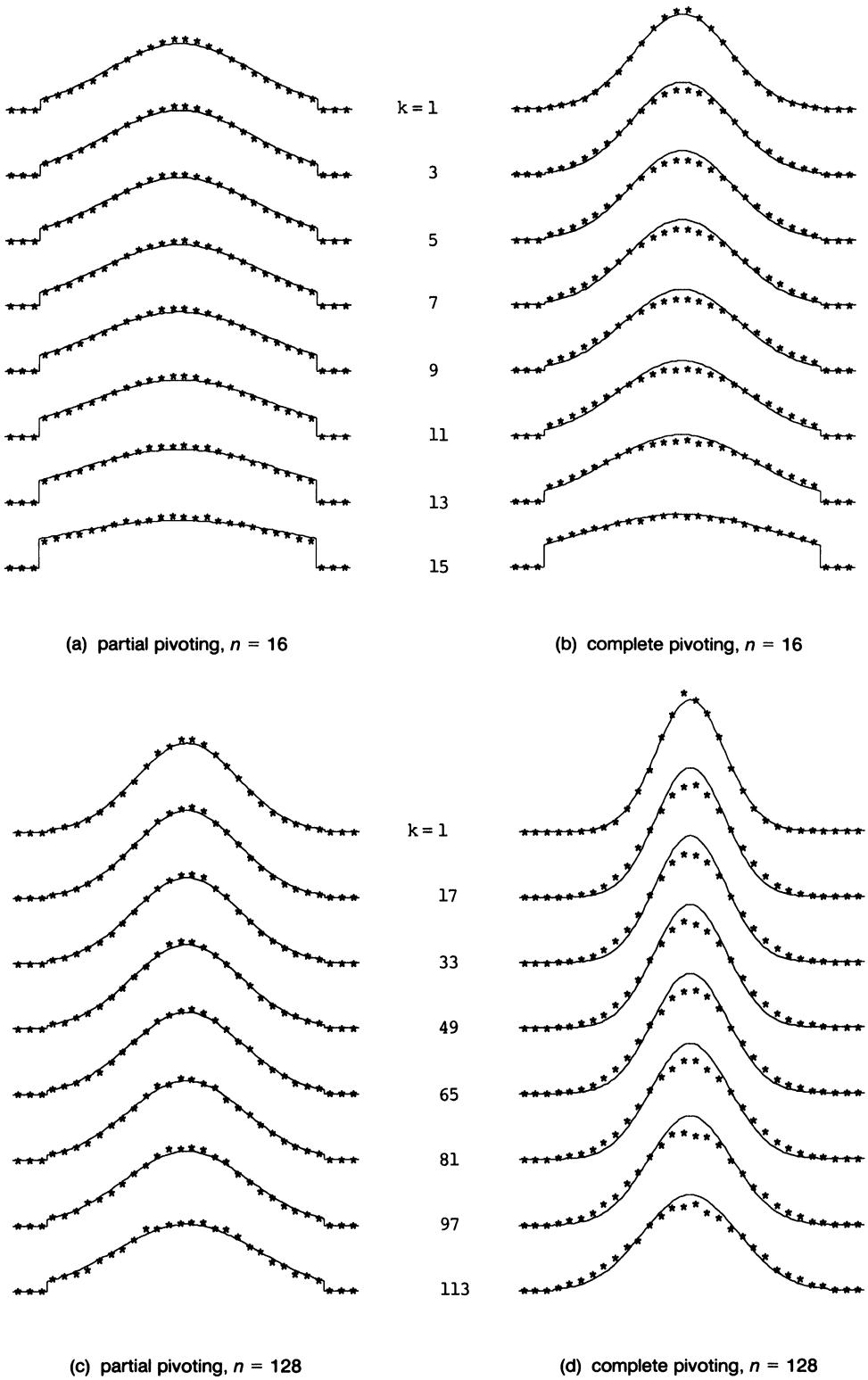


FIG. 4.1. *Distributions of multipliers* $\hat{a}_{ik}^{(k)}/\hat{a}_{kk}^{(k)}$: *observed* (*) *and predicted* (——). *The cutoff points are* ± 1.

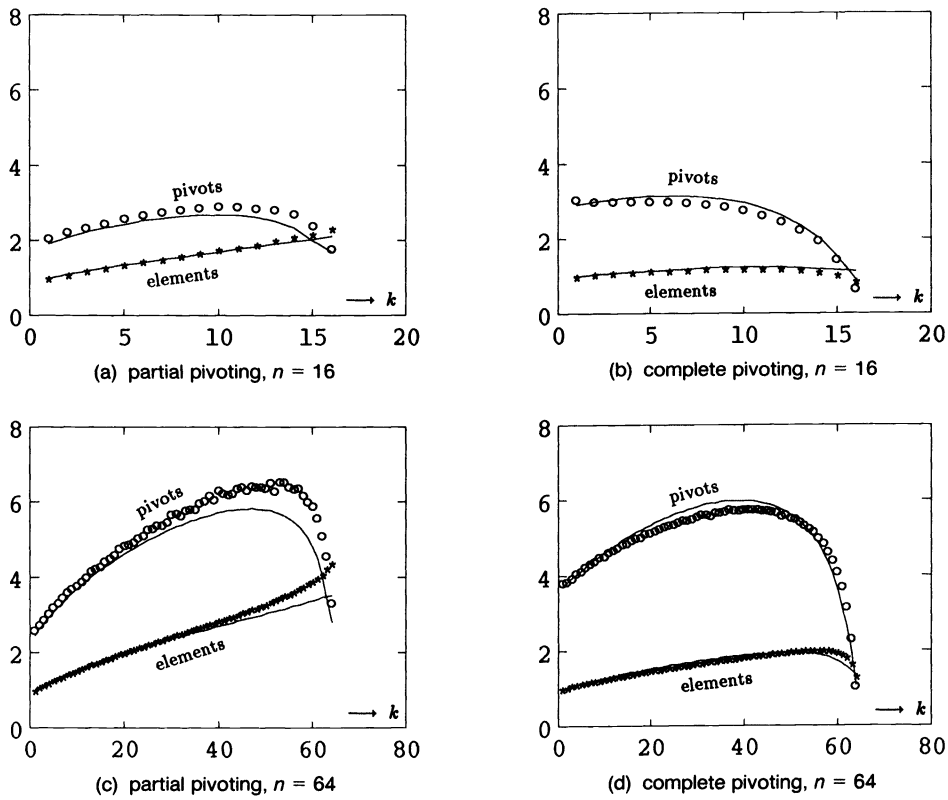


FIG. 5.1. *Average elements* σ_k *and pivots* π_k: *observed* (*, ○) *and predicted* (——).

elimination. At step k the elements $\alpha_{ij}^{(k)}$ with $k \leqq i, j \leqq n$ have variance σ_k^2; thus the expected sum of their squares is $m\sigma_k^2$ with $m = (n+1-k)^2$. When the pivot $\pm\sigma_k W(m)$ is removed from this collection, the remaining $m-1$ elements have expected sum of squares $(m - W(m)^2)\sigma_k^2$. If $\hat{\sigma}_k$ denotes the standard deviation of the elements $\hat{a}_{ij}^{(k)}$ for $k < i, j \leqq n$, we conclude

$$\hat{\sigma}_k^2 = \begin{cases} \sigma_k^2 & \text{(partial pivoting)}, \\ \sigma_k^2\left(\dfrac{m - W(m)^2}{m-1}\right) & \text{(complete pivoting)}. \end{cases} \tag{5.1}$$

The downturn resulting from this mechanism is clearly apparent in Figs. 5.1(b), 5.1(d).[5]

The second half of step k is the elimination calculation $\hat{A}^{(k)} \to A^{(k+1)}$,

$$a_{ij}^{(k+1)} := \hat{a}_{ij}^{(k)} - \frac{\hat{a}_{ik}^{(k)}}{\hat{a}_{kk}^{(k)}}\hat{a}_{kj}^{(k)} \qquad (k < i, j \leqq n). \tag{5.2}$$

By assumption $\hat{a}_{ij}^{(k)}$ and $\hat{a}_{kj}^{(k)}$ have variance $\hat{\sigma}_k^2$, and (4.2) gives a prediction of the vari-

[5] The growth-retarding mechanism just described is analogous to the cooling that occurs in evaporation of a liquid, in which the most energetic molecules escape from the surface, leaving those that remain behind a little less energetic on average. Further analogies can also be found between Gaussian elimination and statistical mechanics.

ance of $\hat{a}_{ik}^{(k)}/\hat{a}_{kk}^{(k)}$, denoted by μ_k^2. If all of these quantities were truly independent, (5.2) would imply that σ_{k+1}^2 was related to $\hat{\sigma}_k^2$ by

$$\sigma_{k+1}^2 = \hat{\sigma}_k^2 + \mu_k^2 \hat{\sigma}_k^2, \tag{5.3}$$

thus completing our model of Gaussian elimination. But this formula is utterly inaccurate: it leads to a prediction for partial pivoting of nearly exponential growth,

$$\frac{\sigma_n}{\sigma_1} \approx e^{n/(4 \log n)}, \tag{5.4}$$

which fails to match experiments except for $n \approx 1$. Equation (5.4) is derived by iterating (5.1) and (5.3),

$$\frac{\sigma_n^2}{\sigma_1^2} = \prod_{k=1}^{n-1} (1+\mu_k^2),$$

and then taking the logarithm and using (4.3) to obtain

$$\log \frac{\sigma_n}{\sigma_1} = \frac{1}{2} \sum_{k=1}^{n-1} \log (1+\mu_k^2) \sim \frac{1}{2} \sum_{k=1}^{n-1} \mu_k^2 = \frac{1}{2} \sum_{m=2}^{n} \frac{1}{2 \log m} \sim \frac{n}{4 \log n}.$$

We have now reached the point where hypothesis (2) mentioned in the Introduction has failed us; it is time to replace it by some quantitative version of (2′).

We have found that the following simple assumption is surprisingly accurate, at least until the last few steps of elimination: the variances σ_k^2 accumulate additively rather than multiplicatively according to the formula

$$\sigma_{k+1}^2 = \hat{\sigma}_k^2 + \mu_k^2 \hat{\sigma}_1^2. \tag{5.5}$$

We do not have a rigorous justification of why (5.5) is an appropriate replacement for (5.3), but here is a heuristic one. Equation (5.5) amounts to the statement that the operations performed in Gaussian elimination do not compound, from the point of view of growth factors; it is as if the kth elimination step were applied to the original matrix $A = A^{(1)}$ rather than to $A^{(k)}$. Why should this be? Our best answer is to describe the following mechanism, which suggests that the growth introduced at one elimination step tends not to contribute to further growth at later steps. At step k, the correction subtracted from $\hat{A}^{(k)}$ by (5.2) is a rank-1 matrix. Taking the extreme, suppose this correction happened to be much larger than the elements it was being added to. Then the new matrix $A^{(k+1)}$ would be close to a matrix of rank one in its lower-right subsquare $k+1 \leqq i, j \leqq n$. Consequently, the large numbers just introduced would vanish at step $k+1$.

This argument is certainly not complete, nor precise enough to distinguish (5.5) from various other possible modifications of (5.3). But we believe the feedback mechanism it describes is essential to the stability of Gaussian elimination: large growth makes the remaining matrix close to a matrix of low rank, which in turn inhibits large growth. Note that in keeping with the distinction in the Introduction between (2) and (2′), the low-rank property would be destroyed if the signs of the correction matrix were randomized. Experiments with a "lobotomized Gaussian elimination" algorithm of this kind confirm that (5.3) and (5.4) then become accurate. See § 8 for the occurrence of this instability phenomenon in a computation of practical interest based on "parallel pivoting."

Equation (5.5) completes our model of average element and pivot growth as a function of k, which consists in its entirety of equations (3.1), (3.2), (3.3), (3.4), (4.2), (5.1), and (5.5).

Figure 5.1 compares predicted element and pivot sizes with experimental observations. For all our inexact assumptions, the agreement is remarkably good except at the very end of the elimination. We emphasize that all of the solid curves in Fig. 5.1 represent predictions from general principles, dependent on no adjustable parameters.

Figure 5.2 returns once again to nonnormal distributions of elements. The first two plots repeat Figs. 5.1(c), 5.1(d) for matrices with elements from the $\{0, 1\}$ distribution. Of the nonorthogonal distributions we have considered, this is as far from having normally distributed elements as any, but even so, the figure reveals that our element and pivot predictions are roughly valid after $k \approx 8$. Similar but generally better agreement is observed in most of the other cases. Figures 5.2(c), 5.2(d), however, repeat the same experiments for random orthogonal matrices, and the results are very different. In keeping with (1.2) and (1.3), we see that the geometric mean of the pivots has increased by a factor of approximately $\sqrt{e}$. Hypothesis (5.5) has failed in this case, although the growth is still far less rapid than (5.3) would predict.

6. Growth factors. At last we are prepared to turn to the problem of average growth factors. We will begin with experiments, and then see how these can be related to the statistical model of the last four sections.

Figure 6.1 summarizes various theoretical and experimental results concerning the average growth factor $\langle \rho \rangle$ of (0.2), all plotted on a log-log scale. The highest curve shows the worst-case bound $\rho \leqq 2^{n-1}$ for partial pivoting, which we know by (0.3) is sharp.

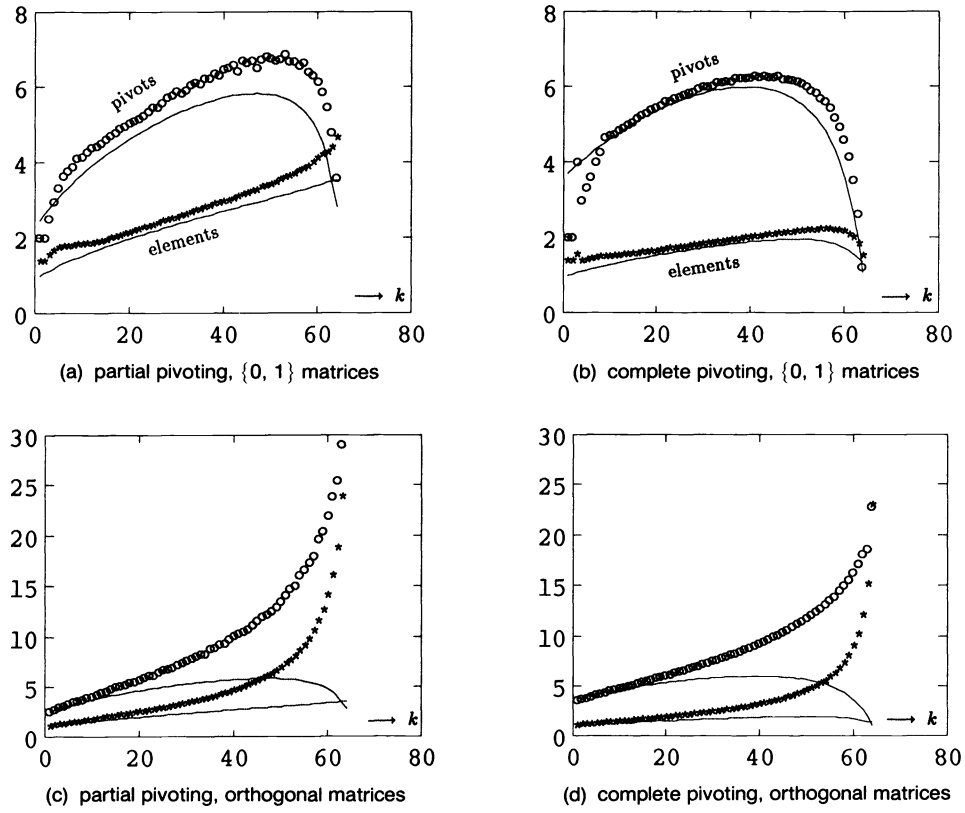


FIG. 5.2. *Repetition of Figs.* 5.1(c), 5.1(d) *for random* $\{0, 1\}$ *and orthogonal matrices,* $n = 64$.

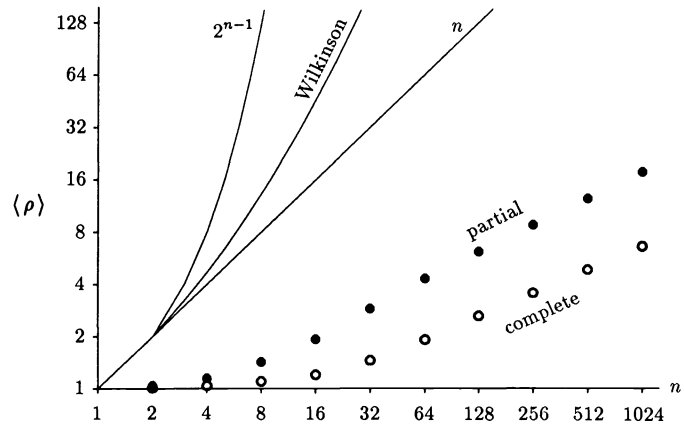


FIG. 6.1. *Average growth factors* $\langle \rho \rangle$. *The solid curves represent various theoretical worst-case bounds.*

The next curve shows the best available worst-case bound for complete pivoting, $\rho \leqq \sqrt{n}(2 \cdot 3^{1/2} \cdots n^{1/(n-1)})^{1/2} \sim Cn^{1/2 + 1/4 \log n}$, due to Wilkinson [32], which is known to be not sharp.[6] The straight line shows the bound $\rho \leqq n$ that was conjectured by Wilkinson for real matrices with complete pivoting [33, p. 213], which has never been proved except for $n \leqq 5$ [4], [7], [18]. Below these curves, we have plotted two sets of experimental values of $\langle \rho \rangle$ based on matrices with random elements from the standard normal distribution.[7] It is evident that the average growth factors for both partial and complete pivoting grow sublinearly with n and lie well below all of the worst-case bounds.[8]

The pattern in these data can be made more apparent if we modify the definition of ρ. Rather than dividing by the maximum element of A, let us divide by the standard deviation σ_A of the initial element distribution,

$$\tilde{\rho} = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\sigma_A}. \tag{6.1}$$

(σ_A is not the same as σ_1, unless the elements of A have mean 0: the former is a true standard deviation, while the latter is defined in (1.4) relative to 0.) For matrices with elements from a uniform distribution, this modification will increase $\langle \rho \rangle$ by a constant factor, whereas for matrices with normally distributed elements, the factor is approximately $W(n^2) = O(\sqrt{\log n})$. Figure 6.2 repeats the experimental data of Fig. 6.1, but showing $\langle \tilde{\rho} \rangle$ instead of $\langle \rho \rangle$. The data points lie strangely close to two straight lines:

$$\text{partial pivoting: } \langle \tilde{\rho} \rangle \approx n^{2/3}, \qquad \text{complete pivoting: } \langle \tilde{\rho} \rangle \approx n^{1/2}. \tag{6.2}$$

[6] The proof of Wilkinson's bound is a reasonably straightforward recursive application of (1.1).

[7] In Fig. 6.1, the last two data points in each sequence ($n = 512$ and $n = 1024$) are fabricated by extrapolation; in our computer experiments we neglected to measure these numbers beyond $n = 256$. All the data in Fig. 6.2 are genuine, however, and since the two figures are nearly equivalent, the extrapolations are unlikely to be far wrong, so we have included the extra points in Fig. 6.1 to make the comparison clearer.

[8] Goodman and Moler [8], [16] report that in the LU factorization of 10,000 random matrices of dimensions $10 \leqq n \leqq 50$ drawn from four different distributions, the largest growth factor encountered was $\rho \approx 23$. MacLeod [21], who increased n to 100, observed a maximum growth factor $\rho \approx 35$. We regret to say that in our own experiments, we were so focused on average-case behavior that we neglected to measure the largest growth factor.

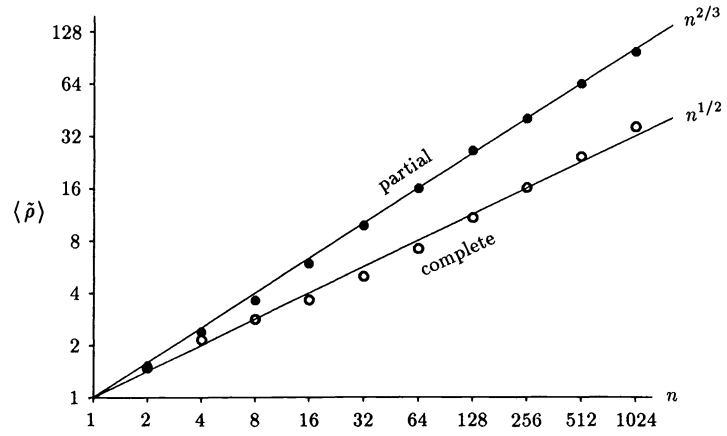


FIG. 6.2. *Average normalized growth factors* $\langle \tilde{\rho} \rangle$. *The solid lines are purely empirical.*

In this observation there is not even a constant factor to worry about—the fractional power of n is multiplied by 1! Despite these surprisingly close agreements, however—especially for partial pivoting and $n^{2/3}$—we do not claim that the approximations (6.2) are asymptotically valid as $n \to \infty$.

The data from Figs. 6.1 and 6.2 are recorded in Table 6.1. The sampling errors in this and subsequent tables probably range from about 1 percent for small n to more like 5 percent for large n.

Average growth factors change remarkably little when we turn to other distributions of matrices. Tables 6.2 and 6.3 list observed growth factors $\langle \tilde{\rho} \rangle$ for Gaussian elimination with partial and complete pivoting for the eight distributions of matrices listed in § 1. For larger n, except in the case of random orthogonal matrices, the numbers are nearly independent of the matrix distribution—so much so that a plot would be uninformative. Thus (6.2) appears to continue to hold with the constant factor 1, independently of the matrix distribution—a remarkable degree of a regularity that would have been obscured had we not normalized by σ_A in (6.1).

TABLE 6.1
Average growth factors $\langle \rho \rangle$ *and* $\langle \tilde{\rho} \rangle$.

	$\langle \rho \rangle$		$\langle \tilde{\rho} \rangle$	
n	Partial pivoting	Complete pivoting	Partial pivoting	Complete pivoting
2	1.04	1.01	1.52	1.48
4	1.15	1.04	2.39	2.15
8	1.42	1.10	3.63	2.82
16	1.93	1.20	5.92	3.64
32	2.89	1.45	9.77	4.97
64	4.31	1.91	15.9	7.17
128	6.14	2.62	26.3	10.8
256	8.74	3.56	40.0	16.1
512	-	-	63.7	24.3
1024	-	-	97.3	36.1

TABLE 6.2
Average growth factors $\langle\tilde{\rho}\rangle$ for matrices from various distributions, partial pivoting.

n	Normal	[−1, 1]	[0, 1]	{−1, 1}	{0, 1}	Symm.	Toep.	Orth.
2	1.52	1.48	2.77	1.50	1.87	1.40	1.41	1.59
4	2.39	2.23	3.33	2.02	2.13	2.26	2.12	2.78
8	3.63	3.50	4.06	3.61	4.18	3.60	3.37	5.20
16	5.92	5.85	6.17	6.63	6.86	6.06	5.83	10.3
32	9.77	9.67	9.85	9.86	9.91	10.0	10.3	20.7
64	15.9	15.5	16.3	15.7	16.6	16.5	18.3	42.5
128	26.3	24.6	25.2	24.9	25.9	25.6	30.1	81.4

The observation that orthogonal matrices fare worse in Gaussian elimination is not new, but goes back at least to Wilkinson (cf. Fig. 5.2). For example, the extreme case of element growth under complete pivoting in any example yet devised is achieved by Hadamard matrices—multiples of orthogonal matrices with elements ± 1—for which $\rho \geqq |u_{nn}| = n$ (proof by Cramer's rule [4]). See [7] and [18] for more on this subject.

It remains to relate these observations to our statistical model of the past four sections.

To begin the discussion, let us for the first time take a look at the effect of Gaussian elimination on individual matrices rather than just averages. Figures 6.3(a), 6.3(b) show pivots $|u_{kk}|$ from the factorization of a single matrix with $n = 64$ compared with the prediction π_k of Fig. 5.1. Figs. 6.3(c), 6.3(d) superimpose the pivots $|u_{kk}|$ from 25 such matrices. These four plots show vividly that our average-case predictions have definite relevance even to an individual matrix, for although $|u_{kk}|$ oscillates considerably, its overall trend follows the predicted average. They also show that the extent of the oscillation is much greater for partial than complete pivoting.

The growth factor $\tilde{\rho}$ will, in general, be larger than $\max_k \pi_k$, since $\tilde{\rho}$ is a maximum while $\max_k \pi_k$ is a maximum of an average. Figure 6.3 suggests that for complete pivoting the excess is typically modest, whereas for partial pivoting it may be quite substantial. These considerations explain how it is possible that the average size of π_k can be insensitive to the type of pivoting (in keeping with (1.3)) while the growth factor still varies significantly.

We can estimate $\langle\tilde{\rho}\rangle$ as follows. Figures 5.1 and 6.3 suggest that very roughly, the last $n/2$ steps of Gaussian elimination are equally likely to contribute the largest element $a_{ij}^{(k)}$. (The crudeness of this estimate is not so important, since $W(m)$ depends very weakly on m.) These final $n/2$ steps generate a total of $\sim n^3/24$ new elements $a_{ij}^{(k)}$. Therefore we estimate

$$\langle\tilde{\rho}\rangle \approx W(n^3/24) \max_k \sigma_k, \tag{6.3}$$

TABLE 6.3
Average growth factors $\langle\tilde{\rho}\rangle$ for matrices from various distributions, complete pivoting.

n	Normal	[−1, 1]	[0, 1]	{−1, 1}	{0, 1}	Symm.	Toep.	Orth.
2	1.48	1.42	2.77	1.50	1.87	1.38	1.39	1.59
4	2.15	1.98	3.27	2.02	2.14	2.06	1.97	2.63
8	2.82	2.75	3.50	3.40	3.76	2.83	2.74	4.30
16	3.64	3.70	4.13	4.10	4.15	3.73	3.84	7.26
32	4.97	5.11	5.34	5.48	5.55	5.10	5.72	13.0
64	7.17	7.40	7.49	7.73	7.88	7.34	8.94	23.3
128	10.8	11.0	11.2	11.3	11.4	11.0	13.9	43.3

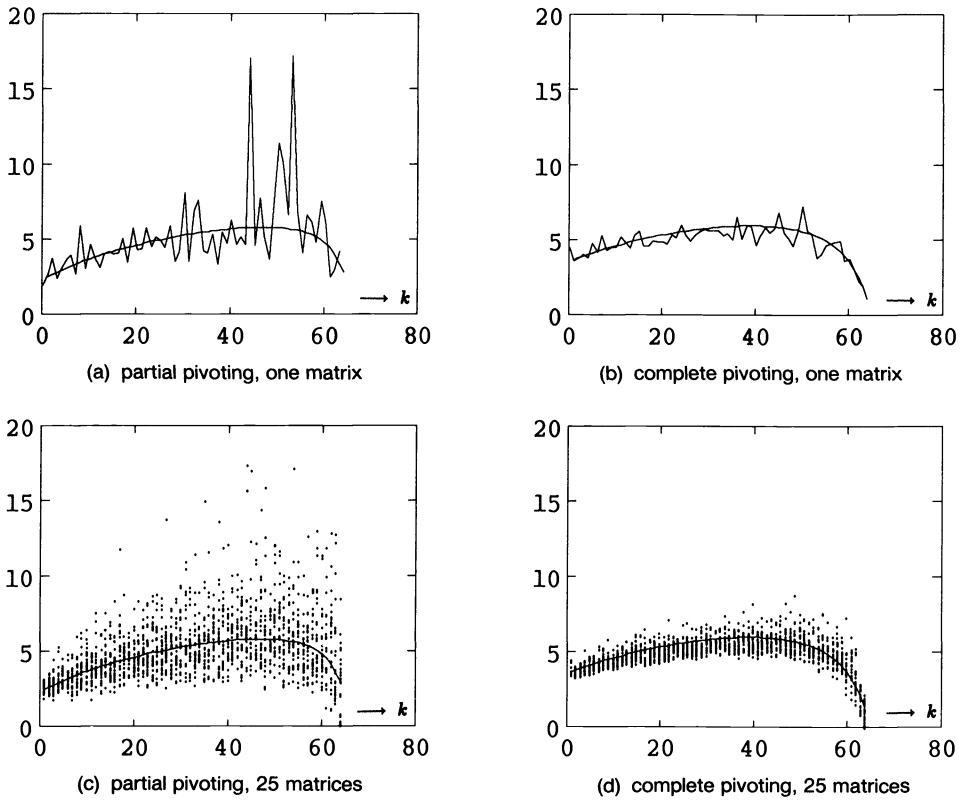


FIG. 6.3. *Pivots* $|u_{kk}|$ *for matrices from the standard normal distribution*, $n = 64$. *The solid lines represent the same predictions as in Figs.* 5.1(c), 5.1(d).

where σ_k is the predicted value derived in (5.5). Figure 6.4 compares this prediction with the lines $n^{1/2}$ and $n^{2/3}$ of Fig. 6.2. The agreement is not bad! The predictions for partial pivoting are somewhat too low, however, which reflects the fact that our predicted values of σ_k were too low toward the end of elimination (Fig. 5.1).

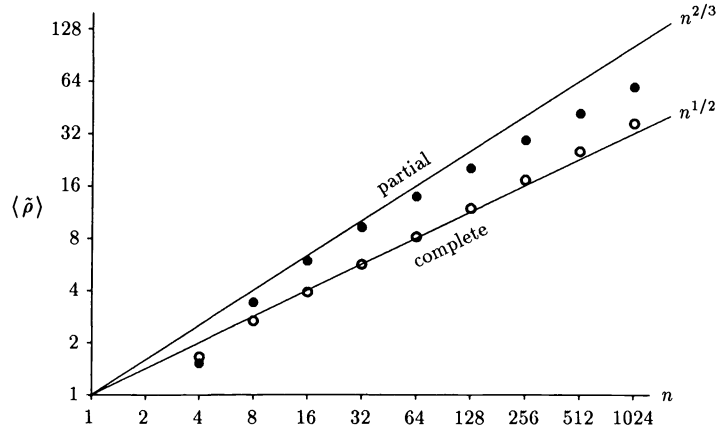


FIG. 6.4. *Predicted average growth factors* $\langle \tilde{\rho} \rangle$. *The solid lines are for comparison with Fig.* 6.2.

What about asymptotics as $n \to \infty$? We must be cautious here, for as is well known, extreme value statistics for normal distributions are approached painfully slowly. But as $n \to \infty$, $W(n^3/24) = O(\sqrt{\log n})$, by (3.2) and (3.3), and $\max_k \sigma_k = O(\sqrt{n}/\sqrt{\log n})$, by (4.2) and (5.5). Thus the natural conjecture appears to be

$$\langle \tilde{p} \rangle = O(\sqrt{n}) \quad \text{as } n \to \infty? \tag{6.4}$$

for both partial pivoting and complete pivoting, despite (6.2). This guess is tidy but hardly astonishing, in the light of (1.3).

7. Residuals. Our next set of experiments concerns the actual errors introduced by Gaussian elimination, and also by QR factorization, as measured by residuals computed in double precision. Let an $n \times n$ matrix A be factored in one of the following ways:

$A = PLU$ (Gaussian elimination with partial pivoting),
$A = P_1 L U P_2$ (Gaussian elimination with complete pivoting),
$A = QR$ (QR factorization),
$A = QRP$ (QR factorization with column pivoting),

where L is unit lower triangular, U and R are upper triangular, Q is orthogonal, and P, P_1, and P_2 are permutation matrices. The QR factorizations are carried out by Householder reflections, and as is customary, the vector associated with these reflections is stored rather than an explicit matrix Q. Let $\bar{L}$, $\bar{U}$, $\bar{R}$, and so on denote the matrices obtained in floating-point arithmetic, and define the *residual* for Gaussian elimination with partial pivoting by $E = A - \bar{P}\bar{L}\bar{U}$, and similarly for the other factorizations.

After a factorization has been carried out, we measure the size of E by its maximum element normalized by σ_A and also by machine epsilon:

$$E_{\max} = \frac{\max_{i,j} |e_{ij}|}{\sigma_A \varepsilon}. \tag{7.1}$$

At the end of a series of N factorizations, we compute the average $\langle E_{\max} \rangle$ as usual.

Figure 7.1 begins with the most important case of the standard normal distribution, showing computed quantities $\langle E_{\max} \rangle$ as a function of n for Gaussian elimination with

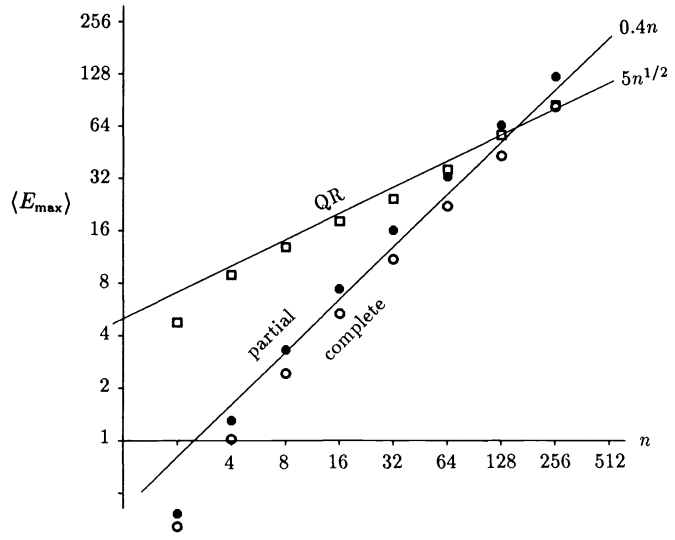


FIG. 7.1. *Average maximum residual element* $\langle E_{\max} \rangle$. *The solid lines are empirical.*

partial and complete pivoting and for QR factorization without pivoting. The data from the figure, together with corresponding numbers for QR factorization with column pivoting, are listed in Table 7.1. They suggest, somewhat surprisingly in the light of the last section, that the maximum residual elements in partial and complete pivoting differ only by a constant factor: both satisfy $\langle E_{\max} \rangle \approx Cn$. They also suggest that QR factorization is asymptotically more stable than either form of Gaussian elimination, with $\langle E_{\max} \rangle \approx Cn^{1/2}$. Thus it would appear that in practice, there may be little difference between partial and complete pivoting in Gaussian elimination, but not because both are entirely stable; apparently both suffer mildly from the unstable effects of pivot growth. The same conclusion is obtained if one measures the residual matrix E in ways other than by its maximum element.

TABLE 7.1
Average maximum residual element $\langle E_{\max} \rangle$.

	Gaussian elim.		QR factorization	
n	Partial pivoting	Complete pivoting	No pivoting	Column pivoting
2	0.43	0.33	4.75	5.10
4	1.47	1.13	8.87	8.42
8	3.64	2.68	12.8	12.5
16	8.13	5.71	18.0	16.7
32	17.2	11.4	24.2	24.9
64	36.0	22.6	35.8	35.6
128	73.0	44.6	56.4	54.6
256	134.	86.2	84.3	-

TABLE 7.2
$\langle E_{\max} \rangle$—*various matrix distributions, partial pivoting.*

n	Normal	[−1, 1]	[0, 1]	{−1, 1}	{0, 1}	Symm.	Toep.	Orth.
2	0.43	0.47	0.75	0.00	0.00	0.46	0.46	0.58
4	1.47	1.52	1.92	0.00	0.00	1.45	1.45	2.01
8	3.64	3.67	4.24	0.12	0.27	3.71	3.63	5.08
16	8.13	8.03	8.80	5.62	6.32	8.33	8.35	12.2
32	17.2	17.3	17.5	16.9	17.6	17.2	17.9	28.2
64	36.0	33.6	35.1	34.5	35.5	36.3	37.6	68.7
128	73.0	71.6	72.1	69.1	71.5	72.2	74.1	149.

TABLE 7.3
$\langle E_{\max} \rangle$—*various matrix distributions, complete pivoting.*

n	Normal	[−1, 1]	[0, 1]	{−1, 1}	{0, 1}	Symm.	Toep.	Orth.
2	0.33	0.42	0.68	0.00	0.00	0.40	0.41	0.58
4	1.13	1.32	1.75	0.00	0.00	1.19	1.27	1.78
8	2.68	2.98	3.61	0.28	0.64	2.79	3.00	4.39
16	5.71	6.03	6.82	4.86	5.34	5.85	6.25	9.25
32	11.4	11.5	12.4	11.5	11.9	11.4	12.7	19.9
64	22.6	22.1	24.2	23.9	24.4	22.9	26.2	42.0
128	44.6	46.5	43.3	44.8	46.8	43.5	50.9	88.2

As with the growth factors of the last section, our observations concerning residuals are closely duplicated for random matrices from many other distributions. Tables 7.2 and 7.3 reveal that once again, only orthogonal matrices among the classes we have examined behave much differently.

8. Alternative pivoting strategies. The previous sections have examined "classical" Gaussian elimination with partial or complete pivoting, and concluded that these algorithms are highly stable on average. In this final section we shall look more superficially at three variants of Gaussian elimination based on alternative pivoting strategies: "threshold," "pairwise," and "parallel" pivoting. All of these variants are less stable than partial or complete pivoting, and the last turns out to be markedly unstable for large n even though the multipliers are all less than 1 in magnitude. Table 8.1 summarizes our conclusions, which are based on experiments with $n \leqq 1024$. The most interesting observation is that as discussed in earlier sections, the stability of Gaussian elimination depends not only on the size of the multipliers, but also on whether the corrections introduced at each step are of low rank.

TABLE 8.1
Summary of experimental results for various pivoting strategies.

Pivoting strategy	Size of multipliers	Rank of corrections	Average-case stability
partial or complete	$\leqq 1$	1	highly stable
threshold	$\leqq \tau^{-1}$	1	reasonably stable for larger τ
pairwise	$\leqq 1$	low	reasonably stable
parallel	$\leqq 1$	$n/2$	unstable

We begin with *threshold pivoting*, a well-known idea that is discussed by various authors (for a discussion and references see [10]). The idea is to require only that

$$|\hat{a}_{kk}^{(k)}| \geqq \tau |\hat{a}_{ik}^{(k)}|, \qquad i > k, \tag{8.1}$$

where $\tau \in [0, 1]$ is a parameter. For $\tau = 1$ this is partial pivoting, and for $\tau = 0$ it is no pivoting at all; of course in practice τ is taken to be positive. The motivation behind threshold pivoting is that it allows for more than one row to be a candidate for the pivot row, and some other criterion, such as sparsity, can be used to make the choice. With this strategy, the multipliers are at most τ^{-1} and the growth factor satisfies $\rho \leqq (1 + \tau^{-1})^{n-1}$. As with partial or complete pivoting, each step involves an elimination operation of rank 1.

Several authors have espoused ways to choose τ, with recommended choices being as low as 0.01 [29] or as high as 0.25 [5]. Duff [9], [10] reports an experiment with four sparse matrices and arrives at the interesting conclusion that $\tau = 0.1$ affords both good reduction of fill-in and loss of only one to two digits of accuracy in the solution, whereas smaller τ (0.01 or less) can be disastrous to accuracy and may actually increase the fill-in. To explain this counterintuitive observation, he notes that when τ is small the variance of elements becomes large, so that the number of elements that satisfy (8.1) becomes small.

We performed a brief series of experiments using dense matrices of dimensions $n \leqq 128$, with independent elements from the standard normal distribution and with $\tau \in \{0.5, 0.25, 0.1, 10^{-2}, 10^{-4}, 10^{-8}\}$. The sample sizes were approximately as listed in § 1, and at each step the pivot row was simply taken to be the first candidate satisfying

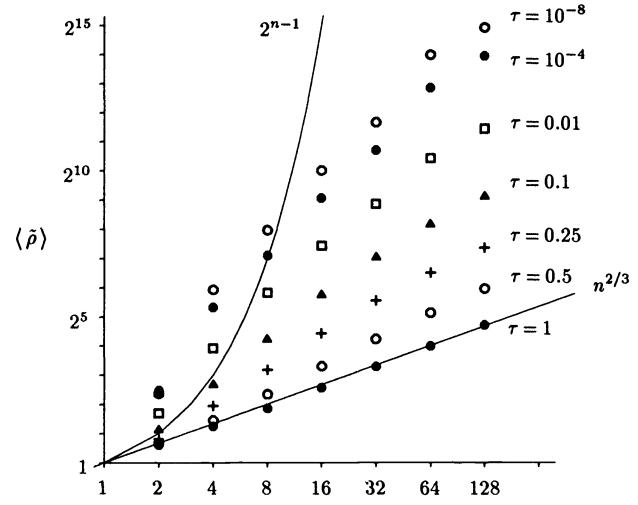


FIG. 8.1. *Average growth factors* $\langle \tilde{\rho} \rangle$ *for threshold pivoting with various thresholds* τ. *The solid lines are for comparison with earlier figures.*

(8.1). In Fig. 8.1, the observed average growth factor $\langle \tilde{\rho} \rangle$ is plotted against n for each of the values of τ, with the curves $n^{2/3}$ and 2^{n-1} shown for comparison. The numbers are listed in Table 8.2. These experiments support the conclusion that for larger values of τ, threshold pivoting is reasonably safe; the growth factors are nowhere near the worst-case bound 2^{n-1}. Of course, in applications involving sparse matrices the behavior may be different.

What about the limit $\tau = 0$—no pivoting? The data for $\tau = 10^{-8}$ in Fig. 8.1 are not much different from what would have been observed in the same experiment with $\tau = 0$, but there is an important mathematical difference nonetheless: although any single experiment will yield a finite result with probability 1, the expected growth factor is infinite in the absence of pivoting. (This is obvious; we need only consider the very first division $a_{21}^{(1)}/a_{11}^{(1)}$.) For a meaningful theory of the statistical behavior of Gaussian elimination without pivoting, we would have to employ a different measure of average-case growth such as $\exp(\langle \log \tilde{\rho} \rangle)$, as in the study of expected condition numbers [11], [26].

Another well-known variant of Gaussian elimination is *pairwise* or *neighbor pivoting*, in which only adjacent rows are interchanged or eliminated. Here is the algorithm. The scope of each control structure is indicated by indentation, and *row* (i) denotes the

TABLE 8.2
Average growth factors $\langle \tilde{\rho} \rangle$ *for threshold pivoting.*

n	$\tau = 0.5$	$\tau = 0.25$	$\tau = 10^{-1}$	$\tau = 10^{-2}$	$\tau = 10^{-4}$	$\tau = 10^{-8}$
2	1.58	1.75	2.06	3.24	5.54	5.16
4	2.74	3.86	6.07	15.1	39.9	60.7
8	5.07	9.05	17.7	56.1	136.	249.
16	9.86	21.6	50.9	172.	535.	1030.
32	18.7	46.6	124.	464.	1660.	3210.
64	34.8	90.6	270.	1370.	7340.	16100.
128	62.2	164.	523.	2670.	15600.	30900.

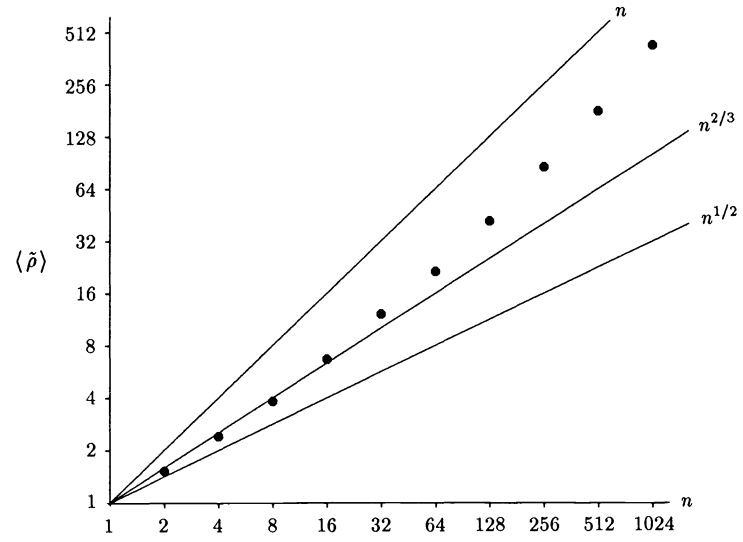


FIG. 8.2. *Average growth factors* $\langle \tilde{\rho} \rangle$ *for pairwise pivoting. The solid lines are for comparison with earlier figures.*

elements $\{a_{i,k}, a_{i,k+1}, \cdots, a_{i,n}\}$:

for $k := 1$ **to** $n - 1$
  for $i := n$ **to** $k + 1$ **step** -1
    if $|a_{i,k}| > |a_{i-1,k}|$ **then**
      exchange *row* (i) and *row* $(i - 1)$
    row $(i) :=$ *row* $(i) - (a_{i,k}/a_{i-1,k}) *$ *row* $(i - 1)$

This algorithm is of interest for parallel computing because it avoids the search for a maximum required by partial and complete pivoting—a substantial bottleneck for parallel computations—yet keeps all multipliers less than 1 in magnitude. Sorensen has obtained a worst-case bound 4^{n-1} on the growth factor [27].

Our experiments involved matrices of dimensions $n = 2, 4, \cdots, 1024$ with independent elements from the standard normal distribution. The sample sizes were 10,000 for $n \leqq 8$, 1000 for $16 \leqq n \leqq 128$, and 250 for $n \geqq 512$. Figure 8.2 plots the observed

TABLE 8.3
Average growth factors $\langle \tilde{\rho} \rangle$ *for complete, partial, pairwise, and parallel pivoting.*

n	Complete pivoting	Partial pivoting	Pairwise pivoting	Parallel pivoting
2	1.48	1.52	1.52	1.55
4	2.15	2.39	2.41	2.41
8	2.82	3.63	3.83	3.94
16	3.64	5.92	6.68	7.67
32	4.97	9.77	12.1	18.7
64	7.17	15.9	21.3	64.1
128	10.8	26.3	41.8	483.
256	16.1	40.0	85.2	1.53×10^4
512	24.3	63.7	179.	7.72×10^6
1024	36.1	97.3	432.	4.2×10^{12}

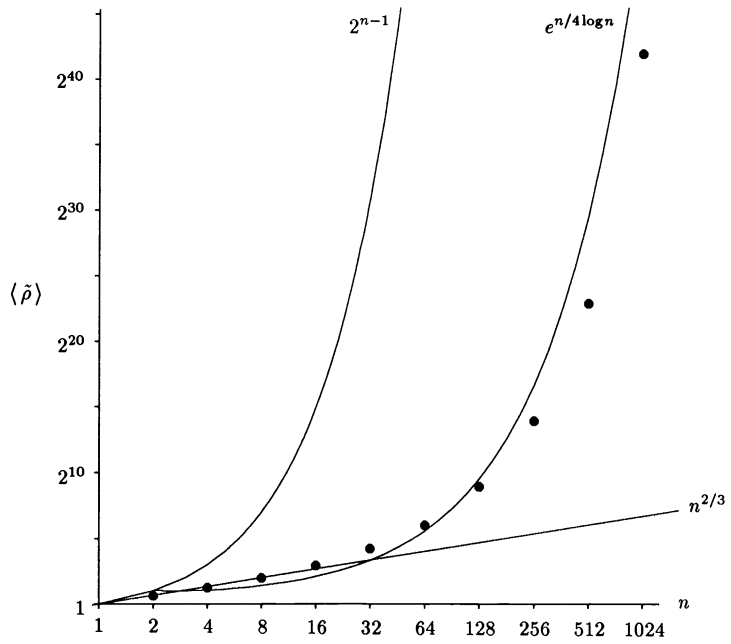


FIG. 8.3. *Average growth factors* $\langle\tilde{\rho}\rangle$ *for parallel pivoting.*

average growth factors $\langle\tilde{\rho}\rangle$ as a function of n, and the numbers are listed in Table 8.3. Evidently pairwise pivoting is quite stable on average, though not as stable as partial or complete pivoting. We explain this by observing that first, the magnitudes of the multipliers are somewhat bigger but still much less than 1 on average; second, the corrections introduced at each step are still on average of low rank, although not of rank 1.

Finally, what we call *parallel pivoting* is a (nonstandard) variant of Gaussian elimination in which as many as $n/2$ elements are eliminated in parallel. For example, if $n = 2m$, we first eliminate $a_{j+m,1}$ for $j = 1, 2, \cdots, m$ by subtracting a multiple of row j from row $j + m$. These two rows are exchanged first if necessary in order to keep the multiplier no greater in magnitude than 1. Here is the algorithm:

```
for k := 1 to n − 1
    nelts := n − k                          /* number below diagonal */
    while nelts > 0
        n2 := (nelts + 1)/2                 /* number to eliminate */
        nelts := nelts − n2                 /* number remaining */
        for i := 1 to n2
            row1 := k + i − 1               /* pivot row */
            row2 := k + i + nelts
            if |a_{row2,k}| > |a_{row1,k}| then exchange row (row1) and row (row2)
            row (row2) := row (row2) − (a_{row2,k}/a_{row1,k}) * row (row1)
```

We tested this algorithm on matrices of orders $n = 2, 4, 8, \cdots, 1024$ from the standard normal distribution. Except for $n = 1024$, where only two matrices were factored, the sample sizes were at least 100 matrices. Figure 8.3 plots the observed growth factors $\langle\tilde{\rho}\rangle$ as a function of n, together with the curves $n^{2/3}$, $e^{n/(4\log n)}$ (equation (5.4)), and 2^{n-1} for comparison. The data were listed already in Table 8.3 above. Clearly, this parallel pivoting strategy is unstable. We explain this by observing that first, the multipliers

are bigger than in standard Gaussian elimination (although still no greater than 1); second and more important, the corrections introduced at each step are of high rank, so that there are no favorable dependences among signs to retard growth. The rough agreement of the data with the curve $e^{n/(4 \log n)}$ suggests that perhaps this particular pivoting strategy, unlike partial or complete pivoting, approximately satisfies hypotheses (1) and (2) of the Introduction.

9. Conclusions. Is Gaussian elimination with partial pivoting stable on average? Everything we know on the subject indicates that the answer is emphatically yes, and that one needs no hypotheses beyond statistical properties to account for the success of this algorithm during nearly half a century of digital computation.

This paper has presented a model of the average-case behavior of Gaussian elimination supported by extensive experiments. Although no theorems have been proved, we believe that there is reasonably good evidence for the following conclusions. These statements are approximate, not exact, and they apply to the average case for many, but not all, distributions of matrices. Except where otherwise indicated, they apply to Gaussian elimination with either partial or complete pivoting.

(1) For QR factorization with or without column pivoting, the average maximum element of the residual matrix is $O(n^{1/2})$, whereas for Gaussian elimination it is $O(n)$. This comparison reveals that Gaussian elimination is mildly unstable, but the instability would only be detectable for very large matrix problems solved in low precision. For most practical purposes Gaussian elimination is highly stable on average. (§§ 6, 7)

(2) The statistical behavior of Gaussian elimination depends on the standard deviation of the initial matrix elements, but is otherwise insensitive to the matrix distribution. In particular, the statements below apply equally to random matrices with elements from normal, uniform, or discrete distributions, as well as to random symmetric and Toeplitz matrices (but not to random orthogonal matrices). (§§ 2–6)

(3) For $n \leqq 1024$, the average growth factor (normalized by the standard deviation of the initial elements) is within a few percent of $n^{2/3}$ for partial pivoting and is approximately $n^{1/2}$ for complete pivoting. (§ 6)

(4) After the first few steps of Gaussian elimination, the remaining matrix elements are approximately normally distributed, regardless of whether they started out that way. (§ 2)

(5) The average magnitudes of pivots relative to elements at each step of elimination can be predicted by extreme value statistics. The distribution of multipliers at each step can then be predicted based on the pivot magnitudes. (§§ 3, 4)

(6) The signs of the elements and multipliers are not independent, and their dependence is essential to the stability of Gaussian elimination. It results from the fact that each step of elimination introduces a rank-1 correction to the remaining matrix, which provides a feedback mechanism that inhibits potential element growth and instability. (§§ 5, 8)

(7) This dependence of elements and multipliers can be modeled by hypothesizing that the corrections added at each step of elimination accumulate additively rather than multiplicatively. The resulting predictions of growth factors agree reasonably well with observations. (§§ 5, 6)

(8) By contrast, nonclassical variants of Gaussian elimination involving higher-rank elimination steps are sometimes markedly unstable, even though the multipliers are small. (§ 8)

Acknowledgments. We are grateful for the advice of a number of colleagues, including Persi Diaconis, Alan Edelman, Nick Gould, Nick Higham, Vel Kahan, Tom

Spencer, Gil Strang, and Dan Zwillinger. Special thanks go to Kapil Mathur and Anne Trefethen of Thinking Machines Corporation for catching an error in our computation of residuals in an earlier draft; these two are in the process of obtaining results for larger n with the aid of a Connection Machine. Trefethen would also like to acknowledge the generous support of Iain Duff and John Reid during a visit to the Harwell Atomic Energy Research Establishment (January 1987) and of Bill Morton during a visit to the Oxford Computing Laboratory (Summer 1987).

Note added in proof. This paper has focused mainly on averages, not distributions. Ultimately, however, it is the tail of the growth factor distribution that is of greatest concern. Experiments leave little doubt that the tail decays exponentially, and to illustrate, the following figure is a histogram of computed growth factors ρ in an experiment involving partial pivoting applied to $N = 20{,}000$ matrices of dimension $n = 32$ with normally distributed elements. Note the logarithmic scale. We leave it to others to determine how close such figures are to standard distributions such as the extreme value distribution. A. J. MacLeod has previously carried out experiments in this line [21], [34], and further statistical analysis of pivoting data is being carried out by D. Hoaglin in the Dept. of Statistics, Harvard University.

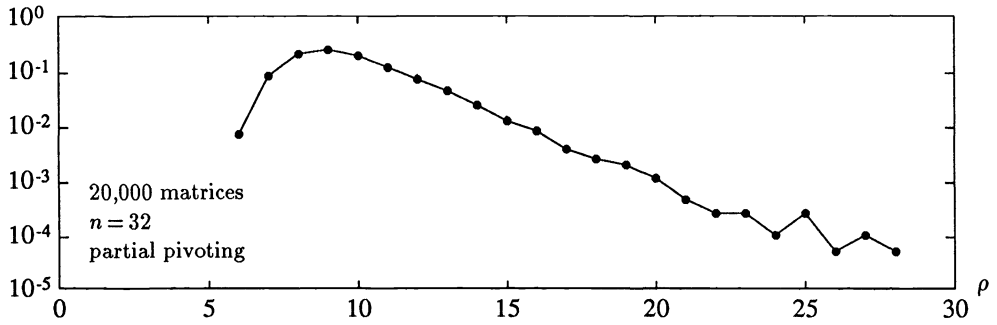


FIG. *Partial pivoting growth factor distribution based on* 20,000 *matrices of dimension* $n = 32$.

REFERENCES

[1] V. BARGMANN, D. MONTGOMERY, AND J. VON NEUMANN, *Solution of linear systems of high order*, Princeton, 1946. Reprinted in von Neumann's *Collected Works*, Vol. 5, A. H. Taub, ed., Pergamon, Elmsford, NY 1963.

[2] G. BIRKHOFF AND S. GULATI, *Isotropic distributions of test matrices*, J. Appl. Math. Phys. (ZAMP) 30 (1979), pp. 148–158.

[3] K. H. BORGWARDT, *The Simplex Method: A Probabilistic Analysis*, Springer-Verlag, Berlin, New York, 1987.

[4] C. W. CRYER, *Pivot size in Gaussian elimination*, Numer. Math., 12 (1968), pp. 335–345.

[5] A. R. CURTIS AND J. K. REID, *The solution of large sparse unsymmetric systems of linear equations*, J. Inst. Maths. Appl., 8 (1971), pp. 344–353.

[6] G. DANTZIG, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[7] J. DAY AND B. PETERSON, *Growth in Gaussian elimination*, Amer. Math. Monthly, 95 (1988), pp. 489–513.

[8] J. J. DONGARRA, J. R. BUNCH, C. B. MOLER, AND G. W. STEWART, LINPACK *Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[9] I. S. DUFF, *Practical comparisons of codes for the solution of sparse linear systems*, in Sparse Matrix Proceedings 1978, I. S. Duff and G. W. Stewart, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[10] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.

[11] A. EDELMAN, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 543–560.

[12] ———, *Eigenvalues and condition numbers of random matrices*, Ph.D. thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[13] R. A. FISHER AND L. H. C. TIPPETT, *Limiting forms of the frequency distribution of the largest or smallest member of a sample*, Proc. Cambridge Philos. Soc., 24 (1928), pp. 180–190.

[14] G. E. FORSYTHE AND C. B. MOLER, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[15a] V. L. GIRKO, *The central limit theorem for random determinants*, Theory Probab. Appl., 24 (1979), pp. 729–740.

[15b] ———, Theory Probab. Appl., 26 (1981), pp. 521–531.

[16] J. T. GOODMAN AND C. B. MOLER, *Three numerical experiments with Gaussian elimination*, Linpack Working Note #8, Applied Mathematics Division, Argonne National Laboratory, Argonne, IL, 1977.

[17] E. J. GUMBEL, *Statistics of Extremes*, Columbia University Press, New York, 1958.

[18] 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.

[19] H. HOTELLING, *Some new methods in matrix calculation*, Ann. Math. Statist., 14 (1943), pp. 1–34.

[20] V. KLEE AND G. MINTY, *How good is the simplex algorithm?*, in Inequalities III, O. Shisha, ed., Academic Press, New York, 1972.

[21] A. J. MACLEOD, *The distribution of the growth factor in Gaussian elimination with partial pivoting*, unpublished Tech. Report, Department of Mathematics and Statistics, Paisley College, Renfrewshire, Scotland.

[22] A. V. OPPENHEIM AND R. W. SCHAFER, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.

[23] W. H. PRESS, B. P. FLANNERY, S. A. TEUKOLSKY, AND W. T. VETTERLING, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, London, 1986.

[24] R. SHAMIR, *The efficiency of the simplex method: a survey*, Management Sci., 33 (1987), pp. 301–334.

[25] S. SMALE, *On the average number of steps in the simplex method of linear programming*, Math. Programming, 27 (1983), pp. 241–262.

[26] ———, *On the efficiency of algorithms of analysis*, Bull. Amer. Math. Soc., 13 (1985), pp. 87–121.

[27] D. C. SORENSEN, *Analysis of pairwise pivoting in Gaussian elimination*, IEEE Trans. Comput., 34 (1984), pp. 274–78.

[28] 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.

[29] J. T. TOMLIN, *Pivoting for size and sparsity in linear programming inversion routines*, J. Inst. Maths. Appl., 10 (1972), pp. 289–295.

[30] A. M. TURING, *Rounding-off errors in matrix processes*, Quart. J. Mech. Appl. Math., 1 (1948), pp. 287–308.

[31a] J. VON NEUMANN AND H. H. GOLDSTINE, *Numerical inverting of matrices of high order*, Bull. Amer. Math. Soc., 53 (1947), pp.1021–1099; von Neumann's *Collected Works*, Vol. 5, A. H. Taub, ed., Pergamon, Elmsford, NY, 1963.

[31b] ———, *Numerical inverting of matrices of high order, Part II*, Proc. Amer. Math. Soc., 2 (1951), pp. 188–202; von Neumann's Collected Works, Vol. 5, A. H. Taub, ed., Pergamon Press, Elmsford, NY, 1963.

[32] J. H. WILKINSON, *Error analysis of direct methods of matrix inversion*, J. Assoc. Comput. Mach., 8 (1961), pp. 281–330.

[33] ———, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[34] A. J. MACLEOD, *Some statistics on Gaussian elimination with partial pivoting*, ACM SIGNUM Newsletter, 24, nos. 2–3 (1989), pp. 10–14.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 361–368, July 1990

002

MONOTONE CORRELATION AND MONOTONE DISJUNCT PIECES*

DEVENDRA CHHETRY†, JAN DE LEEUW‡, AND ALLAN R. SAMPSON§

Abstract. Suppose X, Y are random variables taking values on the $m \times n$ lattice $\{x_1 < \cdots < x_m\} \times \{y_1 < \cdots < y_n\}$ with $Q = \{\text{Prob}\,(X = x_i, Y = y_j)\}$. Let $\rho_{\text{CMC}}(Q)$ and $\rho_{\text{DMC}}(Q)$ be the concordant and discordant monotone correlations defined, respectively, by the maximum and minimum of correlation $f(X)$, $g(Y)$ over all f, g increasing with nonzero variances. A number of results concerning $\rho_{\text{CMC}}(Q)$ and $\rho_{\text{DMC}}(Q)$ and their evaluations are obtained. One result shows that $\rho_{\text{CMC}}(Q) = 1$, if and only if Q consists of at least two increasing disjunct pieces, i.e., $Q = \text{Diag}\,(Q_1, Q_2)$. Necessary and sufficient conditions are also given for $\rho_{\text{CMC}}(Q) = \rho_{\text{DMC}}(Q)$.

Key words. maximal correlation, concordant monotone correlation, disjunct pieces, monotone disjunct pieces

AMS(MOS) subject classifications. primary 15A51; secondary 62H20

1. Introduction.

Let X and Y be two discrete random variables taking values in the $m \times n$ lattice $S \times T \equiv \{x_1 < \cdots < x_m\} \times \{y_1 < \cdots < y_n\}$ with

$$Q \equiv \{q_{ij}\} = \{\text{Prob}\,(X = x_i, Y = y_j)\},$$

where we assume $r_i \equiv \sum_j q_{ij} > 0$ for all i and $c_j \equiv \sum_i q_{ij} > 0$ for all j. There is a substantial literature in statistics and probability dealing with measuring the association between the random variables X and Y (see Goodman and Kruskal (1979), Haberman (1982) or Raveh (1986)). One such measure of association introduced by Hirschfeld (1935) is the maximal correlation coefficient $\rho'(X, Y)$ (or $\rho'(Q)$) defined to be the max $\{\rho(f(X), g(Y))\}$, where ρ denotes correlation and the maximum is over all f and g with nonzero variances. Clearly, $0 \leqq \rho'(X, Y) \leqq 1$.

The properties of $\rho'(X, Y)$ have been extensively studied (e.g., Richter (1949), Rényi (1959), Lancaster (1969), Sarmanov (1958a), (1958b), and Hall (1969)). One of the interesting and important results is that $\rho'(X, Y) = 0$ is equivalent to X and Y being independent random variables, and $\rho'(X, Y) = 1$ is equivalent to Q consisting of at least two disjunct pieces, where this concept is defined as follows.

DEFINITION 1.1 (Richter (1949)). The probability matrix Q is said to consist of k *disjunct pieces* if there exist partitions $S_1, \cdots, S_k$ of S and $T_1, \cdots, T_k$ of T such that

$$\text{Prob}\,((X, Y) \in S_i \times T_i) > 0, \qquad i = 1, \cdots, k, \tag{1.1}$$

and

$$\text{Prob}\,((X, Y) \in S_i \times T_j) = 0 \quad \text{for all } i \neq j. \tag{1.2}$$

* Received by the editors March 30, 1988; accepted for publication (in revised form) August 16, 1989.

† Department of Mathematics, Tribhuvan University, Kirtipur Campus, Kathmandu, Nepal. The research of this author was done in part at the Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, Pennsylvania 15260, and was supported by the Air Force Office of Scientific Research under contract AFOSR-84-0113.

‡ Psychology Department, University of California at Los Angeles, 405 Hilgarde Avenue, Los Angeles, California 90024-1563.

§ Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, Pennsylvania 15260. The research of this author was done in part at the Department of Statistics, Carnegie-Mellon University, Pittsburgh, Pennsylvania, and supported by the Air Force Office of Scientific Research under contract AFOSR-84-0113.

Additionally, the probability matrix Q is said to consist of *exactly* k disjunct pieces, if (1.1) and (1.2) hold, and Q cannot further consist of $k+1$ disjunct pieces. Richter (1949) has extended this result concerning disjunct pieces utilizing Fisher's canonical decomposition of Q. Define $Q^* \equiv D_r^{-1/2} Q D_c^{-1/2}$, where $D_r = \text{Diag}(r_1, \cdots, r_m)$ and $D_c = \text{Diag}(c_1, \cdots, c_n)$. Then, assuming here for convenience $m \leqq n$, the spectral decomposition of Q^* can be written as $Q^* = \Gamma[\text{Diag}(1, \rho_1, \cdots, \rho_{m-1}) : O_{m,n-m}]G'$, where $\Gamma = [D_r^{1/2}\mathbf{l}_m : \Gamma_1]$ and $G = [D_c^{1/2}\mathbf{l}_n : G_1]$ are orthonormal matrices, $O_{m,n-m}$ is an $m \times (n-m)$ matrix of zeros, and $1 \geqq \rho_1^2 \geqq \cdots \geqq \rho_{m-1}^2 \geqq 0$ are the eigenvalues of $Q^{*\prime}Q^*$. Based on this spectral decomposition, Fisher's (1940) canonical decomposition can be written

$$Q = \mathbf{r}\mathbf{c}' + D_r^{1/2}\Gamma_1 D_\rho (D_c^{1/2} G_1)',$$

where $D_\rho = [\text{Diag}(\rho_1, \cdots, \rho_{m-1}) : O_{m-1,n-m}]$. The values $\rho_1, \cdots, \rho_{m-1}$ are called the canonical correlations of the distribution Q, where it is known that $\rho'(X, Y) = \rho_1$. (See Lancaster (1969, Chap. 6) or Chhetry and Sampson (1987) for further discussions concerning the canonical decomposition and its interpretation.) The result obtained by Richter (1949) is that Q consists of exactly k disjunct pieces if and only if $\rho_1 = \cdots = \rho_{k-1} = 1$ and $\rho_k < 1$.

Another related concept is the following one. If $m = n$ and Q consists of m disjunct pieces, then X and Y are called mutually completely dependent (Lancaster (1969)), and there exists a one-to-one function h such that the random variables X and Y are completely related by $Y = h(X)$.

For the purposes of this paper we require a further refinement of the concept of disjunct pieces. To define this refinement, we employ the notation that if U, V are sets of real numbers, $U < V$ means $u < v$ for all $u \in U$ and all $v \in V$.

DEFINITION 1.2. The probability matrix Q is said to consist of k *increasing* (*decreasing*) *disjunct pieces* if there exists partitions $S_1 < S_2 < \cdots < S_k$ of S and $T_1 < (>)\ T_2 < (>) \cdots < (>)\ T_k$ of T such that (1.1) and (1.2) hold.

We say Q consists of k *monotone disjunct pieces* if Q consists of either k increasing or decreasing disjunct pieces.

Q consisting of k increasing disjunct pieces is equivalent to

$$Q = \text{Diag}(Q_1, \cdots, Q_k),$$

where Q_i is an $m_i \times n_i$ matrix and $\sum m_i = m$, $\sum n_i = n$. This also can be viewed as Q being the direct sum $Q_1 \oplus \cdots \oplus Q_k$, when direct sum in this context is analogous to the direct sum of square matrices (see MacDuffee (1949, p. 114)). If $m = n$ and Q consists of m increasing (decreasing) disjunct pieces the notion of X and Y being mutually completely dependent can be refined. In this case X and Y are related by h strictly increasing (decreasing) and the probability matrix corresponds to a special class of probability distributions called the upper (lower) Fréchet bounds (see Kimeldorf and Sampson (1978)).

In order to measure positive association between arbitrary random variables X and Y and also to circumvent some of the difficulties pointed out by Kimeldorf and Sampson (1978), Kimeldorf, May, and Sampson (KMS) (1982) introduced the concordant monotone correlation ρ_{CMC} (or alternatively $\rho_{\text{CMC}}(Q)$), defined by

$$\rho_{\text{CMC}} = \max\{\rho(f(X), g(Y))\} \tag{1.3}$$

where the maximum is taken over all increasing f and g with nonzero variances. Also introduced by KMS is the discordant monotone correlation $\rho_{\text{DMC}}(Q)$ defined by (1.3) where "max" is replaced by "min." KMS show that $-1 \leqq \rho_{\text{DMC}} \leqq \rho_{\text{CMC}} \leqq 1$, and

$\rho_{\mathrm{DMC}} = \rho_{\mathrm{CMC}} = 0$ is equivalent to X and Y being independent random variables. Also they provide an example where $\rho_{\mathrm{DMC}} < \rho_{\mathrm{CMC}} = 0$ and yet X and Y are dependent random variables. It is also direct to show that $\rho_{\mathrm{DMC}} \geqq 0$ ($\rho_{\mathrm{CMC}} \leqq 0$) if and only if X and Y are positively (negatively) quadrant dependent (Lehmann (1966)), i.e., Prob $(X \leqq x, Y \leqq y) \geqq (\leqq)$ Prob $(X \leqq x)$ Prob $(Y \leqq y)$ for all x, y.

The purpose of this paper is to obtain some additional results in the bivariate discrete setting concerning ρ_{CMC} and ρ_{DMC}, and their evaluation.

2. Some results for ρ_{CMC}. For a given probability matrix Q the notation used for the correlation between $f(X)$ and $g(Y)$ is

$$\rho_Q(\mathbf{f},\mathbf{g}) = (\mathbf{f}'(D_r - \mathbf{r}\mathbf{r}')\mathbf{f})^{-1/2}(\mathbf{g}'(D_c - \mathbf{c}\mathbf{c}')\mathbf{g})^{-1/2}(\mathbf{f}'(Q - \mathbf{r}\mathbf{c}')\mathbf{g}),$$

where

$$\mathbf{r} = (r_1, \cdots, r_m)', \qquad \mathbf{c} = (c_1, \cdots, c_n)',$$

$$\mathbf{f} = (f(x_1), \cdots, f(x_m))', \qquad \mathbf{g} = (g(y_1), \cdots, g(y_n))',$$

and the denominator is nonzero.

Throughout we say the vector $(w_1, \cdots, w_p)'$ is nondecreasing if $w_1 \leqq \cdots \leqq w_p$; and use $\mathbf{e}_k$ to denote the kth coordinate unit vector of the appropriate dimension. Often we use the simple fact that for every $m \times n$ probability matrix Q, there uniquely corresponds an $m \times n$ cumulative distribution matrix defined by

$$F \equiv \{F_{ij}\} = \{\text{Prob}\,(X \leqq x_i, Y \leqq y_j)\},$$

i.e., $F_{ij} = \sum_{k=1}^{i} \sum_{l=1}^{j} q_{kl}$.

THEOREM 2.1. *A necessary and sufficient condition for*

$$\rho_{\mathrm{CMC}}(Q) = 1\,(\rho_{\mathrm{DMC}}(Q) = -1)$$

is that Q consists of at least two increasing (decreasing) disjunct pieces.

Proof. The sufficiency follows immediately (see Kimeldorf, May, and Sampson (1982, p. 120)).

To show necessity, suppose $\rho_{\mathrm{CMC}}(Q) = 1$. Then, there exist two nondecreasing vectors $\mathbf{f}_0$ and $\mathbf{g}_0$, such that $\rho_Q(\mathbf{f}_0, \mathbf{g}_0) = 1$ and thus, Q consists of at least two disjunct pieces. Assume that Q consists of exactly t disjunct pieces, where $t \geqq 2$. Hence, there exist permutation matrices P_1 and P_2 such that $Q^* = P_1 Q P_2'$ consists of exactly t increasing disjunct pieces, i.e., $Q^* = \text{Diag}\,(Q_1^*, \cdots, Q_t^*)$, where Q_k^* is an $m_k \times n_k$ matrix, such that $\sum m_k = m$ and $\sum n_k = n$. It then follows (see Richter (1949) or Bastin et al. (1980)) that $\rho_{Q^*}(\mathbf{f}_0^*, \mathbf{g}_0^*) = 1$ if and only if $\mathbf{f}_0^* = \sum_{s=1}^{t} \lambda_s \mathbf{u}_s$, where $\mathbf{u}_s = \mathbf{e}_{m_1 + \cdots + m_{s-1} + 1} + \cdots + \mathbf{e}_{m_1 + \cdots + m_s}$, and $\mathbf{g}_0^* = \sum_{s=1}^{t} (\alpha\lambda_s + \beta)\mathbf{v}_s$, where $\mathbf{v}_s = \mathbf{e}_{n_1 + \cdots + n_{s-1} + 1} + \cdots + \mathbf{e}_{n_1 + \cdots + n_s}$, and where there exists $i < j$ such that $\lambda_i \neq \lambda_j$ and $\alpha > 0$. It is direct to show that $\rho_Q(\mathbf{f}_0, \mathbf{g}_0) = 1$ if and only if $\mathbf{f}_0 = P_1' \mathbf{f}_0^*$ and $\mathbf{g}_0 = P_2' \mathbf{g}_0^*$ for any $\mathbf{f}_0^*, \mathbf{g}_0^*$, which satisfies $\rho_{Q^*}(\mathbf{f}_0^*, \mathbf{g}_0^*) = 1$. For each vector $\mathbf{f}_0^*, \mathbf{g}_0^*$ of the preceding form, let $i^* \geqq 2$ be the first value such that $\lambda_{i^*} \neq \lambda_1$; the existence of i^* follows from $\lambda_i \neq \lambda_j$ for some $i < j$. Because $\mathbf{f}_0$ is nondecreasing and $\mathbf{f}_0 = P_1' \mathbf{f}_0^*$, it follows that $P_1 = \text{Diag}\,(P_1^{(1)}, P_1^{(2)})$, where $P_1^{(1)}$ is an $m^* \times m^*$ permutation matrix and $P_1^{(2)}$ is an $(m - m^*) \times (m - m^*)$ permutation matrix, where $m^* = \sum_{k=1}^{i^*-1} m_k$. Similarly, P_2 is in block diagonal form and, hence Q consists of at least two increasing disjunct pieces.

Now suppose $\rho_{\mathrm{DMC}}(Q) = -1$. Use the preceding argument and the fact that $\rho_{\mathrm{DMC}}(Q) = -\rho_{\mathrm{CMC}}(Q^*)$ where $Q^* = Q(\mathbf{e}_n, \cdots, \mathbf{e}_1)$ to get the result. □

KMS show that monotone correlation $\rho^*(Q)$, introduced by Kimeldorf and Sampson (1978), is also given by $\rho^*(Q) = \max\{\rho_{\mathrm{CMC}}(Q), -\rho_{\mathrm{DMC}}(Q)\}$. From Theorem 2.1,

it immediately follows that $\rho^*(Q) = 1$ if and only if Q consists of at least two monotone disjunct pieces.

While Theorem 2.1 deals with the case $\rho'(Q) = \rho_{CMC}(Q) = 1$, more generally we have $\rho'(Q) \geqq \rho_{CMC}(Q)$. However, in some cases Schriever (1983) shows that $\rho'(Q) = \rho_{CMC}(Q)$ without their necessarily being unity. We observe that $\rho'(Q) = \rho_{CMC}(Q)$ means that there exists at least one pair of nondecreasing functions f_0 and g_0 such that $\rho(f_0(X), g_0(Y)) = \rho'(Q)$. For a further discussion of Schriever's results we need the following Definition due to Lehmann (1966).

DEFINITION (Lehmann (1966)). A random variable X is said to be *positively regression dependent* (PRD) on Y if $\text{Prob}(X > x | Y = y)$ is nondecreasing in y for all x.

In terms of the probability matrix Q, the condition that X is PRD on Y can be written as follows: For all $i = 2, \cdots, m-1, j < j'$ implies $\sum_{l=i}^{m} q_{lj}/c_j \leqq \sum_{l=i}^{m} q_{lj'}/c_{j'}$.

THEOREM 2.2 (Schriever (1983)). *If X is* PRD *on Y and Y is* PRD *on X, then* $\rho'(Q) = \rho_{CMC}(Q)$.

We note that it is easily shown if Q corresponds to Y being PRD on X (X being PRD on Y), then every $\tilde{Q}$ has the same property, where $\tilde{Q}$ is obtained from Q by adding together (which is equivalent to statistically collapsing data categories) any sets of adjacent rows or adjacent columns. As a consequence of this fact and of Theorem 2.2, it follows that Q corresponding to Y is PRD on X and X is PRD on Y implies that $\rho'(\tilde{Q}) = \rho_{CMC}(\tilde{Q})$ for every collapsed $\tilde{Q}$. However, Chhetry and Sampson (1987) provide an example that the conditions of Theorem 2.2 are not necessary for $\rho'(Q) = \rho_{CMC}(Q)$.

In the study of bivariate dependence concepts, it oftentimes is of interest to consider $P(\mathbf{r}, \mathbf{c})$, the class of all $m \times n$ probability matrices with fixed row and column marginals, $\mathbf{r}$ and $\mathbf{c}$, respectively. It is well known that (see Schriever (1985, Ex. 4.2.3)) $\rho_{CMC}(Q^+) \geqq \rho_{CMC}(Q)$ for all $Q \in P(\mathbf{r}, \mathbf{c})$, where Q^+ is the probability matrix uniquely corresponding to the cumulative distribution matrix of the upper Fréchet bound, which has $F^+ = \{(\min(F_i, G_j))\}$, where $F_i = \sum_{k=1}^{i} r_k$ and $G_j = \sum_{k=1}^{j} c_k$. If the random variables X and Y are both continuous, the CMC for the correspondingly defined upper Fréchet bound is one (see Kimeldorf and Sampson (1978)). However, in the discrete situation it is not always the case that $\rho_{CMC}(Q^+)$ is one. In the following theorem we provide a necessary and sufficient condition for $\rho_{CMC}(Q^+) = 1$ in terms of the marginal row and column sums.

THEOREM 2.3. *A necessary and sufficient condition for* $\rho_{CMC}(Q^+) = 1$ *is that there exist $s < m$ and $t < n$ such that $F_s = G_t$.*

Proof. In view of Theorem 2.1, we need to show that $Q^+ = \text{Diag}(Q_1^+, Q_2^+)$ if and only if $F_s = G_t$, where Q_1^+ is $s \times t$ and Q_2^+ is $(m-s) \times (n-t)$. Obviously, $Q^+ = \text{Diag}(Q_1^+, Q_2^+)$ implies that $F_s = G_t$. To prove the converse assume that $F_s = G_t$. Let F_{ij}^+ be the (i,j)th element of F^+; then it can be easily checked that

$$F_{ij}^+ = \begin{cases} F_i & \text{if } i = 1, 2, \cdots, s \text{ and } j \geqq t, \\ G_j & \text{if } i = s, \text{ and } j < t, \\ G_j & \text{if } i > s, \quad j \leqq t. \end{cases}$$

This implies that the corresponding Q^+ is of the required form. □

To motivate the next theorem, consider first the simple case when Q is a 2×2 probability matrix. Then it is trivial to show that $\rho_{CMC}(Q) = \rho_{DMC}(Q)$; additionally, $\rho_{CMC}(Q) = -1$ ($\rho_{DMC}(Q) = 1$) if and only if $q_{11} = q_{22} = 0$ ($q_{12} = q_{21} = 0$). The analogous results do not continue to hold when $m > 2$ or $n > 2$, as we now show.

THEOREM 2.4. *If $m > 2$ or $n > 2$, then $\rho_{CMC}(Q) = \rho_{DMC}(Q)$ if and only if X and Y are independent.*

Proof. Suppose $\rho_{CMC}(Q) = \rho_{DMC}(Q) = \eta \neq 0$ (if $\eta = 0$, independence follows). Without loss of generality assume $m > 2$, so that we can choose three nondecreasing functions a_1, a_2, and b such that (i) $\rho(a_1(X), a_2(X)) < 1$ and (ii) $\mathrm{Var}[a_1(X)] = \mathrm{Var}[a_2(X)] = \mathrm{Var}[b(Y)] = 1$. Then, by the assumption that $\rho_{CMC}(Q) = \rho_{DMC}(Q)$,

$$\eta = \rho(a_1(X)+a_2(X), b(Y)) = 2\eta(2+2\rho(a_1(X), a_2(X)))^{-1/2},$$

which implies that $\rho(a_1(X), a_2(X)) = 1$, a contradiction. □

COROLLARY 2.5. *If $m > 2$ or $n > 2$, then $\rho_{CMC}(Q) > -1$ and $\rho_{DMC}(Q) < 1$.*

The proof of Corollary 2.5 is obvious.

3. Some results concerning evaluation. While the quantities $\rho'(Q)$ and $\rho_{CMC}(Q)$ are of interest in their own right as measures of association, the vectors at which these maxima occur play an important role in rescaling of the values of the random variables. These notions are particularly useful in statistically analyzing both nominal and ordinal contingency tables (e.g., Nishisato (1980)). The vectors that maximize $\rho'(Q)$ can be derived from certain results of statistical correspondence analysis (e.g., Benzecri (1973) and Hill (1974)). The increasing vectors that yield $\rho_{CMC}(Q)$ can be interpreted as either providing dual scalings for ordinal contingency tables or a form of ordinal correspondence analysis. However, their evaluation is substantially more complicated than the nonordinal case (e.g., see KMS, or Breiman and Friedman (1985), and the comments of Buja and Kass (1985)). Chhetry and Sampson (CS) (1987) provide an approach that simplifies somewhat the calculation of $\rho_{CMC}(Q)$ and the maximizing vectors. We briefly discuss that approach and then detail how to employ it effectively when the ordinal table is collapsed, i.e., when neighboring row or columns are added. The latter issue is important for the statistical modeling using hierarchies for ordinal tables in which collapsing is used for model simplification.

For every $m \times n$ probability matrix Q, CS define the $(m + n - 2) \times (m + n - 2)$ matrix $\Sigma(Q)$ (denoted where there is no ambiguity as Σ) by

$$\Sigma(Q) = \begin{pmatrix} \bar{A}' & 0 \\ 0 & \bar{B}' \end{pmatrix} \begin{pmatrix} D_r & Q \\ Q' & D_c \end{pmatrix} \begin{pmatrix} \bar{A} & 0 \\ 0 & \bar{B} \end{pmatrix}, \tag{3.1}$$

where $\bar{A} = (I_m - \mathbf{1}_m \mathbf{1}'_m D_r)\Psi_m$, $\bar{B} = (I_n - \mathbf{1}_n \mathbf{1}'_n D_c)\Psi_n$, and Ψ_p is the $p \times (p-1)$ matrix whose (i, j)th element is zero, if $i \leqq j$, and 1, otherwise. Let $\Sigma_{11} = \bar{A}' D_r \bar{A}$, $\Sigma_{12} = \bar{A}' Q \bar{B}$, $\Sigma_{22} = \bar{B}' D_c \bar{B}$, and $\Sigma_{21} = \Sigma'_{12}$. CS also show that Σ_{11} and Σ_{22} are positive definite and Σ is a nonnegative-definite matrix. For any Q, let Σ be given by (3.1) and define for $\alpha \in R^{m-1}$, $\beta \in R^{n-1}$

$$r_Q(\alpha, \beta) = (\alpha' \Sigma_{11} \alpha)^{-1/2} (\alpha' \Sigma_{12} \beta)(\beta' \Sigma_{22} \beta)^{-1/2} \tag{3.2}$$

where $\alpha \neq 0$ and $\beta \neq 0$. Then CS show that the maximal correlation coefficient and the two monotone correlation coefficients can be evaluated as follows:

$$\rho'(Q) = \max_{\alpha, \beta} r_Q(\alpha, \beta), \tag{3.3a}$$

$$\rho_{CMC}(Q) = \max_{\alpha \geqq 0, \beta \geqq 0} r_Q(\alpha, \beta), \tag{3.3b}$$

$$\rho_{DMC}(Q) = \min_{\alpha \geqq 0, \beta \geqq 0} r_Q(\alpha, \beta). \tag{3.3c}$$

The relationships of (3.3a)–(3.3c) can be viewed as simplifying computation by reducing dimensionality. Also note that if α_0 and β_0 optimize any of (3.3a), (3.3b), or (3.3c), then the corresponding maximizing vectors $\mathbf{f}_0$ and $\mathbf{g}_0$ defining the left-hand sides are related by $\mathbf{f}_0 = \bar{A}\alpha_0$ and $\mathbf{g}_0 = \bar{B}\beta_0$. For example, if $r_Q(\alpha, \beta)$ is maximized at α_0, β_0, then $\rho_Q(\mathbf{f}, \mathbf{g})$ is maximized at $\mathbf{f}_0 = \bar{A}\alpha_0$ and $\mathbf{g}_0 = \bar{B}\beta_0$.

An additional advantage of the problem formulation given by (3.2) and (3.3) is that these optimization problems can be reformulated analogously to the problem of finding the canonical correlation for the multivariate normal. A good discussion concerning traditional multivariate normal canonical correlations is given in Anderson (1984, Chap. 12). For the p-dimensional multivariate normal distribution with positive-definite covariance matrix Σ, canonical correlation analysis involves a study of the determinental roots and solutions for $\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} - \lambda^2\Sigma_{22}$, where Σ_{11}, Σ_{12}, Σ_{21}, Σ_{22} are a partitioning of Σ with the dimension of Σ_{11} being $p_1 < p$. A description of the relationship between our problem and traditional canonical correlation analysis is given in the following lemma whose proof follows from Lemma 4.1 and Theorem 4.2 of CS.

LEMMA 3.1. *The positive square root of the largest eigenvalue* ρ_1^2 *of* $\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ (*or* $\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$) *is* $\rho'(Q)$. *If* $\alpha^{(1)} \neq 0$ *and* $\beta^{(1)} \neq 0$ *satisfy the equations*

$$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\alpha^{(1)} = \rho_1^2\alpha^{(1)} \tag{3.4a}$$

and

$$\beta^{(1)} = \Sigma_{22}^{-1}\Sigma_{21}\alpha^{(1)}, \tag{3.4b}$$

then $\rho_Q(\alpha^{(1)}, \beta^{(1)}) = \rho'(Q)$. *Moreover,* $\rho'(Q) = \rho_{\mathrm{CMC}}(Q)$ *if and only if there exist nonnegative vectors* $\alpha^{(1)}$ *and* $\beta^{(1)}$ *satisfying* (3.4).

We now relate the computation of the maximal correlation and the monotone correlations for collapsed contingency tables to the original uncollapsed tables. Recent discussions on the general issue of collapsing nonordinal contingency tables are given by Gilula and Krieger (1983) and Gilula (1986). The following definition is useful in our discussion.

DEFINITION 3.2. An $m \times n$ matrix $P = \{p_{ij}\}$, $m \leqq n$, is said to be a C-matrix if (a) the rank of P is m; (b) each column of P has one and only one nonzero element, and the nonzero element is unity; and (c) if $p_{ij} = p_{ik} = 1$ for $k > j$ implies $p_{ie} = 1$ for all $e = j + 1, \cdots, k - 1$.

Obviously, in the above definition, if $m = n$ then P is a permutation matrix; and if $m < n$ then appropriate multiplication of a probability matrix by P collapses sets of adjacent rows or columns. Suppose Q is transformed to $\tilde{Q}$ by $\tilde{Q} = P_1QP_2'$, where P_1 and P_2 are, respectively, $s \times m$ and $t \times n$ C-matrices. Then, $\tilde{Q}$ is an $s \times t$ probability matrix obtained from Q by collapsing and with row and column marginals $\tilde{\mathbf{r}} = P_1\mathbf{r} = (\tilde{r}_1, \cdots, \tilde{r}_s)'$ and $\tilde{\mathbf{c}} = P_2\mathbf{c} = (\tilde{c}_1, \cdots, \tilde{c}_t)'$, respectively. Moreover, if $D_{\tilde{r}} = \mathrm{Diag}(\tilde{r}_1, \cdots, \tilde{r}_s)$ and $D_{\tilde{c}} = \mathrm{Diag}(\tilde{c}_1, \cdots, \tilde{c}_t)$, then $D_{\tilde{r}} = P_1D_rP_1'$ and $D_{\tilde{c}} = P_2D_cP_2'$.

In the following theorem, we establish the relationship between $\Sigma(Q)$ and $\Sigma(\tilde{Q})$.

THEOREM 3.3. *If* $\tilde{Q} = P_1QP_2'$, *where* P_1 *and* P_2 *are, respectively,* $s \times m$ *and* $t \times n$ C-*matrices, then*

$$\Sigma(\tilde{Q}) = \mathrm{Diag}(K_m', K_n')\Sigma(Q)\,\mathrm{Diag}(K_m, K_n),$$

where $K_m = \Delta_m'P_1'\Psi_s$, $K_n = \Delta_n'P_2'\Psi_t$, *and* Δ_p *is the* $p \times (p-1)$ *matrix*

$$(\mathbf{e}_2 - \mathbf{e}_1, \mathbf{e}_3 - \mathbf{e}_2, \cdots, \mathbf{e}_p - \mathbf{e}_{p-1}).$$

Proof. From CS (Lemma 3.2(i))

$$\Sigma_{12}(\tilde{Q}) = \Psi_s'(\tilde{Q} - \tilde{\mathbf{r}}\tilde{\mathbf{c}}')\Psi_t$$

$$= \Psi_s' P_1 (Q - \mathbf{r}\mathbf{c}') P_2' \Psi_t.$$

From the quadrant dependence decomposition (CS (equation (3.4))), we obtain

$$\Sigma_{12}(\tilde{Q}) = \Psi_s' P_1 \Delta_m \Sigma_{12}(Q) \Delta_n' P_2' \Psi_t$$

$$= K_m' \Sigma_{12}(Q) K_n.$$

The relationship concerning $\Sigma_{11}(\tilde{Q})$ and $\Sigma_{22}(\tilde{Q})$ are established similarly. □

Note that the results of Theorem 3.3 also hold if P_1 and P_2 are more general in that they collapse nonadjacent rows and columns; however, such matrices would not be meaningful for ordinal tables. The usefulness of Theorem 3.3 especially when used in conjunction with Lemma 3.1 can be seen in the following example.

Example 3.4. Let P_1 and P_2 be C-matrices of orders $(m-s)\times m$ and $(n-t)\times n$, respectively, where

$$P_1 \equiv (\mathbf{e}_1, \cdots, \mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{m-s}) \quad \text{and} \quad P_2 \equiv (\mathbf{e}_1, \cdots, \mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{n-t}).$$

Then, the matrices, K_m and K_n defined in Theorem 3.3 reduce to the form

$$K_m' = (0_1, I_{(m-s-1)}) \quad \text{and} \quad K_n' = (0_2, I_{(n-s-1)}) \tag{3.5}$$

where 0_1 and 0_2 are zero matrices of orders $(m-s-1)\times s$ and $(n-t-1)\times t$, respectively. Hence, using (3.5) in Theorem 3.3, we obtain

$$\Sigma_{12}(\tilde{Q}) = \Sigma_{12}[1, 2, \cdots, s; 1, 2, \cdots, t],$$

$$\Sigma_{11}(\tilde{Q}) = \Sigma_{11}[1, 2, \cdots, s; 1, 2, \cdots, s],$$

and

$$\Sigma_{22}(\tilde{Q}) = \Sigma_{22}[1, 2, \cdots, t; 1, 2, \cdots, t],$$

where $\Sigma_{11}[1, 2, \cdots, i; 1, 2, \cdots, k]$ is the submatrix obtained from $\Sigma_{11}(Q)$, by deleting the first i rows and the first k columns, etc.

REFERENCES

T. W. ANDERSON (1984), *An Introduction to Multivariate Statistical Analysis*, 2nd ed., John Wiley, New York.

CH. BASTIN, J. P. BENZECRI, CH. BOURGARIT, AND P. CAZES (1980), *Pratique de l'analyse de donnees*, Dunod, Paris.

J. P. BENZECRI (1973), *L'analyse des donnees* II, Dunod, Paris.

L. BREIMAN AND J. H. FRIEDMAN (1985), *Estimating optimal transformations for multiple regression and correlation*, J. Amer. Statist. Assoc., 80, pp. 580–598.

A. BUJA AND R. E. KASS (1985), *Some observations on* ACE *methodology*, J. Amer. Statis. Assoc., 80, pp. 602–607.

D. CHHETRY AND A. SAMPSON (1987), *A projection decomposition for bivariate discrete probability distributions*, SIAM J. Algebraic Discrete Methods, 8, pp. 501–509.

R. A. FISHER (1940), *The precision of discriminant functions*, Ann. Eugen. London, 10, pp. 422–429.

Z. GILULA (1986), *Grouping and association in contingency tables: An exploratory canonical correlation approach*, J. Amer. Statist. Assoc., 81, pp. 773–779.

Z. GILULA AND A. M. KRIEGER (1983), *The decomposability and monotonicity of Pearson's chi-square for collapsed contingency tables*, J. Amer. Statist. Assoc., 78, pp. 176–80.

L. GOODMAN AND W. KRUSKAL (1979), *Measure of Association for Cross Classifications*, Springer-Verlag, New York.

S. HABERMAN (1982), Association, measures of, in Encyclopedia of Statistical Sciences (Vol. 1), S. Kotz and N. Johnson, eds., John Wiley, New York, pp. 130–137.

W. J. HALL (1969), *On characterizing dependence in joint distributions*, in Essays in Probability and Statistics, R. Bose, I. Chakravarti, P. Mahalanobis, C. Rao, and K. Smith, eds., University of North Carolina Press, Chapel Hill, NC.

M. O. HILL (1974), *Correspondence analysis: A neglected multivariate method*, Appl. Statist., 23, pp. 340–354.

H. O. HIRSCHFELD (1935), *A connection between correlation and contingency*, Proc. Cambridge Philos. Soc., 31, pp. 520–524.

G. KIMELDORF AND A. R. SAMPSON (1978), *Monotone dependence*, Ann. Statist., 6, pp. 895–903.

G. KIMELDORF, J. MAY, AND A. R. SAMPSON (1982), *Concordant and discordant monotone correlations and their evaluation by nonlinear optimization*, in Optimization in Statistics, S. Zanakis and J. Rustagi, eds., TIMS Studies Management Sci., 19, pp. 117–130.

H. O. LANCASTER (1969), *The Chi-Squared Distribution*, John Wiley, New York.

——— (1958), *The structure of bivariate distributions*, Ann. Math. Statist., 29, pp. 719–736.

E. L. LEHMANN (1966), *Some concepts of dependence*, Ann. Math. Statist., 37, pp. 1137–1153.

C. MACDUFFEE (1949), *Vectors and Matrices*, The Mathematical Association of America, Washington, DC.

S. NISHISATO (1980), *Analysis of Categorical Data: Dual Scaling and its Applications*, University of Toronto Press, Toronto, Ontario, Canada.

A. RAVEH (1986), *On measures of monotone association*, Amer. Statist., 40, pp. 117–123.

A. RÉNYI (1959), *On measures of dependence*, Acta. Math. Acad. Sci. Hungar., 10, pp. 441–451.

H. RICHTER (1949), *Zur maximal correlation*, Z. Angew. Math. Mech., 19, pp. 127–128.

O. V. SARMANOV (1958a), *The maximal correlation coefficient (symmetric case)*, Dokl. Akad. Nauk. SSSR, 120, pp. 715–718. (In Russian.) Sep. Transl. Math. Statist. Probab., 4, pp. 271–275. (In English.)

——— (1958b), *The maximal correlation coefficient (non-symmetric case)*, Dokl. Akad. Nauk. SSSR, 121, pp. 52–55. (In Russian.) Sep. Transl. Math. Statist. Probab., 4, pp. 207–210. (In English.)

B. F. SCHRIEVER (1985), *Order dependence*, Ph.D. thesis, Free University of Amsterdam, Amsterdam, the Netherlands.

——— (1983), *Scaling of order dependent categorical variables with correspondence analysis*, Internat. Statist. Rev., 51, pp. 225–238.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 369–372, July 1990

003

AN EIGENVALUE FORMULA FOR THE RADIUS OF STABILITY OF A STABLE GAME MATRIX*

MARVIN D. TROUTT†

Abstract. This note develops an alternative to computing the radius of stability of a stable game matrix in terms of the eigenvalues of matrices derived from the game matrix.

Key words. stable game matrices, eigenvalue, perturbation methods

AMS(MOS) subject classifications. 65F15, 90D99

Connections between games and eigensystems have been studied by several authors, including [1]–[6]. In a previous paper [7], the author introduced the concept of stable optimal mixed strategies in two-person zero sum games and gave a computational formula for the radius of stability. An alternative formula for the radius of stability is presented here. This result demonstrates a relationship to eigenvalue problems for matrices derived from the game payoff matrix by deleting rows and columns.

Let A be an $n \times n$ matrix, and let $M(i)$ be the $n \times n$ matrix whose ith row consists of 1's and whose other entries are 0. Let $\lambda^* = (\lambda_1^*, \cdots, \lambda_n^*)$ be an optimal strategy of player I in the game with payoff matrix $A_{n\times n}$, where λ_i^* is the optimal probability of playing row i. Let v be the value of the game. A strategy λ is called completely mixed if $\lambda_i > 0$, $i = 1$ to n. A game is said to be completely mixed if and only if every optimal strategy of either player is completely mixed. We have the following.

DEFINITION 1. We will call λ^* stable if there is a $\delta > 0$ such that λ^* is an optimal strategy of player I in all of the games with payoff matrices[1] $A + \sum_{i=1}^{n} x_i M(i)$ for all $\|x\| < \delta$. Similar concepts for player II and columns are evident.

The following result was obtained in [7] and is stated without proof.

THEOREM 1. *Let λ^* be an optimal strategy of player* I *in the game with payoff matrix $A_{n\times n}$. Suppose λ^* is completely mixed. Then the game is completely mixed if and only if λ^* is stable.*

Remark. A similar characterization of completely mixed games was obtained earlier and independently by Filar [8, Cor. 3.2].

DEFINITION 2. Let $\rho(x)$ be the largest value of ρ for which λ^* is an optimal strategy for player I in the game with payoff matrix.

$$A + \rho \sum_{i=1}^{n} \frac{x_i}{\|x_i\|} M_i.$$

If this number is not finite, define $\rho(x)$ to be $+\infty$. Then define

$$\rho^* = \min_{\|x\| = 1} \rho(x).$$

We may call ρ^* the radius of stability of A.

* Received by the editors June 15, 1987; accepted for publication (in revised form) October 18, 1989.

† Department of Management, 215 Rehn Hall, Southern Illinois University, Carbondale, Illinois 62901 (GA0435@SIUCVMB).

[1] Alternate notation: Let e be the $1 \times n$ matrix $[1, 1, \cdots, 1]$ and $[x]$ be the $1 \times n$ matrix $[x_1, x_2, \cdots, x_n]$. Then $\sum x_i M(i)$ may be written as $[x]^T e$.

DEFINITION 3. Let $A_{i,j|k,l}$ be the matrix A with rows i and k, and columns j and l deleted. Denote by $d_{i,k|j,l}$ the determinant det $A_{i,j|k,l}$. Define

$$D = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Cof } A_{ij},$$

where Cof denotes cofactor of, and

$$c_i^j = \frac{1}{D}(-1)^{i+j} \sum_{k \neq i} \sum_{l \neq j} (-1)^{k+l} d_{k,i|j,l}.$$

Note $D \neq 0$ here.

Finally, let C^j be the matrix with entries

$$C_{i,k}^j = \frac{1}{\mu_j^2} c_i^j c_k^j = \frac{1}{\mu_j^2} c^j (c^j)^T,$$

where $\hat{\mu}$ is the optimal strategy vector for player II. That is, $\hat{\mu}$ is the optimal probability for selections of the pure strategy column j by Player II.

Hence C^j is positive definite and symmetric and has only positive eigenvalues. The main result is the following.

THEOREM 2. *For a completely mixed game,* $\rho^* = (\max_j e_j)^{-1/2}$, *where* e_j *denotes the largest eigenvalue of matrix* C^j.

Before completing the proof of this theorem, two lemmas are obtained.

LEMMA 1. *If* $A(x)$ *is completely mixed then*

$$\mu_j(x) = \mu_j + \sum_{i=1}^{n} x_i c_{ij}.$$

Proof. This result follows easily by expression of the μ_j in terms of cofactors. □

LEMMA 2.

$$(\rho^*)^2 = \min_{\|x\|=1} \quad \min_j \left(\frac{\mu_j}{\sum_{i=1}^{n} x_i c_{ij}} \right)^2.$$

Proof. It is easy to see that

$$\rho^* = \min_{\|x\|=1} \quad \max \{ \rho : \mu_j(x) \geqq 0, j = 1 \quad \text{to } n \}.$$

Hence

$$(\rho^*) = \min_{\|x\|=1} \max \{ \rho^2 : \mu_j(\rho x) \geqq 0; j = 1 \quad \text{to } n \}$$

$$= \min_{\|x\|=1} \max \{ \rho^2 : \mu_j + \rho \sum_{i=1}^{n} x_i c_{ij} \geqq 0, j = 1 \quad \text{to } n \}.$$

Clearly for α sufficiently large, and taking $x = (\alpha, 0, \cdots, 0)$, the optimal for player I in the game with matrix $A + \sum_i^n x_i M_i$ will be $\lambda^* = (1, 0, \cdots, 0)$. Hence ρ^* is finite. Also $\rho^* \geqq 0$. Now $\min_{\|x\|=1} \max \{ \rho : \mu_j + \rho \sum x_i c_{ij} \geqq 0 \}$ must be attained for an x for which $\sum_i x_i c_{ij} < 0$ for at least one j, otherwise the positivity of the μ_j and ρ^* would provide a contradiction. For a given x, ρ may be increased until the first of the $\mu_j + \rho \sum_{i=1} x_i c_{ij}$ achieves its zero at

$$\frac{-\mu_j}{\sum_{i=1}^{n} x_i c_{ij}}.$$

Thus evidently for a given $x \neq 0$,

$$\max \{\rho: \mu_j + \rho \sum_1^n x_i c_{ij} \geqq 0, j \geqq 1 \text{ to } n\}$$

$$= \min \frac{-\mu_j}{\sum x_i c_{ij}},$$

where the minimum is over those j for which

$$\sum_1^n x_i c_{ij} < 0.$$

Since x and $-x$ are candidates for the minimum point, and the last relation is true for all $x \neq 0$, we obtain

$$\rho^* = \min_{\|x\|=1} \min_j \left| \frac{\mu_j}{\sum_{i=1}^n x_i c_{ij}} \right|.$$

Hence

$$(\rho^*)^2 = \min_{\|x\|=1} \min_j \left(\frac{\mu_j}{\sum_{i=1}^n x_j c_{ij}} \right)^2$$

and the lemma is proved. □

To complete the proof of the Theorem, note that Lemma 2 implies

$$\left(\frac{1}{\rho^*}\right)^2 = \max_j \max_{\|x\|=1} \left(\frac{\sum_{i=1}^n x_i c_{ij}}{\mu_j} \right)^2$$

$$= \max_j \max_{\|x\|=1} \sum_{k=1}^n \sum_{i=1}^n x_i x_k C_{ik}^j.$$

It is well known that $\max_{\|x\|=1} \sum_{k=1}^m \sum_{i=1}^m x_i x_k C_{ik}^j$ is the largest eigenvalue of C_{ik}^j. Let us denote this eigenvalue as e_j. Now

$$\left(\frac{1}{\rho^*}\right)^2 = \max_j e_j,$$

and hence

$$\rho^* = (\max_j e_j)^{-1/2}.$$

This concludes the proof of Theorem 2. □

The above result provides an alternative computational formula to that of the following theorem also obtained in [7].

THEOREM 3.

$$\rho^* = \min_j \mu_j^* \| c^j \|^{-1},$$

where c^j *is the* j*th row of* $A^{-1} - \dfrac{\mu^* \lambda^*}{v}$, λ^* *and* μ^* *are optimal strategies for players* I *and* II *respectively, and* v *is the value of the game* (*here* λ^* *is* $1 \times n$ *and* μ^* *is* $n \times 1$ *so that* $\mu^* \lambda^*$ *is* $n \times n$).

Remarks. It is not known whether Theorem 2 or Theorem 3 provides the most efficient computational formula. Interest in Theorem 2 is in providing a further connection

between games and eigensystems. Clearly a translation that adds the same number to each element of every column preserves stability. Hence ρ^* is actually the radius of a cylinder in which stability holds. It should also be noted that in [8], Filar raised the question of characterizing the size of such perturbations and gave a cubical neighborhood result [8, Lem. 3.3] independent of eigenvalue considerations.

Acknowledgments. The author thanks the editor and an anonymous referee for many useful suggestions, especially calling attention to the Filar reference.

REFERENCES

[1] R. L. WEIL, *Game theory and eigensystems*, SIAM Rev., 10 (1968), pp. 360–367.

[2] G. S. L. THOMPSON AND R. L. WEIL, *Further relations between game theory and eigensystems*, Management Science Report 136, Management Science Department, Carnegie Mellon University, Pittsburgh, PA 1968.

[3] T. E. S. RAGHAVAN, *On positive game matrices and their extensions*, J. London Math. Soc., 40 (1965), pp. 467–477.

[4] ———, *Completely mixed games and M-matrices*, Linear Algebra Appl., 21 (1978), pp. 35–45.

[5] J. E. COHEN AND M. FRIEDLAND, *The game theoretic value and spectral radius of a nonnegative matrix*, preprint, Department of Mathematics, Rockefeller University, New York, NY, February 16, 1984.

[6] J. E. COHEN, *Perturbation theory of completely mixed matrix games*, preprint, Department of Mathematics, Rockefeller University, New York, NY, August 20, 1984.

[7] M. TROUTT, *A stability concept for matrix game optimal strategies and its application to linear programming sensitivity analysis*, Math. Programming, 36 (1986), pp. 353–361.

[8] J. A. FILAR, *Semi-antagonistic equilibrium points and action costs*, Cahiers Centre Études Rech. Oper., 26 (1984), pp. 227–239.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 373–380, July 1990

004

AFFINE PSEUDOMONOTONE MAPPINGS AND THE LINEAR COMPLEMENTARITY PROBLEM*

M. SEETHARAMA GOWDA†

Abstract. In this article, it is shown that for an affine pseudomonotone mapping, the feasibility of the (linear) complementarity problem implies its solvability. A result of this type was proved earlier by Karamardian under a strict feasibility condition.

Key words. pseudomonotone, copositive, Linear Complementarity Problem, Lemke's algorithm

AMS(MOS) subject classification. 90C33

1. Introduction. In the theory of Linear Complementarity Problems, the class of positive semidefinite matrices has played a prominent role. It is well known that the positive semidefiniteness of a matrix M is equivalent to the monotonicity of the affine mapping $x \mapsto Mx + q$. In [7], Karamardian introduced the class of (nonlinear) pseudomonotone mappings. In this article, we establish a connection between affine pseudomonotone mappings and the Linear Complementarity Problem. Given a real $n \times n$ matrix M and a vector $q \in \mathbb{R}^n$, we say that the mapping $x \mapsto Mx + q$ is *pseudomonotone* (on $\mathbb{R}^n_+$) if

$$x, y \geqq 0, (y-x)^T(Mx+q) \geqq 0 \Rightarrow (y-x)^T(My+q) \geqq 0.$$

The *Linear Complementarity Problem* corresponding to the pair (M, q), denoted by LCP(M, q), is to find a vector $x \in \mathbb{R}^n$ such that

$$x \geqq 0, Mx+q \geqq 0, \quad \text{and} \quad x^T(Mx+q) = 0.$$

LCP(M, q) is said to be *feasible* (strictly feasible) if there is an $x \geqq 0$ such that $Mx + q \geqq 0$ ($Mx + q > 0$).

A result of Karamardian [7] (when stated for affine mappings) says that pseudomonotonicity of $x \mapsto Mx + q$ and strict feasibility of LCP(M, q) imply the solvability of LCP(M, q). In this article we extend this result by replacing the strict feasibility condition by the ordinary feasibility condition. We further show that, when M has no zero column, pseudomonotonicity and feasibility for a single q imply the solvability of all feasible LCPs. We also show that LCP(M, q) can be solved by Lemke's algorithm [8]. From pseudomonotonicity and feasibility, we deduce several interesting properties of the matrix M.

2. Preliminaries. The (usual) inner product of two vectors x and y in $\mathbb{R}^n$ is denoted by x^Ty. $\mathbb{R}^n_+$ denotes the nonnegative orthant in $\mathbb{R}^n$, and we write $z \geqq 0$ when $z \in \mathbb{R}^n_+$. For any $\lambda \in \mathbb{R}$, $\lambda^+ := \max\{\lambda, 0\}$ and $\lambda^- = (-\lambda)^+$. For any $z = (z_1, \cdots, z_n)^T \in \mathbb{R}^n$, $z^+ := (z_1^+, \cdots, z_n^+)^T$ and $z^- := (-z)^+$. Clearly $z = z^+ - z^-$ and $(z^+)^Tz^- = 0$. For $1 \leqq i \leqq n$, e_i denotes the ith coordinate vector (containing 1 at the ith spot and zero elsewhere). $\mathbb{R}^{n\times n}$ denotes the set of all $n \times n$ real matrices. For an $M \in \mathbb{R}^{n\times n}$, M^T denotes the transpose.

* Received by the editors May 16, 1988; accepted for publication (in revised form) June 10, 1989.

† Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, Maryland 21228. (GOWDA@UMBC and GOWDA@UMBC1.UMBC.EDU).

A matrix M is said to be

(a) *copositive* if $x^TMx \geqq 0\ \forall x \geqq 0$,
(b) *positive semidefinite* if $z^TMz \geqq 0\ \forall z \in \mathbb{R}^n$,
(c) *pseudomonotone* if $x, y \geqq 0$, $(y-x)^TMx \geqq 0 \Rightarrow (y-x)^TMy \geqq 0$,
(d) a $\mathbf{P}_0$-*matrix* (or $M \in \mathbf{P}_0$) if every principal minor of M is nonnegative,
(e) a *row sufficient matrix* (cf. [3]) if for any z,

$$z_i(M^Tz)_i \leqq 0\,(i=1,2,\cdots,n) \Rightarrow z_i(M^Tz)_i = 0\,(i=1,2,\cdots,n),$$

(f) a *column sufficient matrix* if M^T is row sufficient, and
(g) a $\mathbf{Q}_0$-*matrix* (or $M \in \mathbf{Q}_0$) if for any q, feasibility of LCP(M, q) implies its solvability.

The notion of pseudomonotonicity of a map can be defined on any cone [6], [7]. However, in this article, pseudomonotonicity is studied only on $\mathbb{R}^n_+$. We note that M is pseudomonotone if and only if the mapping $x \mapsto Mx$ is pseudomonotone.

3. General results.

THEOREM 1. *Suppose that* LCP(M, q) *is feasible and that the mapping* $x \mapsto Mx + q$ *is pseudomonotone. Then* M *is copositive and belongs to* $\mathbf{P}_0$.

Proof. Let $x_0 \geqq 0$ be such that $Mx_0 + q \geqq 0$. For any $x \geqq 0$, we have

$$\{(x_0+x)-x_0\}^T(Mx_0+q) \geqq 0$$

and by pseudomonotonicity

$$x^T\{M(x+x_0)+q\} \geqq 0.$$

Since $x \geqq 0$ is arbitrary, we get $x^TMx \geqq 0$. Thus M is copositive. To show that M is a $\mathbf{P}_0$-matrix, we show that M^T is a $\mathbf{P}_0$-matrix. In view of a result of Fiedler and Pták [5, Thm. 1.3], it is enough to show that for any nonzero z, $\max_{z_i \neq 0} z_i(M^Tz)_i \geqq 0$. Assume that for some nonzero z,

$$\max_{z_i \neq 0} z_i(M^Tz)_i < 0.$$

Since M is copositive, the sets $I = \{i : z_i > 0\}$ and $J = \{j : z_j < 0\}$ are nonempty. Let λ be a large positive number such that

$$z^Tq - \lambda \sum_J z_j(M^Tz)_j > 0 \quad \text{and} \quad z^Tq + \lambda \sum_I z_i(M^Tz)_i < 0.$$

Then the inequalities

$$(\lambda z^+ - \lambda z^-)^T\{M(\lambda z^-)+q\} = \lambda^2(z^-)^TM^Tz + \lambda z^Tq$$
$$= \lambda\left[z^Tq - \lambda \sum_J z_j(M^Tz)_j\right] > 0$$

and

$$(\lambda z^+ - \lambda z^-)^T\{M(\lambda z^+)+q\} = \lambda\left[z^Tq + \lambda \sum_I z_i(M^Tz)_i\right] < 0$$

contradict the pseudomonotonicity assumption. Hence M is a $\mathbf{P}_0$-matrix. □

Remark. The above result may not hold if the feasibility condition is dropped. For example, if we let

$$M = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

then the mapping $x \mapsto Mx + q$ is pseudomonotone while M is neither copositive nor a $\mathbf{P}_0$-matrix. We shall see later that if the mapping $x \mapsto Mx$ is pseudomonotone, i.e., if M is a pseudomonotone matrix, then M is a row sufficient matrix. Such a result fails for general affine pseudomonotone mappings. To see this, we let

$$M = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Then the mapping $x \mapsto Mx + q$ is pseudomonotone and LCP(M, q) is feasible, whereas M is neither row sufficient nor column sufficient.

We note that in the above theorem, we can replace feasibility by copositivity and then deduce that M is a $\mathbf{P}_0$-matrix from the pseudomonotonicity of the mapping $x \mapsto Mx + q$. This observation gives the following corollary.

COROLLARY 1. *Suppose that M is symmetric. Then the pseudomonotone mapping $x \mapsto Mx + q$ is monotone (i.e., M is positive semidefinite) if and only if M is copositive.*

Proof. The assertion follows from a well-known fact that a symmetric $\mathbf{P}_0$-matrix is positive semidefinite. □

When M is symmetric, the pseudomonotonicity of $x \mapsto Mx + q$ is equivalent (cf. [7]) to the pseudoconvexity of the quadratic function $Q(x) = x^T(Mx + q)$ on $\mathbb{R}^n_+$. It follows from Theorem 1 that, when M is symmetric, the pseudoconvex quadratic function $Q(x)$ is nonconvex on $\mathbb{R}^n_+$ only when LCP(M, q) is infeasible. From Corollary 1, we see that on $\mathbb{R}^n_+$ a pseudoconvex quadratic function $Q(x) = x^T(Mx + q)$ is convex if and only if M is copositive. (For more results concerning pseudoconvex quadratic functions, we refer the reader to the articles by Cottle and Ferland [2], Ferland [4], Schaible [9], and to the references therein.)

Our next result (Theorem 2) leads to a characterization of pseudomonotone matrices. It turns out that a matrix is pseudomonotone if and only if the corresponding quadratic form is nonnegative on a certain (nonzero) set. First we prove a simple lemma.

LEMMA 1. *Let $a \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$. Let*

$$I = \{i : a_i > 0\}, \qquad J = \{i : a_i < 0\},$$

$$U(\alpha) = \{u : u \geqq 0, u^T a \geqq \alpha\}, \qquad U(\beta) = \{u : u \geqq 0, u^T a \geqq \beta\}.$$

Then $U(\alpha) \subseteq U(\beta)$ if and only if the following hold:

(i) *$\alpha \geqq \beta$ when $I \neq \varnothing$ and $\alpha \geqq 0$,*

(ii) *$\alpha \geqq \beta$ when $J \neq \varnothing$ and $\alpha \leqq 0$,*

(iii) *$\beta \leqq 0$ when $\alpha \leqq 0$.*

Proof. Suppose that $U(\alpha) \subseteq U(\beta)$ and let i be an index such that either $a_i > 0$ and $\alpha \geqq 0$ or $a_i < 0$ and $\alpha \leqq 0$. Then $u = (\alpha / a_i)e_i$ belongs to $U(\alpha)$ and hence to $U(\beta)$. Therefore $\alpha = u^T a \geqq \beta$ so that (i) and (ii) hold. If $\alpha \leqq 0$, then $0 \in U(\alpha) \subseteq U(\beta)$ so that $\beta \leqq 0$. This gives (iii). To see the converse, let $u \in U(\alpha)$, i.e., $u \geqq 0$ and $u^T a \geqq \alpha$. If $a = 0$, then $0 \geqq \alpha$. From (iii), we get $0 \geqq \beta$, i.e., $u \in U(\beta)$. So let $a \neq 0$. We consider two cases.

Case 1. $\alpha \geqq 0$.

(a) If $a_i > 0$ for some i, then by (i), $\alpha \geqq \beta$. Now $u^T a \geqq \alpha \geqq \beta$ gives $u \in U(\beta)$.

(b) If $a \leqq 0$, then $u^T a \geqq \alpha \geqq 0$ implies $\alpha = 0$. From (iii), $\beta \leqq 0$ and so $u^T a \geqq \alpha = 0 \geqq \beta$. Hence $u \in U(\beta)$.

Case 2. $\alpha < 0$. From (iii) we have $\beta \leqq 0$.

(a) If $a \geqq 0$, then $u^T a \geqq 0 \geqq \beta$ and hence $u \in U(\beta)$.

(b) If $a_i < 0$ for some i, then (ii) holds and hence $u^T a \geqq \alpha \geqq \beta$. Therefore $u \in U(\beta)$. □

The definition of pseudomonotonicity of an affine map on $\mathbb{R}^n$ involves *two* variables in $\mathbb{R}^n_+$. The following theorem shows that the pseudomonotonicity can be described in terms of a *single* variable in $\mathbb{R}^n$. For the pair (M, q), we let

$$A := \{z : (M^T z)_i > 0 \quad \text{for some} \quad i \quad \text{and} \quad z^T(Mz^- + q) \leqq 0\}, \tag{3.1}$$

$$B := \{z : (M^T z)_i < 0 \quad \text{for some} \quad i \quad \text{and} \quad z^T(Mz^- + q) \geqq 0\}, \tag{3.2}$$

$$C := \{z : z^T(Mz^- + q) \geqq 0\}, \qquad D := \{z : z^T(Mz^+ + q) \geqq 0\}. \tag{3.3}$$

THEOREM 2. *The mapping $x \mapsto Mx + q$ is pseudomonotone if and only if*
(a) $z^T Mz \geqq 0$ *for all* $z \in A \cup B$,
(b) $C \subseteq D$.

Proof. We notice that the condition

$$x \geqq 0, y \geqq 0, (y-x)^T(Mx+q) \geqq 0 \Rightarrow (y-x)^T(My+q) \geqq 0$$

is equivalent to

$$u \geqq 0, \qquad z \in \mathbb{R}^n, \qquad z^T\{M(z^- + u) + q\} \geqq 0 \Rightarrow z^T\{M(z^+ + u) + q\} \geqq 0$$

which is the same as

$$u \geqq 0, \qquad z \in \mathbb{R}^n, \qquad u^T M^T z \geqq -z^T(Mz^- + q) \Rightarrow u^T M^T z \geqq -z^T(Mz^+ + q).$$

But this amounts to saying that for every z, the implication

$$u \geqq 0, u^T a \geqq \alpha \Rightarrow u^T a \geqq \beta$$

holds, where $a = M^T z$, $\alpha = -z^T(Mz^- + q)$, $\beta = -z^T(Mz^+ + q)$. Equivalently, for every z, conditions (i)–(iii) of Lemma 1 hold. We note that $\alpha \geqq \beta$ is equivalent to $z^T Mz \geqq 0$. Hence, upon rewriting conditions (i)–(iii) of Lemma 1 in terms of z, we get conditions (a) and (b) of the theorem. □

It is clear that if M is positive semidefinite, then for any q, the mapping $x \mapsto Mx + q$ is pseudomonotone. It turns out that even the converse is true.

COROLLARY 2. *A matrix M is positive semidefinite if and only if for every q, the mapping $x \mapsto Mx + q$ is pseudomonotone.*

Proof. We prove only the sufficiency part. Let z be any vector. If $M^T z = 0$, then $z^T Mz = 0$. Suppose that $M^T z \neq 0$ and let $q = -Mz^-$. Then $Mz^- + q = 0$ and hence $z \in A \cup B$, where A and B are as in (3.1) and (3.2). From Theorem 2 we get $z^T Mz \geqq 0$. □

We see below that when $q = 0$, the inclusion $C \subseteq D$ in Theorem 2 is superfluous.

COROLLARY 3. *A matrix M is pseudomonotone if and only if $z^T Mz \geqq 0$ for all $z \in A' \cup B'$, where*

$$A' = \{z : (M^T z)_i > 0 \quad \textit{for some } i \quad \textit{and} \quad z^T Mz^- \leqq 0\}, \tag{3.4}$$

$$B' = \{z : (M^T z)_i < 0 \quad \textit{for some } i \quad \textit{and} \quad z^T Mz^- \geqq 0\}. \tag{3.5}$$

Proof. We prove (only) the sufficiency part by proving the inclusion $C \subseteq D$ (for $q = 0$). Let $z \in C$. Then $z^T Mz^- \geqq 0$. If $M^T z \geqq 0$, then $z^T Mz^+ \geqq 0$. If $(M^T z)_i < 0$ for some i, then $z \in B'$ and so $z^T Mz^+ - z^T Mz^- = z^T Mz \geqq 0$. Therefore, $z^T Mz^+ \geqq z^T Mz^- \geqq 0$, i.e., $z \in D$. □

COROLLARY 4. *If M is pseudomonotone, then M is a row sufficient matrix.*

Proof. Suppose that $z_k(M^T z)_k \leqq 0$ for $k = 1, 2, \cdots, n$. Let $I = \{i : z_i > 0\}$ and $J = \{j : z_j < 0\}$. Since LCP$(M, 0)$ is feasible, by Theorem 1, M is copositive. Thus, if I or J is empty, then $z^T M^T z \geqq 0$, which gives $z_i(M^T z)_i = 0$ for $i = 1, 2, \cdots, n$.

So we can assume that I and J are nonempty. Since $z^T Mz^- = -\sum_J z_j(M^T z)_j \geqq 0$, either from Theorem 2(b) or directly from the definition of pseudomonotonicity, we have $z^T Mz^+ \geqq 0$, i.e., $\sum_I z_i(M^T z)_i \geqq 0$. This yields $z_i(M^T z)_i = 0 \; \forall i \in I$. Replacing z by $-z$ and repeating the above argument we get $z_j(M^T z)_j = 0 \; \forall j \in J$. Hence $z_k(M^T z)_k = 0$ for all k. □

THEOREM 3. *Suppose that M has no zero column. If the map $x \mapsto Mx + q$ is pseudomonotone and* LCP(M, q) *is feasible, then M is pseudomonotone.*

Proof. In view of Corollary 3, it is enough to show that $z^T Mz \geqq$ for all $z \in A' \cup B'$.

(a) Let $z \in A'$ so that $(M^T z)_i > 0$ for some i and $z^T Mz^- \leqq 0$.

Case 1. $z^T Mz^- < 0$. In this case, choose a large positive λ such that

$$(\lambda z)^T \{M(\lambda z^-) + q\} < 0.$$

By Theorem 2(a), we have $(\lambda z)^T M(\lambda z) \geqq 0$, i.e., $z^T Mz \geqq 0$.

Case 2. $z^T Mz^- = 0$. Suppose, if possible, that $z^T Mz < 0$. Then $z^T Mz^+ < 0$. We put $w = -z$ and observe that $(M^T w)_i < 0$ and $w^T Mw^- > 0$. Using a suitable λ we get $[M^T(\lambda w)]_i < 0$ and $(\lambda w)^T \{M(\lambda w^-) + q\} > 0$. By Theorem 2(a), we get $\lambda^2 w^T Mw \geqq 0$, i.e., $w^T Mw \geqq 0$. Hence $z^T Mz \geqq 0$.

(b) Let $z \in B'$ so that $(M^T z)_i < 0$ for some i and $z^T Mz^- \geqq 0$.

Case 1. $z^T Mz^- > 0$. We can proceed as in Case 1 of (a) and get $z^T Mz \geqq 0$.

Case 2. $z^T Mz^- = 0$. If $z^T q \geqq 0$, then $z^T(Mz^- + q) \geqq 0$ and hence (by Theorem 2(a)) $z^T Mz \geqq 0$. So we can assume that $z^T q < 0$. Thus

$$(M^T z)_i < 0 \quad \text{for some } i, \quad z^T Mz^- = 0 \quad \text{and} \quad z^T q < 0.$$

Subcase (i). $(M^T z)_j > 0$ for some j. In this case, we can use Theorem 2(a) to get $z^T Mz \geqq 0$.

Subcase (ii). $M^T z < 0$. In this case, $z^T Mz^- = 0$ gives $z^- = 0$, i.e., $z \geqq 0$. By copositivity of M (cf. Theorem 1), $z^T Mz \geqq 0$.

Subcase (iii). $(M^T z)_j = 0$ for some j. We find an $e \in \mathbb{R}^n$ such that $(M^T e)_j > 0$. (This is true, since M has no zero column.) By continuity, we have for all small $\varepsilon > 0$, $[M^T(z + \varepsilon e)]_j = \varepsilon(M^T e)_j > 0$ and

$$(z + \varepsilon e)^T \{M(z + \varepsilon e)^- + q\} < 0.$$

From Theorem 2(a), we get $(z + \varepsilon e)^T M(z + \varepsilon e) \geqq 0$ for all small $\varepsilon > 0$. Hence $z^T Mz \geqq 0$. □

Remark. We note that in the above theorem, the feasibility condition can be replaced by the copositivity condition. If M has a zero column, then the conclusion of the theorem cannot be drawn. For example, let

$$M = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

It is easily seen that the map $x \mapsto Mx + q$ is pseudomonotone and LCP(M, q) is feasible. But M is not pseudomonotone, since $(e_2 - e_1)^T Me_1 = 0$ and $(e_2 - e_1)^T Me_2 < 0$. Also, we cannot conclude the pseudomonotonicity of $x \mapsto Mx + q$ from that of M (even if LCP(M, q) is feasible). To see this, let

$$M = \begin{bmatrix} 0 & -1 \\ 2 & 0 \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} 1 \\ -2 \end{bmatrix}.$$

Then, for $x, y \geqq 0$, $(y - x)^T Mx = 2x_1 y_2 - x_2(x_1 + y_1) \geqq 0$, we have $(y - x)^T My =$

$y_2(x_1 + y_1) - 2x_2y_1 \geqq 0$ where x_1 and x_2 are, respectively, the first and second components of x, etc. (If $x_1 = 0$, the implication is easy to see; otherwise, $(y - x)^T My \geqq (2x_1)^{-1}x_2(x_1 + y_1)^2 - 2x_2y_1 \geqq 0$.) The mapping $x \mapsto Mx + q$ is not pseudomonotone, since $(y - x)^T(Mx + q) = 0$ and $(y - x)^T(My + q) = -1$, where

$$x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad y = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

4. Complementarity results.

THEOREM 4. *Suppose that M is pseudomonotone. Then $M \in \mathbf{P}_0 \cap \mathbf{Q}_0$ and every feasible* LCP(M, q) *is solvable by Lemke's algorithm.*

Proof. If M is pseudomonotone, then M is a row sufficient matrix (by Corollary 4). By Theorem 4 and subsequent remarks in [3], $M \in \mathbf{P}_0 \cap \mathbf{Q}_0$. Hence, by the main theorem in [1], every feasible LCP(M, q) is solvable by Lemke's algorithm. □

Remark. A proof for the above theorem (which avoids the sufficiency property) can be given. Lemke [8, p. 104] showed (but did not state the result this way) that *if M is copositive and*

$$x \geqq 0, \qquad Mx \geqq 0, \qquad x^TMx = 0 \Rightarrow x^Tq \geqq 0 \tag{4.1}$$

then LCP(M, q) *is solvable by Lemke's algorithm.* Now let M be pseudomonotone and LCP(M, q) be feasible. Since LCP$(M, 0)$ is feasible, by Theorem 1, $M \in \mathbf{P}_0$ and is copositive. Let $x \geqq 0$, $Mx \geqq 0$, $x^TMx = 0$. From $(u - x)^TMx \geqq 0$ $(\forall u \geqq 0)$ and pseudomonotonicity we get $M^Tx \leqq 0$. Now feasibility of LCP(M, q) gives $q = v - Mu$ for some u, $v \geqq 0$. We have $x^Tq = v^Tx - u^TM^Tx \geqq 0$. Thus (4.1) holds and so LCP(M, q) is solvable.

The following result is immediate from Theorems 3 and 4.

THEOREM 5. *Suppose that M has no zero column. If* LCP(M, q) *is feasible and the mapping $x \mapsto Mx + q$ is pseudomonotone, then $M \in \mathbf{P}_0 \cap \mathbf{Q}_0$ and every feasible* LCP(M, q') *is solvable (by Lemke's algorithm).*

In order to prove the next theorem, we need

LEMMA 2. *Suppose that*

$$M = \begin{bmatrix} 0 & A \\ 0 & B \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

where $A \in \mathbb{R}^{m \times k}$, $B \in \mathbb{R}^{k \times k}$, $q_1 \in \mathbb{R}^m$, $q_2 \in \mathbb{R}^k$, and $n = m + k$. Let

$$x \in \mathbb{R}^m, \quad y \in \mathbb{R}^k, \quad \text{and} \quad z = \begin{bmatrix} x \\ y \end{bmatrix}.$$

If the mapping $z \mapsto Mz + q$ is pseudomonotone and LCP(M, q) *is feasible, then*

(i) *A is a nonnegative matrix,*

(ii) *$q_1 \geqq 0$,*

(iii) *$y \mapsto By + q_2$ is pseudomonotone and* LCP(B, q_2) *is feasible,*

(iv) *if $\bar{y}$ solves* LCP(B, q_2), *then $\bar{z}$ solves* LCP(M, q), *where*

$$\bar{z} = \begin{bmatrix} 0 \\ \bar{y} \end{bmatrix}.$$

Proof. (i) Let $1 \leqq i \leqq m$. Since $Me_i = 0$, copositivity (which follows from Theorem 1) implies $(M + M^T)e_i \geqq 0$. Hence every row vector in A is nonnegative.

(ii) If any row of A is zero, feasibility of LCP(M, q) implies that the component of q corresponding to that row is nonnegative. Without loss of generality, let the first row

of A be nonzero. We show that the first component $q_1^{(1)}$ of q_1 is nonnegative. We write

$$M=\begin{bmatrix}0 & a\\ 0 & C\end{bmatrix}, \qquad q=\begin{bmatrix}q_1^{(1)}\\ \bar{q}\end{bmatrix}$$

where $a \in \mathbb{R}^{1\times(n-1)}$, $C \in \mathbb{R}^{(n-1)\times(n-1)}$, etc. Assume, if possible, that $q_1^{(1)} < 0$. Let e be the vector of one's in $\mathbb{R}^{(n-1)}$. Put $u = (-q_1^{(1)}/e^T a)e$,

$$\xi=\begin{bmatrix}\mu\\ v\end{bmatrix}, \quad \text{and} \quad \eta=\begin{bmatrix}\lambda\\ u\end{bmatrix}$$

where $\lambda \geqq 0$, $\mu \geqq 0$ (in $\mathbb{R}$), $v \geqq 0$ in $\mathbb{R}^{n-1}$. We have

$$\begin{aligned}(\xi-\eta)^T(M\eta+q) &= [u^T a+q_1^{(1)}](\mu-\lambda)+(v-u)^T(Cu+\bar{q})\\ &= (v-u)^T(Cu+\bar{q}) \text{(by the choice of } u).\end{aligned}$$

Now choose $v \geqq 0$ such that $(v-u)^T(Cu+\bar{q}) \geqq 0$ and $v^T a + q_1^{(1)} \neq 0$. (If $b := Cu + \bar{q}$ is zero put $v = 0$; otherwise, we find w such that $w^T b \geqq 0$ and $w^T a \neq 0$ and then put $v = u + \varepsilon w$ where ε is a small positive number.)

By pseudomonotonicity, $(\xi-\eta)^T(M\xi+q) \geqq 0$, i.e.,

$$[v^T a+q_1^{(1)}](\mu-\lambda)+(v-u)^T(Cv+\bar{q})\geqq 0.$$

Since $v^T a + q_1^{(1)} \neq 0$ and $\mu - \lambda$ is arbitrary, we reach a contradiction (to the pseudomonotonicity). Hence $q_1^{(1)} \geqq 0$.

(iii) This follows from

$$(v-y)^T(By+q_2)=\left\{\begin{bmatrix}0\\ v\end{bmatrix}-\begin{bmatrix}0\\ y\end{bmatrix}\right\}^T\left\{M\begin{bmatrix}0\\ y\end{bmatrix}+\begin{bmatrix}q_1\\ q_2\end{bmatrix}\right\}$$

and

$$\begin{bmatrix}Ay_0+q_1\\ By_0+q_2\end{bmatrix}=\begin{bmatrix}0 & A\\ 0 & B\end{bmatrix}\begin{bmatrix}x_0\\ y_0\end{bmatrix}+\begin{bmatrix}q_1\\ q_2\end{bmatrix}.$$

(iv) Suppose that $\bar{y}$ solves LCP(M, q). Then

$$M\begin{bmatrix}0\\ \bar{y}\end{bmatrix}+\begin{bmatrix}q_1\\ q_2\end{bmatrix}=\begin{bmatrix}A\bar{y}+q_1\\ B\bar{y}+q_2\end{bmatrix}\geqq 0$$

(by (i) and (ii)), and $\bar{z}^T(M\bar{z}+q) = \bar{y}^T(B\bar{y}+q_2) = 0$. Hence $\bar{z}$ solves LCP(M, q). □

We now come to the main result.

THEOREM 6. *Suppose that the mapping $x \mapsto Mx + q$ is pseudomonotone and* LCP(M, q) *is feasible. Then* LCP(M, q) *is solvable (by Lemke's algorithm).*

Proof. If M has no zero column, then the result follows from Theorem 5. Suppose that M has a zero column. We show that (4.1) holds, i.e., $x^T q \geqq 0$ for all $x \geqq 0$ such that $Mx \geqq 0$ and $x^T Mx = 0$, and then use Lemke's result. (See the remark following Theorem 4.) Let B be a proper principal submatrix of largest size having no zero column. (If no such B exists, then, by Lemma 2, $q \geqq 0$ and in this case the zero vector solves LCP(M, q).) We note that the mapping $x \mapsto E^T MEx + Eq$ is pseudomonotone for any permutation matrix E, and solving LCP(M, q) is equivalent to solving LCP$(E^T ME, Eq)$. Thus we can assume without loss of generality that B occupies the lower right-hand corner of M. We can write

$$q=\begin{bmatrix}q_1\\ q_2\end{bmatrix}$$

where the size of q_2 agrees with that of B. Repeated use of Lemma 2 shows that $q_1 \geqq 0$. Also, LCP(B, q_2) is feasible and B is pseudomonotone (since B has no zero column). From the remarks following Theorem 4 (with B in place of M) we see that

$$y \geqq 0, \quad By \geqq 0, \quad y^T By = 0 \Rightarrow q_2^T y \geqq 0. \tag{4.2}$$

Now let $x \geqq 0$ such that $Mx \geqq 0$ and $x^T Mx = 0$. Writing

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(where the sizes of x_1 and x_2 agree with those of q_1 and q_2), we see that

$$x_2 \geqq 0, \quad Bx_2 \geqq 0, \quad \text{and} \quad x_2^T Bx_2 = 0.$$

From (4.2) we get $x_2^T q_2 \geqq 0$. Since $q_1 \geqq 0$, we get $x^T q = x_1^T q_1 + x_2^T q_2 \geqq 0$. This completes the proof. □

Remark. We note that the solvability also follows from Lemma 2 and an induction argument. The following example shows that stronger conclusions (such as the ones given in Theorem 5) cannot be drawn in Theorem 6. Let

$$M = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad q = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad q' = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

Then the mapping $x \mapsto Mx + q$ is pseudomonotone, LCP(M, q) and LCP(M, q') are feasible but LCP(M, q') is not solvable. In particular, $M \notin \mathbf{Q}_0$.

Acknowledgments. I am grateful to Professor J.-S. Pang and to the referees for their valuable suggestions.

REFERENCES

[1] M. AGANAGIĆ AND R. W. COTTLE, *A constructive characterization of Q_0-matrices with nonnegative principal minors*, Math. Programming, 37 (1987), pp. 223–231.

[2] R. W. COTTLE AND J. A. FERLAND, *Matrix-theoretic criteria for the quasi-convexity and pseudo-convexity of quadratic functions*, Linear Algebra Appl., 5 (1972), pp. 123–136.

[3] 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.

[4] J. A. FERLAND, *Quasiconvexity and pseudoconvexity of functions on the nonnegative orthant*, in Generalized Concavity in Optimization and Economics, S. Schaible and W. T. Ziemba, eds. North Atlantic Treaty Organization Advanced Study Institute, Academic Press, New York, 1981, pp. 169–181.

[5] M. FIEDLER AND V. PTÁK, *Some generalizations of positive definiteness and monotonicity*, Numer. Math., 9 (1966), pp. 163–172.

[6] M. S. GOWDA, *Pseudomonotone and copositive star matrices*, Linear Algebra Appl., 113 (1989), pp. 107–118.

[7] S. KARAMARDIAN, *Complementarity problems over cones with monotone and pseudomonotone maps*, J. Optim. Theory Appl., 18 (1976), pp. 445–454.

[8] C. E. LEMKE, *On complementary pivot theory*, in Mathematics of Decision Sciences, Part I, G. B. Dantzig and A. F. Veinott, Jr., eds. American Mathematical Society, Providence, RI, 1968, pp. 95–114.

[9] S. SCHAIBLE, *Generalized convexity of quadratic functions*, in Generalized Concavity in Optimization and Economics, S. Schaible and W. T. Ziemba, eds. North Atlantic Treaty Organization Advanced Study Institute, Academic Press, New York, 1981, pp. 183–197.

PREFACE TO THE SPECIAL SECTION ON SPARSE MATRICES

Computation with sparse matrices unites many different parts of applied mathematics, bringing together techniques from numerical analysis, combinatorics, and computer science.

The SIAM Activity Group on Linear Algebra sponsored a Symposium on Sparse Matrices at Salishan Resort in Gleneden Beach, Oregon, from May 22 through 24, 1989. John Lewis chaired the organizing committee, which also included Loyce Adams, David Scott, and Horst Simon. The Symposium advisory committee consisted of Iain Duff, Stan Eisenstat, Alan George, Gene Golub, Beresford Parlett, Ahmed Sameh, and Bob Ward. The committees composed a program of 16 plenary talks and 12 workshops containing 65 presentations. The topics covered the entire range of sparse matrix computation, from iterative to direct methods, from error analysis to graph theory, from engineering applications to standards to experimental comparisons among competing codes. A frequent theme was the emerging influence of large-scale parallel computation on sparse matrix methods.

The editors of the *SIAM Journal on Matrix Analysis and Applications* (SIMAX) and the conference organizers are pleased to present a selection of papers from the Symposium. Authors were invited to submit their papers to SIMAX, after which they underwent the usual SIAM refereeing process. This special section contains five papers, and several more will appear in forthcoming issues of SIMAX.

The organizers would like to thank the SIAM Activity Group on Linear Algebra, Intel Scientific Computer Corporation, and Cray Research for support of the Symposium.

J. G. Lewis
J. R. Gilbert
R. J. Plemmons
H. D. Simon

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 383–402, July 1990

005

A NONDETERMINISTIC PARALLEL ALGORITHM FOR GENERAL UNSYMMETRIC SPARSE LU FACTORIZATION*

TIMOTHY A. DAVIS† AND PEN-CHUNG YEW‡

Abstract. A parallel algorithm for the direct **LU** factorization of general unsymmetric sparse matrices is presented. The algorithm D2 is based on a new nondeterministic parallel pivot search that finds a *compatible pivot set* **S** of size m, followed by a parallel rank-m update. These two steps alternate until switching to dense matrix code or until the matrix is factored. The algorithm is based on a shared-memory multiple-instruction-multiple-data (MIMD) model and takes advantage of both concurrency and (gather-scatter) vectorization. The detection of parallelism due to sparsity is based on Markowitz's strategy, an unsymmetric ordering method. As a result, D2 finds more potential parallelism for matrices with highly asymmetric nonzero patterns than algorithms that construct an elimination tree using a symmetric ordering method (minimum degree or nested dissection, for example) applied to the symmetric pattern of $\mathbf{A} + \mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$. The pivot search exploits more parallelism than previous algorithms that are based on unsymmetric ordering methods. Possible extensions to the D2 algorithm are discussed, including the use of dense matrix kernels and a software-combining tree to enhance parallelism in the pivot search.

Key words. **LU** factorization, general unsymmetric sparse matrices, parallel algorithms, nondeterministic algorithms, synchronization techniques

AMS(MOS) subject classifications. 65F50, 65W05, 65F05, 65-04, 68-04

1. Introduction.

Structural analysis, computational fluid dynamics, economic modeling, chemical plant modeling, oil reservoir simulation, circuit and device simulation, electric power network modeling, and many other problems in science and engineering often require the numerical solution of a general unsymmetric system of sparse linear equations, $\mathbf{Ax} = \mathbf{b}$, where most of the elements in the matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ are zero, and where $\mathbf{b} \in \mathbb{R}^n$ is given and the unknown $\mathbf{x} \in \mathbb{R}^n$ is to be determined. The most common direct method for solving a system of linear equations is **LU** factorization, where **A** is first factored into the product of a lower triangular matrix **L** and an upper triangular matrix **U**, followed by forward and backward substitution to compute an estimate $\tilde{\mathbf{x}}$ of the true solution **x**. In the outer-product formulation of **LU** factorization, **A** is transformed into the product $\mathbf{L}^{(k)}\mathbf{A}^{(k)}\mathbf{U}^{(k)}$ after stage k ($1 \leqq k \leqq n$, $\mathbf{A}^{(0)} = \mathbf{A}$, and $\mathbf{A}^{(n)} = \mathbf{I}$). Each stage k selects a pivot $a_{ij}^{(k-1)}$, permutes the pivot row i with row k and the pivot column j with column k, and then applies a rank-one update (using the outer product of the pivot row and pivot column) to compute the lower right submatrix $\mathbf{A}_k \in \mathbb{R}^{(n-k)\times(n-k)}$ of $\mathbf{A}^{(k)}$. The notation $a_{ij}^{(k)}$ refers to the element in row i and column j of $\mathbf{A}^{(k)}$.

At least in a single-processor nonvector computer, the ideal ordering is usually the one with the least fill-in. However, finding the ordering with minimum fill-in is too difficult a problem to solve, so a heuristic must be used (it is probably NP-complete, see for example [35] and [40]). Markowitz's strategy for general unsymmetric matrices selects as pivot the element $a_{ij}^{(k-1)}$ with the lowest upper bound on the amount of fill-in

* Received by the editors September 11, 1989; accepted for publication (in revised form) December 29, 1989. The work of both authors was supported by the National Science Foundation (MIP-88-07775 and MIP-84-10110), the U.S. Department of Energy (DE-FG02-85ER25001), and Defense Advanced Research Projects Agency (NASA NCC 2-559).

† Center for Supercomputing Research and Development, 104 S. Wright St., University of Illinois, Urbana, Illinois 61801.

‡ European Center for Research and Advanced Training in Scientific Computation, 42 Avenue G. Coriolis, 31057 Toulouse Cedex, France (DAVIS@FRTLS12.BITNET). The work of this author was supported in part by a fellowship awarded by the American Electronics Association with funds from Digital Equipment Corporation.

that the pivot can cause, $(r_i^{(k-1)} - 1) \cdot (c_j^{(k-1)} - 1)$, where $r_i^{(k-1)}$ is the number of nonzeros in the ith row of $\mathbf{A}^{(k-1)}$ and $c_j^{(k-1)}$ is the number of nonzeros in the jth column of $\mathbf{A}^{(k-1)}$. The number of nonzeros in a row or column will also be referred to as the *length* of the row or column. Each pivot must satisfy a threshold pivoting constraint (a relaxed form of partial pivoting) [13],

$$|a_{kk}^{(k-1)}| \geqq u \cdot \max_{k \leqq j \leqq n} |a_{kj}^{(k-1)}|, \tag{1}$$

where $0 \leqq u \leqq 1$.

Orderings for symmetric matrices can be applied to the symmetric pattern of $\mathbf{A} + \mathbf{A}^{\mathbf{T}}$ or $\mathbf{A}^{\mathbf{T}}\mathbf{A}$, where $\mathbf{A}$ is a general unsymmetric matrix with either symmetric or unsymmetric nonzero pattern. The minimum degree algorithm by Tinney and Walker [37] is a special case of Markowitz's strategy. When the pivots are selected only from the diagonal of a symmetric matrix, then the symmetry is preserved during factorization and the length of row i ($r_i^{(k-1)}$) will always equal the length of column i ($c_i^{(k-1)}$). The pivot search is reduced to finding the diagonal pivot $a_{ii}^{(k-1)}$ with minimum $r_i^{(k-1)}$. One advantage over Markowitz's strategy is that it can preorder the matrix with only as much storage as the original matrix $\mathbf{A}$ [19]. In the undirected graph representation $G = (X, E)$ of a symmetric sparse matrix $\mathbf{A}$, each node in X corresponds to a diagonal nonzero in $\mathbf{A}$, and each edge from node i to node j in E corresponds to the nonzero a_{ij} in $\mathbf{A}$ [18]. The graphs associated with the matrices $\mathbf{A}_0, \mathbf{A}_1, \cdots, \mathbf{A}_n$ are called elimination graphs. Performing one stage of **LU** factorization to obtain $\mathbf{A}_k$ corresponds to removing a node from the elimination graph of $\mathbf{A}_{k-1}$ and adding fill-in edges so that the nodes adjacent to the pivot node k form a *clique* (that is, a set of nodes that are all adjacent to each other). A recent version of the minimum-degree algorithm by Liu [30] improves upon the earlier version by eliminating more than one node from the elimination graph between each update. George's nested dissection method [18] finds a set of nodes that split the graph representing the matrix into two disconnected subgraphs when removed from the graph. Each subgraph is recursively divided into smaller graphs. For some problems, nested dissection can result in shorter elimination trees than the minimum degree method, and it is particularly appropriate for finite-element problems [18].

The ordering schemes form the basis of many of the parallel algorithms summarized in § 2, which introduces the mathematical basis for the additional parallelism due to sparsity, which does not occur in the solution of dense linear systems, and summarizes the *elimination tree* and *compatible pivot set* concepts that describe the parallelism due to sparsity in a given matrix. An algorithm based on a symmetric ordering uses a symmetrized pattern of $\mathbf{A}$ (either $\mathbf{A} + \mathbf{A}^{\mathbf{T}}$ or $\mathbf{A}^{\mathbf{T}}\mathbf{A}$) during the phase (or phases) of the algorithm that determines the parallelism due to sparsity in the matrix (such as the construction of the elimination tree, for example). An unsymmetric ordering works with the unsymmetric pattern of $\mathbf{A}$ during all phases of the algorithm. A counterexample matrix is given which shows that methods using a symmetric ordering may exploit much less parallelism than is possible for matrices with highly asymmetric nonzero pattern. Previous algorithms that use an unsymmetric ordering avoid this dilemma, but most do not exploit parallelism in the pivot search. Section 3 presents a new algorithm, D2, which addresses some of the limitations of previous approaches by exploiting parallelism in all major phases of the factorization. It is based on the shared-memory multiple-instruction-multiple-data (MIMD) model, and on a new nondeterministic parallel pivot search heuristic that builds a compatible pivot set $\mathbf{S}$ of size m for a subsequent parallel rank-m update. Section 4 summarizes a series of experiments that compares the D2 algorithm with previous algorithms on an Alliant FX/8 and an Alliant VFX/80. Portions of the research discussed in these two sections have been previously reported [6], [8].

2. Parallelism due to sparsity. A sequential sparse factorization can improve over sequential dense factorization by skipping trivial operations, and further improvement can be achieved in a parallel sparse factorization. There are two types of parallelism to take advantage of in sparse **LU** factorization: the first is apparent in dense **LU** factorization, such as updating two rows of $\mathbf{A}^{(k)}$ for the same pivot in parallel, and the second is parallelism due to sparsity. The reductions for two pivots a_{kk} and a_{qq} can execute in parallel if $a_{qk} = a_{kq} = 0$ [4]. For example, consider the case of $q = k + 1$. Because

$$a_{k,k+1}^{(k-1)} = a_{k+1,k}^{(k-1)} = 0,$$

the first reduction does not affect column $k + 1$ or row $k + 1$ of $\mathbf{A}^{(k)}$, i.e.,

$$a_{i,k+1}^{(k)} = a_{i,k+1}^{(k-1)} \quad \text{for} \quad k+1 \leqq i \leqq n, \quad \text{and} \quad a_{k+1,j}^{(k)} = a_{k+1,j}^{(k-1)} \quad \text{for} \quad k+1 \leqq j \leqq n.$$

The second reduction,

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - (a_{k+1,j}^{(k)} \cdot a_{i,k+1}^{(k)} / a_{k+1,k+1}^{(k)}) \quad \text{for} \quad k+1 \leqq i \leqq n, k+1 \leqq j \leqq n,$$

can then be rewritten as

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - (a_{k+1,j}^{(k-1)} \cdot a_{i,k+1}^{(k-1)} / a_{k+1,k+1}^{(k-1)}) \quad \text{for} \quad k+2 \leqq i \leqq n, k+2 \leqq j \leqq n.$$

Substituting for $a_{ij}^{(k)}$ gives

$$\begin{aligned} a_{ij}^{(k+1)} &= a_{ij}^{(k-1)} - (a_{kj}^{(k-1)} \cdot a_{ik}^{(k-1)} / a_{kk}^{(k-1)}) - (a_{k+1,j}^{(k-1)} \cdot a_{i,k+1}^{(k-1)} / a_{k+1,k+1}^{(k-1)}) \\ &= a_{ij}^{(k-1)} + \text{update}_k + \text{update}_{k+1}. \end{aligned}$$

The two update terms, update_k and update_{k+1}, can be computed in parallel from $\mathbf{A}^{(k-1)}$, followed by additions to the $a_{ij}^{(k-1)}$ term to compute $\mathbf{A}^{(k+1)}$. The two additions cannot execute in parallel, but they can be computed in any order if variations in roundoff errors are neglected. This concept can be generalized to more than two pivots. An *independent* or *compatible pivot set* **S** is a set of m pivots whose update terms can be computed in parallel due to sparsity,

$$\mathbf{S} = \{ a_{jj}^{(k-1)} \mid a_{ij}^{(k-1)} = a_{ji}^{(k-1)} = 0, \quad \text{for} \quad k \leqq j \leqq m+k-1, k \leqq i \leqq m+k-1, \quad \text{and} \quad i \neq j \}.$$

The m pivots in a set **S** form a diagonal m-by-m leading submatrix of $\mathbf{A}_{k-m}$.

The parallelism due to sparsity in a particular matrix can also be described by the *elimination tree* associated with the matrix [31]. Elimination trees are usually associated only with symmetric matrices. There are n nodes in the tree for a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. Each node k of the tree is a single task representing, for example, the elimination of pivot k in the outer-product formulations, or the computation of column k in the column-Cholesky method. The tree represents the dependences in the parallel task ordering that must be satisfied to compute the factorization. In some algorithms, a parent task can start as soon as all its children have finished, whereas in others, a parent can start after its children start but must finish after its children finish (*pipelining*). The maxtrix **A** is first permuted according to the pivot sequence. The *parent* of node k is given by the location of the first off-diagonal nonzero in column k of the lower triangular factor **L**. The parent of a root node is nil. The *children* of node k are all nodes whose parent is node k. Since **A** has symmetric pattern, the patterns of **L** and **U** are identical, and the parent of node k is also given by the first off-diagonal nonzero in row k of **U**. That is,

$$\text{parent}(k) = \min \{ i \mid l_{ik} \neq 0, i > k \} = \min \{ j \mid u_{kj} \neq 0, j > k \}.$$

The *ancestors* of node k are all nodes on the single path from node k to the root, and the *descendants* of node k are all nodes in the subtree rooted at node k. Figure 1 shows

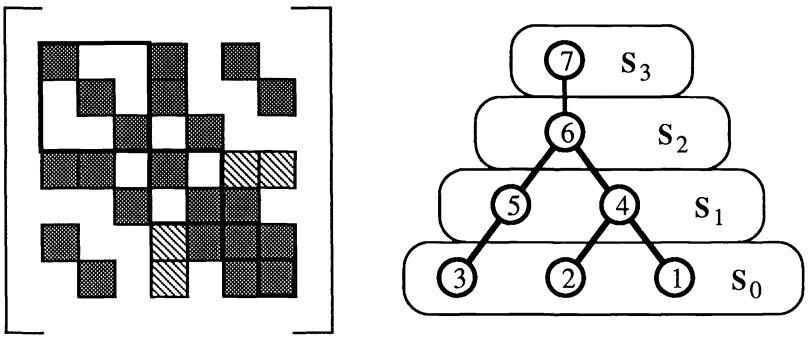


FIG. 1. *Compatible pivot sets and elimination tree.*

an example **LU** factorization and its associated elimination tree. Original nonzeros in **A** are shaded darker than the four fill-ins. Note that if **A** can be permuted into block diagonal form, the associated elimination tree will actually be a *forest*, with one tree for each irreducible component. This discussion assumes that **A** has a single tree, but it can be easily generalized to the reducible case.

In the context of a symmetric matrix, the idea of a compatible set can be related to the elimination tree. First, a leaf node is a node with no children. That is, node k is a leaf node if $l_{kj} = u_{jk} = 0$ for $1 \leqq j < k$. The set of leaf nodes of the elimination tree forms the first compatible set $\mathbf{S}_0$, with size m_0. The leaves are removed from the elimination tree, and the second set, $\mathbf{S}_1$, with size m_1, consists of the leaves of the reduced elimination tree. Equivalently, the compatible set $\mathbf{S}_q$ consists of all nodes with *height* q,

$$\text{height}(k) = \left\{ \begin{array}{l} 0, \quad \text{if } k \text{ is a leaf node,} \\ 1 + \max\limits_{j=1,n} \{\text{height}(j) \mid k = \text{parent}(j)\}, \text{ otherwise} \end{array} \right\},$$

$$\mathbf{S}_q = \{k \mid \text{height}(k) = q, \text{for } 1 \leqq k \leqq n\}.$$

The elimination tree height, h, is simply

$$h = \max_{k=1,n} \{\text{height}(k)\} + 1.$$

Note that the definition of the elimination tree height used in this paper is one more than the standard definition of the height of the root node of a tree. The matrix in Fig. 1 is permuted such that the compatible sets, $\mathbf{S}_0$, $\mathbf{S}_1$, $\mathbf{S}_2$, and $\mathbf{S}_3$, lie on the diagonal, in sequence. The compatible sets are highlighted in the matrix, and the nodes of the elimination tree in each compatible set are circled. The height, h, of the elimination tree in Fig. 1 is four.

2.1. Previous parallel algorithms based on symmetric orderings. Some of the previous algorithms for general unsymmetric sparse matrices use a symmetric ordering and work with a symmetric pattern of the matrix (either $\mathbf{A} + \mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$) during the analysis phase when determining the parallelism due to sparsity (i.e., the elimination tree). If the pattern of **A** is already symmetric (or nearly so) this approach has several advantages. It is easier to predict or estimate the pattern of **L** and **U**, and quick symmetric orderings such as minimum degree and nested dissection can be done using no more space than the original matrix. Finally, all of the powerful graph theory for the symmetric matrices (with undirected graphs) can be applied to the unsymmetric case [34], [35].

Duff and Reid's symmetric *multifrontal* method (MA27) [11], [13], [16] is an outer-product factorization, in which each node k in the elimination tree represents a rank-one update of $\mathbf{A}^{(k)}$ with the outer product of row and column k, and the computation of column k of $\mathbf{L}$. The updates are computed as small, dense frontal matrices to reduce gather-scatter indirection. In their unsymmetric version with Amestoy and Dayde (MA37, which in this paper refers to the experimental version [2], [3], [12], not to the one in the Harwell Library [17]), they use the symmetric nonzero pattern of $\mathbf{A} + \mathbf{A}^{\mathbf{T}}$ during factorization. The frontal matrices still have symmetric pattern as in the symmetric multifrontal method, but the nonzero values are unsymmetric. Nodes in the tree are merged to increase the granularity of the dense matrix operations within each node. *Node amalgamation*, as it is called, is done carefully, since it also increases fill-in. Numerical pivoting constraints can change the original permutation and can delay the work of a node. Alaghband and Jordan's algorithm [1] uses symmetric permutations to create a sequence of compatible pivot sets. Since the algorithm does not permute off the diagonal while building the compatible set, it essentially uses the pattern of $\mathbf{A} + \mathbf{A}^{\mathbf{T}}$ for the pivot search and for determining the parallelism due to sparsity.

In George and Ng's Sparspak-C outer-product factorization algorithm [20], [21], [22], symbolic factorization creates a static data structure that allows for any row interchanges during the numerical factorization phase, which can then employ conventional partial pivoting to maintain numerical accuracy. The patterns of $\mathbf{L}$ and $\mathbf{U}$ for any partial pivoting decisions are contained in the patterns of the Cholesky factors $\mathbf{L}_c$ and $\mathbf{L}_c^{\mathbf{T}}$ of the symmetric positive definite matrix $\mathbf{A}^{\mathbf{T}}\mathbf{A}$, assuming that $\mathbf{A}$ has a zero-free diagonal. Fill-in is controlled with a symmetric ordering of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ applied as a preordering to the columns of $\mathbf{A}$. The symmetric patterns of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ and $\mathbf{L}_c + \mathbf{L}_c^{\mathbf{T}}$ are used only in the ordering phase and are not used in the symbolic or numerical factorization phases. In their parallel version, the symmetric ordering is followed by Liu's heuristic *elimination tree rotations* to reduce the elimination tree height [32]. In [23], Gilbert develops a parallel implementation of Gilbert and Peierls's partial pivoting scheme [24]. As in Sparspak-C, the elimination tree is formed from the pattern of the Cholesky factor of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$. It is based on a column-oriented inner-product form of $\mathbf{LU}$ factorization, in which stage k computes column k of both $\mathbf{L}$ and $\mathbf{U}$. It uses a sparse triangular solver that takes advantage of sparsity in the right-hand side and works in time proportional to the number of floating-point operations (a unique feature of the algorithm). In contrast to Sparspak-C, the work at each node can be pipelined with the work of its children. Both of these algorithms (Sparspak-C and that of Gilbert and Peierls) are based on a partial pivoting method, and therefore they recommend a symmetric ordering of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ for the columns of $\mathbf{A}$. However, it might also be possible to use a purely unsymmetric ordering for the columns of $\mathbf{A}$. No preordering can be found for the rows because the algorithms use partial pivoting with row interchanges during the numerical factorization.

Alternative symmetric orderings include Leuze's method [28], which builds a sequence of compatible pivot sets for a symmetric positive definite matrix, or the method of Lewis, Peyton, and Pothen [29], which is an efficient implementation of Jess and Kees's method [25], and is based on a *clique tree* representation of the chordal graph.

Using a symmetric ordering to determine the parallelism due to sparsity in an unsymmetric matrix and to build the elimination tree has its drawbacks. The symmetrized nonzero pattern of $\mathbf{A}$ has more nonzeros than the true unsymmetric pattern. Including these elements (which are actually zero) during the ordering phase can reduce the amount of parallelism due to sparsity, because parallelism due to sparsity depends, inherently, on sparsity. The denser a matrix, the less parallelism due to sparsity. This is particularly true for matrices with highly asymmetric nonzero pattern. Consider the matrix $\mathbf{A}$ in Fig.

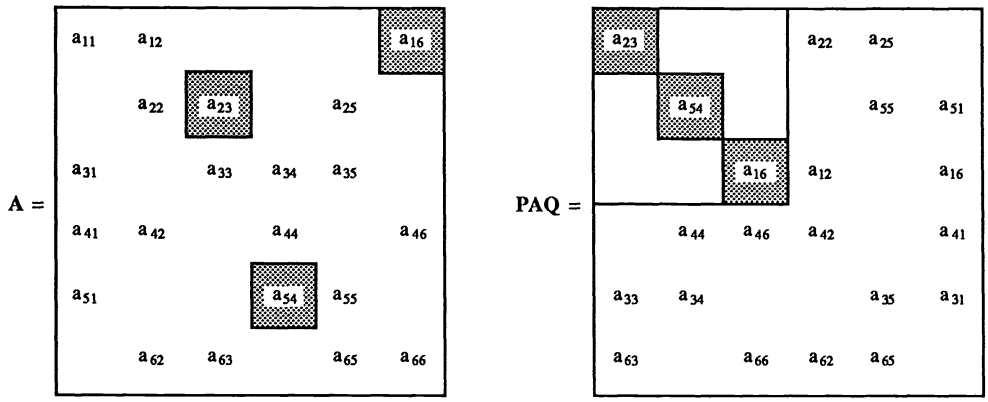


FIG. 2. *Counterexample sparse matrix.*

2, designed so that both $\mathbf{A} + \mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ are dense. Note that the matrix is irreducible (it cannot be permuted to upper block triangular form). A symmetric ordering based on $\mathbf{A} + \mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$ would find no parallelism due to sparsity. Even if the partial pivoting method would use a column preordering that is not based on $\mathbf{A} + \mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$, the best case would be a dense $\mathbf{U}$ and an $\mathbf{L}$ with three zeros in the last three rows of the first column. No parallelism due to sparsity can be found for the partial pivoting method for this matrix, because the preordering must allow for the worst-case pivoting of the rows during numerical factorization. The parallelism must be found dynamically during the numerical factorization, because no parallelism is available if the worst-case numerical pivoting is assumed. In the unsymmetric multifrontal method, numerical pivoting during factorization increases fill-in, delays the work of a node, and modifies the prediction of parallelism. In both cases the prediction of parallelism due to sparsity can be much less than what can be found with a purely unsymmetric ordering. For example, an unsymmetric ordering during numerical factorization can select a_{23}, a_{54}, a_{16} as the first three pivots (assuming they are numerically acceptable). The result after selecting these three pivots is the matrix $\mathbf{PAQ}$, also shown in Fig. 2. The reductions associated with the three pivots can be carried out in parallel. This is an extreme example, but it highlights the limitations of using a symmetric ordering for determining the parallelism due to sparsity in a matrix.

2.2. Previous parallel algorithms based on unsymmetric orderings. In 1973, Calahan developed a sequential unsymmetric pivot search algorithm for building a compatible pivot set [4]. The method searches the matrix by rows according to the original ordering. Smart and White [36] extend Calahan's original method by placing a Markowitz cost criterion on the pivots in $\mathbf{S}$, and they search the nonzeros in order of increasing cost. As in Calahan's method, their search algorithm is sequential, and no numerical accuracy criterion is placed on the pivots. Wing and Huang [38] present a fine-grain task graph for scheduling unsymmetric factorization. They assume that the matrix is already ordered, and they do not consider ordering for fill-in, accuracy, or finding the shortest task graph. Yang [39] proposes a modified Markowitz cost function for use in a conventional sparse factorization algorithm in order to produce a parallel ordering for the factorization of a second matrix with the same pattern as the first. Parallelism is not exploited in the first factorization. Finally, PSolve is a medium-grain parallel algorithm that computes a factorization $\mathbf{L}_p\mathbf{U}$ with pairwise reductions, in which a single row is used to reduce a single nonzero in another row, and where $\mathbf{L}_p$ is a product of pairwise operators [7].

Previous algorithms that use an unsymmetric ordering either do not consider the pivot search at all, or use a sequential pivot search to build a compatible pivot set for subsequent parallel reduction. The exception is PSolve, which can have numerical accuracy problems for some matrices, and also requires a good column preordering to reduce fill-in (such as that used by Sparspak-C). The next section presents a new parallel algorithm based on an unsymmetric ordering for the direct solution of general unsymmetric sparse systems of linear equations. The motivation behind the algorithm is to exploit more parallelism for matrices with highly asymmetric nonzero pattern than is possible with a symmetric ordering, and to exploit parallelism in all phases of the algorithm (including the pivot search itself). In addition, the algorithm should provide as accurate a solution as an accurate, sequential code, such as MA28 [15]. The algorithm should also generate **LU** factors with as little fill-in and memory usage as possible.

3. D2, a nondeterministic parallel algorithm for general unsymmetric sparse matrices. The algorithm D2 is based on a new nondeterministic parallel pivot search heuristic that finds a compatible pivot set **S** of size m, followed by a parallel rank-m update. These two steps alternate until switching to dense matrix code or until the matrix is factored. The algorithm is based on a shared-memory multiprocessing model and takes advantage of both concurrency and (gather-scatter) vectorization. Between each phase of the algorithm (initializations, pivot search, rank-m update, switch to dense matrix code, update of pivot search data structure, garbage collection, and forward and backward substitution) is a barrier synchronization point, and parallelism is exploited within the phases. The pivot search heuristic described below finds the first pivot set of size three for the counterexample matrix shown in Fig. 2. The data structures underlying the algorithm are first described, followed by a description of each phase of the algorithm. The pivot search phase is also demonstrated with an example.

3.1. Data structures. Three one-dimensional arrays hold the initial matrix as a set of unordered triples (numerical value of a_{ij}, row index i of a_{ij}, and column index j of a_{ij}) with one triple for each nonzero in the matrix. Additional space is provided at the end of the three arrays for space to hold $\mathbf{A}_k$, **L**, and **U** during factorization, since fill-in causes the number of nonzeros to increase. The initial matrix is sorted and stored into two data structures: (1) a row-oriented data structure containing the nonzero numerical values and pattern on a row-by-row basis, and (2) a column-oriented data structure representing only the nonzero pattern of the matrix on a column-by-column basis. The pattern of the matrix is stored in both formats, since the parallel pivot search phase accesses it by both rows and by columns. Other sparse matrix codes such as MA28 also maintain the pattern of matrix in both forms during the factorization.

In the row-oriented data structure, each row of $\mathbf{A}_k$ is represented as a doubly linked list of blocks, where each row block holds up to 32 column indices and nonzero values (equal to the Alliant FX/8 vector register length). A block link list structure was chosen to simplify memory allocation and garbage collection. As the algorithm progresses, fill-ins cause the rows to grow in length, increasing the amount of memory required to store them. A common method (used by MA28) is to store each row in a contiguous region of memory just large enough to hold it. If a row exceeds its allotment, the entire row is moved to a new region large enough to hold it. In a parallel algorithm, multiple processors are competing for access to the memory allocation mechanism, which can become a serial bottleneck. In addition, garbage collection is a more costly operation in a parallel algorithm than it is in a sequential algorithm. Rather than copying the entire row, the block link list structure allows a processor to allocate only a small additional amount of memory (one block, for example), and link it onto the end of the row. The original

space used by the row does not become garbage. It is still in use. Reducing the amount of garbage reduces the amount of garbage collection required by the algorithm. The column-oriented data structure is similar to the row-oriented structure, except that no numerical values are stored in the blocks; the data structure holds only the pattern of the matrix during factorization. The **L** and **U** factors are stored in the more conventional sparse matrix format similar to that used by MA28, with each row of **U** (or column of **L**) residing in a contiguous chunk of memory of length equal to the number of nonzeros in the row of **U** (or column of **L**).

Vector computers exploit a pipelined architecture to achieve high performance when computing a sequence of common operations on an entire stream of data (a vector operation). Many vector computers employ fixed-length vector registers which can be operated on by a single vector instruction, and vector operations of length greater than the vector register length are done with multiple vector instructions. The performance of a vector operation equal to the computer's vector register length is typically close to the peak performance of that operation, since it can be handled by a single vector instruction. This leads to a natural choice for the size of the block in the block link list data structure used to hold the rows and columns of $\mathbf{A}_k$. A block size equal to the computer's vector register length was selected as a reasonable trade-off between performance and the amount of internal fragmentation (memory wasted due to the mismatch between the row length and the fixed block length). The best block size for a scalar computer depends on the relative trade-offs between memory and run time. A large block size (32, for example) results in higher internal fragmentation and fewer links to follow during the scan of a row or column than a small block size (eight, for example). The other extreme is a block size of one, which is used in SPICE [33] and results in no fragmentation but many links to traverse during the scan of a row or column. A structure with a small block size also requires more memory to store the links.

Finally, in order to assist the pivot search phase, a data structure of size $O(n)$ is maintained that records the ordering of the rows, from shortest to longest. Other $O(n)$ data structures include work spaces for each processor and permutation vectors that hold the current row and column permutation matrices **P** and **Q**.

3.2. Pivot search. The nondeterministic parallel pivot search constructs a compatible pivot set **S** at stage k from the current $\mathbf{A}_{k-1}$ for a subsequent sparse rank-m update that computes $\mathbf{A}_{k+m-1}$. First, a single numerically acceptable pivot with lowest Markowitz cost (MinCost) in the four shortest rows is found and added as the first pivot in **S** (a pivot is numerically acceptable if it meets the row-oriented threshold pivoting criterion listed in (1)). All rows with nonzeros in the pivot column and all columns with nonzeros in the pivot row are then *marked*. No other pivots can come from marked rows or columns, since those rows and columns will be updated by the reduction associated with this pivot. This sequential step is based on the method used in Y12M [42]. Once MinCost is found, the parallel nondeterministic search can begin. Each partially independent task searches a single unmarked row for the nonzero with lowest cost in the row (among those nonzeros that are numerically acceptable). Nonzeros in marked columns are ignored. The task is skipped if the row itself is marked. The search algorithm never backtracks by removing a pivot from **S** once it has been added. Therefore, the tasks are executed roughly in order of increasing row length as an additional heuristic to keep fill-in low by searching first in rows likely to hold low-cost pivots.

If the nonzero with lowest Markowitz cost in the row (among the numerically acceptable nonzeros in unmarked columns) has cost $\leqq$ Factor · MinCost, then this is a *potential pivot*, where *Factor* is a user-selectable parameter (typically two to eight). At

this point, however, two (or more) tasks could have found potential pivots that are compatible with the current set but incompatible with each other. Only one can be added to **S**. A sequential search algorithm would add either the first pivot it found, or it would add the lower cost of the two. The second option would require that all the pivots be searched in order of increasing cost, an approach taken by Smart and White [36]. Since finding the single lowest cost pivot can sometimes lead to searching the entire matrix [41], this can be a costly approach. The first option is simpler, and it also implies that either pivot is acceptable, since the first one found is the one selected. The first option also leads naturally to a fast parallel search. Since either pivot is acceptable, the parallel algorithm should have the freedom to include in **S** whichever pivot it came across first. This is a "greedy" approach which emphasizes speed, as opposed to other possible approaches that could include elaborate schemes to find more optimal pivot sets. Such elaborate schemes would require additional interaction and synchronization among the processors taking part in the search. For example, two tasks that simultaneously found potential pivots that are incompatible with each other would have to coordinate between themselves in order to select the best pivot according to some higher-level heuristic, such as selecting from the two the pivot with lower cost.

On a MIMD computer with asynchronous processors, which processor found its pivot first depends on variables beyond the scope of the algorithm, such as memory-bank conflict, page faults, cache behavior, etc. Restricting the asynchronous behavior of a parallel computer requires additional synchronization overhead, while an algorithmic paradigm that allows for asynchronous behavior can lead to an algorithm with low synchronization overhead. Rather than strictly forcing the parallel algorithm to mimic the behavior of a sequential algorithm, the following search algorithm allows for asynchronous behavior and maps well to the underlying architecture.

Each task that finds a potential pivot attempts to lock the pivot set by entering a critical section guarded by a single lock implemented with an atomic test-and-set instruction. While attempting to enter the critical section, the task continually rechecks to see if the row and column of its potential pivot are still unmarked. If it becomes marked, the task abandons the attempt at entering the critical section. Once the task gains exclusive access to the critical section, it rechecks its potential pivot one last time. If the task finds the row and column of its potential pivot unmarked, it adds the pivot to **S**, marks all rows with nonzeros in the pivot column and all columns with nonzeros in the pivot row, allocates space for the pivot row and column in **U** and **L**, and updates the row and column permutations (**P** and **Q**). The pivot set **S** is then unlocked, and the successful task stores the pivot row in **U**, stores the pattern of the pivot column in **L**, and searches a new unmarked row. If instead the task finds the row or column of its potential pivot marked, it abandons its choice and does not add its pivot to **S**. The work inside the critical section is O(pivot row length + pivot column length) if the task is successful, or $O(1)$ if it is not. If the task was unsuccessful and the row it is searching is marked, it abandons the row and searches another. Otherwise, if only the column of its previous potential pivot was marked, another potential pivot might exist in the same (unmarked) row in a different (unmarked) column. The task rescans the row and tries again. When the row is exhausted it is abandoned and a new unmarked row is searched (the exhausted row is either marked by another task, or there are no numerically acceptable nonzeros in the row that are in unmarked columns and that have acceptable cost).

The pivot search algorithm is demonstrated in Figs. 3 through 6 with an example highlighting some of the possible interactions between two processors. Two processors build a pivot set **S** in a 10-by-10 sparse matrix, and a time-line dialog is shown in Fig. 3. An attempt to access the critical section is shown as a shaded box, and this is sometimes

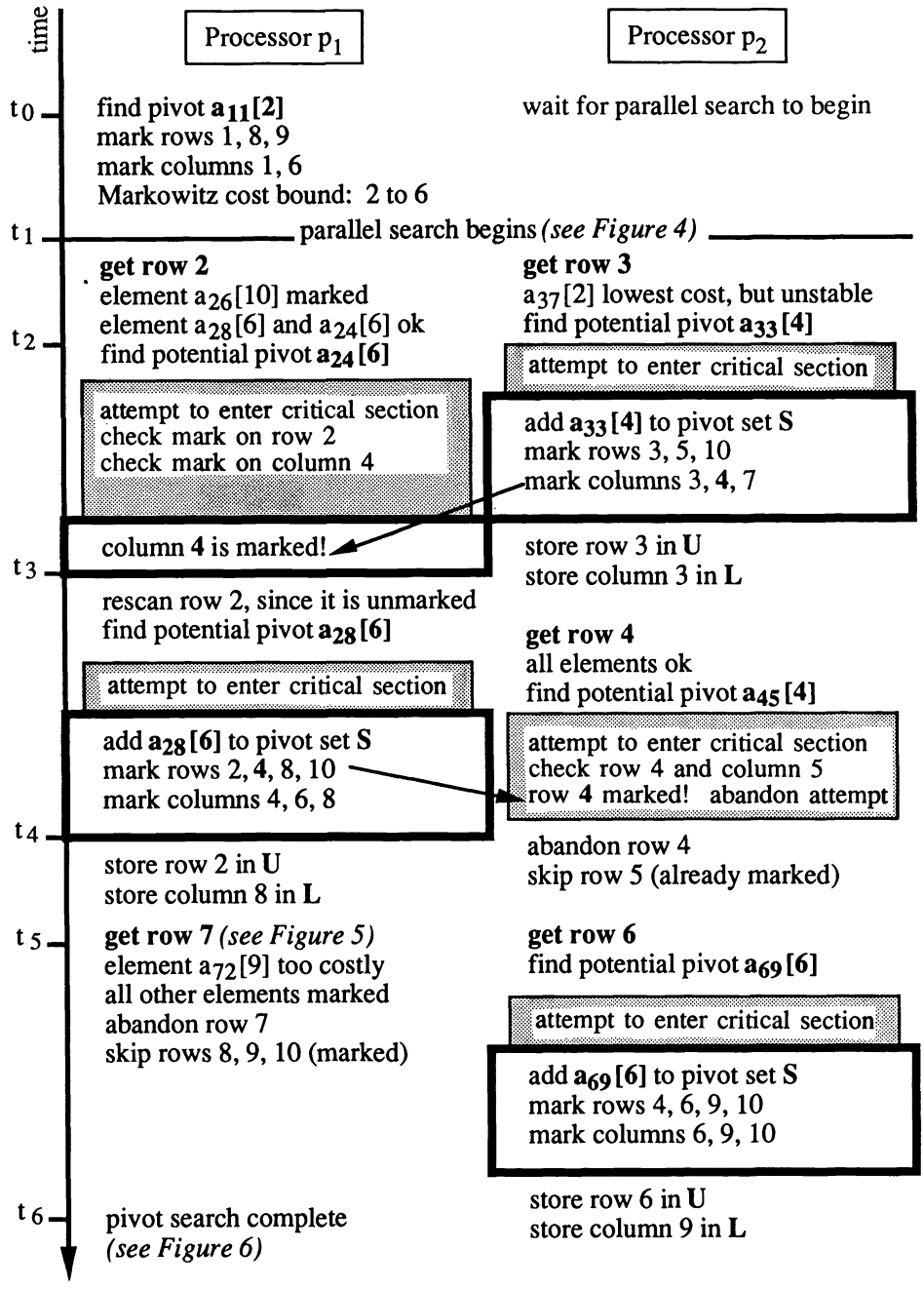


FIG. 3. *Pivot search example: time-line dialog.*

followed by a thick-edged box drawn around the access to the critical section itself. An arrow represents a conflict (when one processor marks the potential pivot of the other). There are many conflicts and there is little parallelism evident in this small example; a larger matrix typically exhibits more parallelism. The rows of the matrices shown in Figs. 4 and 5 are ordered by increasing the number of nonzeros from top to bottom.

At time t_0, processor p_1 finds the first pivot a_{11} with a Markowitz cost of two (the Markowitz cost of each nonzero discussed in Fig. 3 is shown in brackets). Because Factor

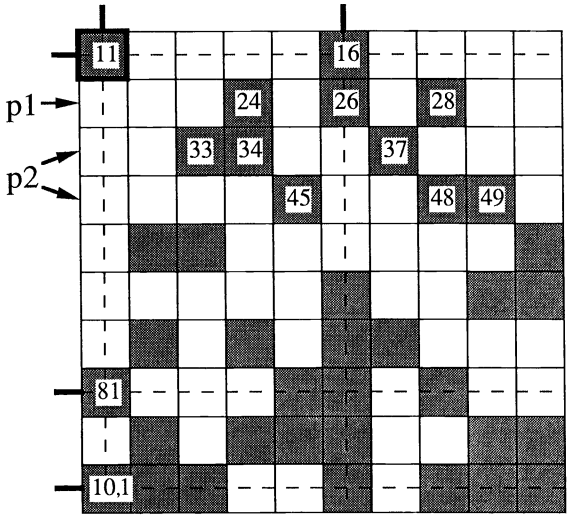


FIG. 4. *Pivot search example: finding the first three pivots.*

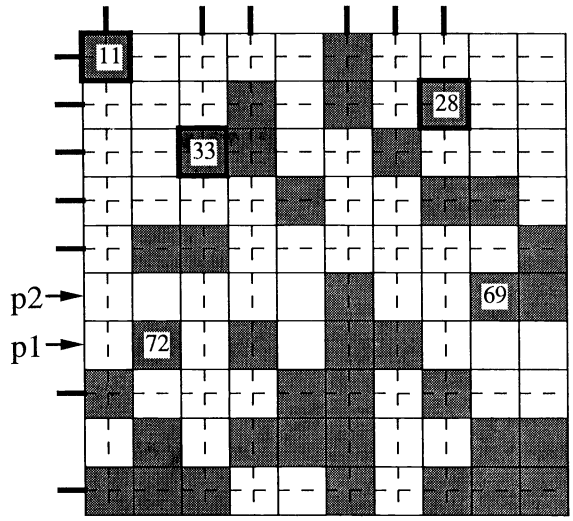


FIG. 5. *Pivot search example: finding the last pivot.*

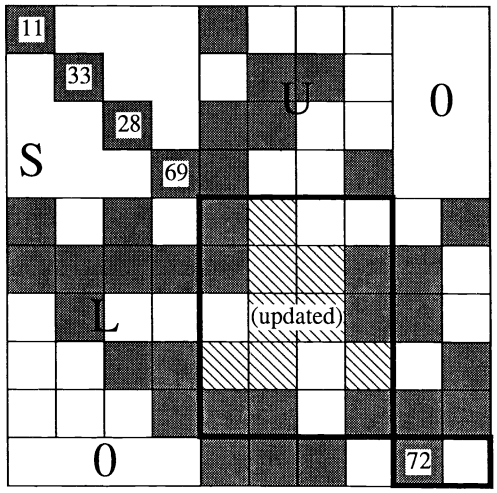


FIG. 6. *Pivot search example: after the rank-four update.*

is three for this example, nonzeros with cost greater than six are not allowed in the set. The appropriate rows and columns are marked, and the parallel search begins at time t_1 (see the matrix in Fig. 4). Processor p_1 finds a potential pivot a_{24} with cost six (time t_2), while processor p_2 finds a potential pivot a_{33} with cost four (numerical values are not shown, but in this example processor p_2 cannot select a_{37}, since it does not meet the threshold pivoting criterion). Both potential pivots are compatible with the current set but incompatible with each other because a_{34} is nonzero. Processor p_2 happens to get to the critical section first and marks column four, thereby marking the potential pivot of the other processor (time t_2 to t_3). Processor p_1 detects this within its critical section and must abandon its choice, but it can look for another potential pivot in the same row, since the row itself is not marked. It finds potential pivot a_{28} and adds it to the set (time t_3 to t_4). Meanwhile, processor p_2 has located another potential pivot, a_{45}. Unfortunately for processor p_2, the previous pivot a_{28} has marked its choice; it detects this while it is attempting to enter the critical section and abandons a_{45} *before* it enters the critical section.

At time t_5, each processor picks up a new row, and the state of the matrix is shown in Fig. 5. Row five is skipped because it has been marked. Processor p_1 finds only one unmarked nonzero, a_{72}, but its Markowitz cost is too high. Processor p_2 finds potential pivot a_{69} and adds it to the set. The search is complete because rows eight, nine, and ten have been marked. The final pivot set is permuted to the diagonal and the matrix is now ready for a sparse rank-four update, as shown in Fig. 6. The rows and columns updated are ordered according to when they were first marked. A box is drawn around the region updated by the pivot set, and fill-ins are striped. Any nonzeros in the box in the lower right-hand corner are compatible with the current set, but they were disallowed due to one or both of the other two criteria (threshold pivoting and Markowitz cost).

3.3. Sparse rank-*m* update. The rank-m update is divided into two major phases, each consisting of a set of completely independent tasks. The first phase is a symbolic update of the column-oriented data structure holding only the pattern of $\mathbf{A}_{k+m-1}$ on a column-by-column basis. The second phase is a numerical and symbolic update of the row-oriented data structure holding both the pattern and the nonzero values of $\mathbf{A}_{k+m-1}$ on a row-by-row basis. In both phases, the algorithm takes advantage of medium-grain parallelism, rather than the large-grained parallelism of m rank-one updates. Instead, the update is divided into tasks for each row and column of $\mathbf{A}_{k+m-1}$ that is updated by $\mathbf{S}$. This allows the update to proceed without synchronization, since multiple updates to the same element are handled in a single task. The rank-m update is divided into two separate parallel phases so that nonzeros whose absolute values fall below a drop tolerance [42] can be removed in parallel. However, the use of a drop tolerance does not improve the speed of the code for most problems, since it complicates the parallelism in the rank-m update. The experimental results summarized in § 4 were taken without using a drop tolerance. At this point, all rows and columns of $\mathbf{A}_{k+m-1}$ are unmarked, stage $k+m-1$ is finished, and stage $k+m$ is ready to begin ($k \leftarrow k+m$).

3.4. Switch to dense matrix code. The switch to dense matrix code is controlled by two user-selectable parameters, SwPiv and SwDen (4 and 20 percent for this paper). The switch takes place if the size of the latest pivot set is less than or equal to SwPiv, if the density of $\mathbf{A}_k$ (number of nonzeros in $\mathbf{A}_k$ over $(n-k)^2$) is greater than SwDen, and if there is enough room to convert $\mathbf{A}_k$ into an $(n-k)$-by-$(n-k)$ dense array. The pivot set size is a rough measure of the amount of parallelism due to sparsity the algorithm is finding during the factorization. As the parallelism drops, the matrix is becoming dense, and it becomes more economical to factor the rest of the matrix as a dense matrix. If

the factorization is not yet complete, the algorithm then updates the pivot search data structure for the next parallel pivot search. The pivot search, rank-m update, and the test for switch to dense matrix code repeat until the matrix is factored. Garbage collection is performed as needed at the barrier synchronization points between the phases of the algorithm.

4. Experimental comparisons with previous methods. A series of experiments are summarized below to determine if the design criteria for the new algorithm, D2, are met. The algorithm should exploit more parallelism than methods using symmetric orderings for matrices with highly asymmetric nonzero patterns; it should exploit similar parallelism for matrices with symmetric patterns. The algorithm should compute an accurate solution and should generate **LU** factors with as little fill-in as possible.

4.1. Experimental design. Forty unsymmetric matrices with order between 500 and 5005 were selected from the Harwell/Boeing sparse matrix test collection [14] with varying degrees of asymmetry. The *asymmetry* of a matrix is the number of unmatched off-diagonal nonzeros over the total number of off-diagonal nonzeros. An unmatched nonzero is one for which $a_{ij}^{(0)}$ is nonzero but for which $a_{ji}^{(0)}$ is zero (for $i \neq j$). The matrices divide into three groups according to their asymmetry: 12 matrices in group one with symmetric or nearly symmetric patterns (asymmetry < 0.1), 17 matrices in group two with $0.1 \leqq$ asymmetry $\leqq 0.5$, and 11 matrices in group three with highly asymmetric patterns (asymmetry > 0.5). Most of the results are from an Alliant FX/8 (running Xylem) with eight processors, 48 megabytes of main memory, 128 kilobytes of cache memory, and a peak performance of over 90 megaflops. The Xylem operating system sets aside 16 megabytes of main memory to simulate Cedar global memory [27], so each program uses only 32 megabytes. Each code was compiled with the Alliant FORTRAN compiler, FX/FORTRAN (version 3.1.33). A second set of experiments are from an Alliant VFX/80 (running the standard operating system) with eight processors, 192 megabytes of main memory, 512 kilobytes of cache memory, and a peak performance of 188 megaflops.

The performance of the D2 algorithm is compared with that of MA28, Y12M, Sparspak-C, and MA37. The reordering to lower block triangular form is disabled in MA28 to make a consistent comparison with the other algorithms, although this can improve the results for reducible matrices. In addition, MA28 uses its original pivot search method, rather than the faster search originally used in Y12M and later incorporated into MA28. The Y12M code has been partially optimized by Zlatev for vectorization and concurrency on the Alliant FX/8. Ng provided both a parallel and a sequential version of Sparspak-C, but because the parallel version runs only on a Sequent multiprocessor, only the sequential version is tested on the Alliant. Rows with 50 or more nonzeros are removed from **A** before forming $\mathbf{A}^T\mathbf{A}$ in the ordering phase of Sparspak-C. The experimental version of MA37 uses the Level 3 BLAS [10] for the computations within the dense frontal matrices. The threshold for pivoting decisions (u) in MA28, Y12M, and D2 is 0.1, while the parameter is 0.001 for MA37 because it scales the matrix before factorization to improve numerical accuracy. First, MC19 [5] is applied to the matrix, which attempts to reduce the distance between the smallest and largest nonzero absolute values. Then the rows and columns are divided by the maximum absolute values in each row and column, respectively. The Factor parameter used for D2 is four; potential pivots with Markowitz cost greater than four times the lowest cost pivot in the compatible pivot set are excluded. D2 uses the same scaling when it is compared with MA37.

4.2. Experimental results. The Y12M algorithm typically finds the most accurate solution as compared with MA28, Sparspak-C, and D2 (without scaling), but all of them

find an acceptable solution (if more than one potential pivot have the same Markowitz cost and meet the threshold pivoting requirement, Y12M chooses the nonzero with the largest absolute value, which leads to better numerical accuracy). However, MA37 (with scaling) does much better for a few poorly scaled matrices in the test set. The D2 algorithm gets nearly the same accuracy when it solves the same scaled system as MA37. Codes such as MA28 and Y12M need not relax the Markowitz constraint as D2 does to find a parallel set, so they tend to find an **LU** factorization with fewer nonzeros. MA28 typically returns 90 percent of the number of nonzeros in $\mathbf{L} + \mathbf{U}$ as D2 (when D2 does *not* switch to dense matrix code); the ratio drops to 60 percent when D2 does switch. For matrices in group one or two, Sparspak-C and MA37 are comparable to MA28, and typically find an **LU** factorization with 40 to 70 percent of the number of nonzeros as D2 (with switch to dense). However, the comparison reverses for the highly asymmetric matrices in group three, as shown in Table 1, when D2 computes a factorization with much less fill-in than Sparspak-C or MA37 (including the nonzeros introduced in the switch to dense matrix code). The decrease in fill-in is not reflected in a decrease in memory usage. Sparspak-C typically usually uses half the memory required by D2, which points to internal fragmentation in D2's block link list data structure.

One way of measuring the potential parallelism that the various parallel algorithms might be able to exploit is to compare the height of the elimination trees built by each algorithm. The comparison has the advantage of being constant across different architectures, since the elimination tree height represents a lower bound on the completion time on a hypothetical parallel machine with an unlimited number of processors, assuming that each node of the tree requires unit time to execute. Elimination tree height is an incomplete measure of potential parallelism, because the unit-time assumption is false, and because the work for a single node can be done in parallel. But in general, a shorter elimination tree should allow for more parallelism. Reducing the elimination tree height is the goal of Liu's elimination tree rotations, for example. Also, D2 does not actually create an elimination tree, since doing so would require the symmetric-order assumption made in the analysis phases of the other parallel algorithms under consideration (MA37 and Sparspak-C). However, D2 produces a sequence of compatible pivot sets, the number of which is roughly comparable to the height of an elimination tree and is referred to as the elimination tree height for the D2 algorithm. This correlation is made only if D2 does not switch to a dense matrix code.

Figures 7, 8, and 9 display the ratio of the tree height for Sparspak-C (with both minimum degree and nested dissection) and MA37 (with minimum degree and no amalgamation) over the tree height found by D2. The relative tree height is plotted versus matrix asymmetry; each point represents a single matrix. Although the matrices come from a wide range of problems in science and engineering, a loose correlation can be seen in each graph between the relative tree height and the asymmetry. The median tree height ratio is listed for each group of test matrices. As measured by elimination tree height, D2 displays a comparable amount of potential parallelism for matrices whose nonzero pattern is symmetric or nearly so, and much more parallelism for matrices with highly asymmetric nonzero pattern.

TABLE 1
Relative number of nonzeros in $\mathbf{L} + \mathbf{U}$ *for highly asymmetric matrices.*

Number of nonzeros as compared with D2	Minimum	Median	Maximum
Sparspak-C (with minimum degree)	0.64	1.10	1.66
MA37 (with some node amalgamation)	1.00	2.65	3.15

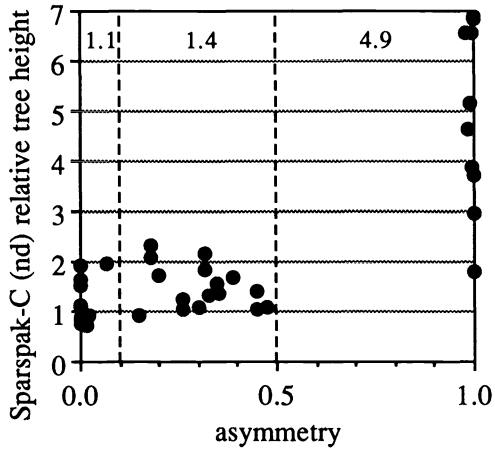


FIG. 7. Sparspak-C (*nested dissection*) *relative tree height versus asymmetry*.

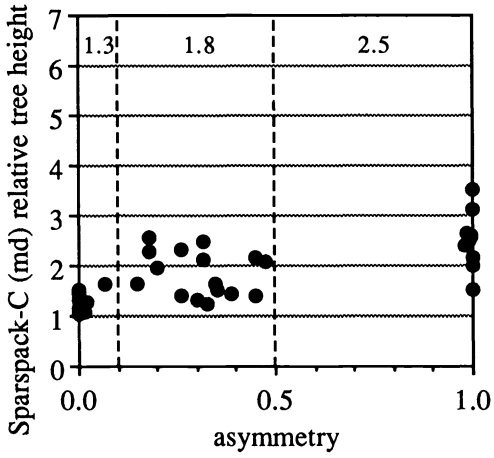


FIG. 8. Sparspak-C (*minimum degree*) *relative tree height versus asymmetry*.

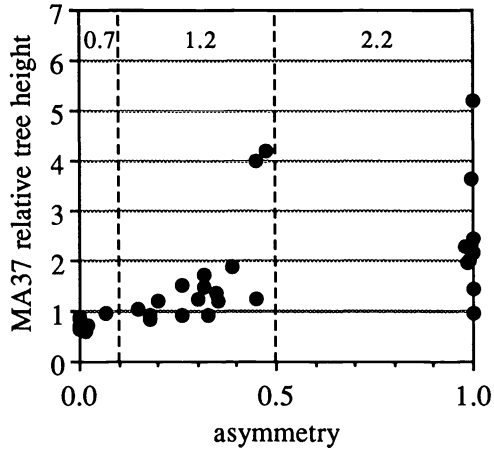


FIG. 9. MA37 *relative tree height versus asymmetry*.

Table 2 presents a summary of the relative run time on the Alliant FX/8. It lists the minimum, median, and maximum of the ratio of the run time of MA28, Y12M, and Sparspak-C over the run time of D2 (the run time includes everything from storing the matrix into the internal data structure to computing the final solution). The Sparspak-C algorithm was written on a multiprocessor without vector hardware, so D2 and Sparspak-C are compared on a single processor using only the scalar floating-point arithmetic (the PARALIN dense matrix code [26] normally used in D2 was replaced with LINPACK [9] compiled with the vector optimizer turned off). The sequential version of the D2 algorithm is nearly identical to the parallel version, except that all calls to the low-level synchronization routines are removed. The sequential version still creates a compatible pivot set S of size m and then performs a rank-m update. MA28 was actually compiled with both the vector and concurrency optimizers turned on, since the code does gain a slight benefit from running on eight processors with automatic detection of concurrent loops. Codes with dynamic memory allocation are given enough memory so that garbage collection does not occur (including D2). The first section compares the performance of each code on a single processor. The second section lists the parallel performance of Y12M with respect to D2, the overall speedup of D2, and the speedup of D2's pivot search. D2 gets a speedup in the pivot search phase that is nearly equal to the overall speedup of the algorithm. Y12M gets a speedup of about two in its search for a single pivot and in its rank-one update (the *speedup* is the run time of the sequential version of an algorithm over the run time of the parallel version of the same algorithm).

The D2 algorithm is not only a good parallel code (with both good potential parallelism (elimination tree height) and fast run time on an Alliant FX/8), it is also a competitive sequential algorithm. Sparspak-C is a faster algorithm for matrices with symmetric patterns, but D2 is much faster for highly asymmetric matrices. There are two explanations for the performance of the sequential version of D2. First, it is the only algorithm under comparison which incorporates a switch to dense matrix code. Second, the structure of the computations in the rank-m update can allow it to outperform a sequence of m rank-one updates by reducing gather-scatter indirection.

Finally, Fig. 10 summarizes the experimental results on eight processors of the Alliant VFX/80. The unsymmetric multifrontal algorithm, MA37, divides into a symbolic analysis phase followed by a numerical factorization phase. Normally the time for numerical factorization is dominant, but the numerical factorization phase has been highly

TABLE 2
Relative run time on the Alliant FX/8.

	Minimum	Median	Maximum
Comparisons on a single processor:			
MA28 (original version) vs. D2 (vec.)	1.2	4.3	13.7
Y12M (optimized on FX/8) vs. D2 (vec.)	0.52	1.9	8.9
Sparspak-C (min. degree, no vec.) vs. D2 (no vec.)			
asymmetry < 0.1	0.32	0.72	3.1
$0.1 \leqq$ asymmetry $\leqq 0.5$	0.25	1.2	3.1
asymmetry > 0.5	1.3	3.9	21.0
Comparisons on eight processors:			
Y12M (vec., 8 proc.) vs. D2 (vec., 8 proc.)	0.93	3.6	16.0
D2 (vec., 1 proc.) vs. D2 (vec., 8 proc.)			
overall	1.3	3.9	7.2
pivot search only	1.8	3.5	6.3

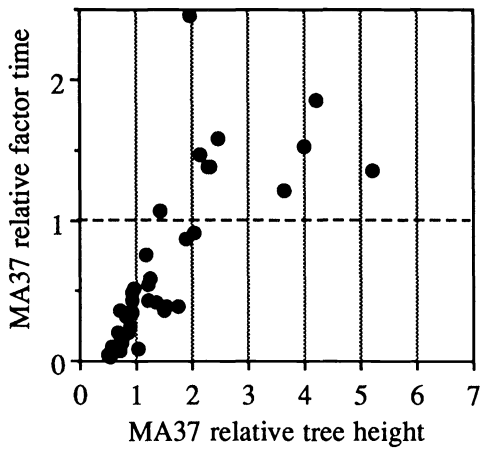


FIG. 10. MA37 *relative factor time versus* MA37 *relative tree height*.

optimized on a range of parallel vector computers, while the analysis phase has not. As a result, the run time of the analysis phase of MA37 is about three times that of the numerical factorization. If the pivot search and the update of the pivot search data structures are considered the "analysis" phase of the D2 algorithm (and the rank-m update the "numerical factorization"), then the analysis phase of D2 takes about one third the time of the numerical factorization. The key contribution of D2 is a fast parallel pivot search, which makes it difficult to compare it with another code (MA37) whose key contribution is a fast numerical factorization. However, the ratio of the MA37 numerical factorization time (with node amalgamation) over the D2 factor time (with switch to dense) is plotted in Fig. 10 as a function of the relative tree height (the tree height found by MA37 over the tree height found by D2). Elimination tree height gives only an incomplete measure of the potential parallelism due to sparsity, but the graph shows an intriguing correlation. D2 is faster when it finds a tree height less than half that of MA37, and usually slower otherwise.

5. Summary. The solution of a sparse system of linear equations, $\mathbf{Ax} = \mathbf{b}$, was introduced in § 1, along with the symmetric and unsymmetric ordering schemes that underlie the detection of parallelism due to sparsity as discussed in § 2. A six-by-six counterexample matrix highlighted the difficulty that symmetric ordering methods have for matrices with highly asymmetric nonzero patterns. Previous methods based on unsymmetric orderings do not take significant advantage of parallelism in the pivot search phase. The exception is PSolve, which can have numerical accuracy problems for some matrices and can also experience high fill-in. This was the motivation for the new algorithm, D2, presented in § 3. The algorithm is based on a new nondeterministic parallel pivot search that constructs a large compatible pivot set to allow for parallelism in the reduction phase. The asynchronous behavior of the pivot search maps well to the underlying shared-memory architecture with multiple, asynchronous processors. The experimental results presented in § 4 confirm the initial hypothesis that the D2 algorithm should outperform methods using a symmetric ordering when solving matrices with highly asymmetric nonzero patterns. The algorithm has comparable fill-in and numerical accuracy with MA28, although its memory usage can be high because of internal fragmentation in its block link list data structure. It finds a much shorter tree for highly asymmetric matrices, as compared with Sparspak-C and MA37, and is often faster for

these matrices. Further implementation details and experimental results on an Alliant FX/8, a Cray-2, and a Cray-XMP/48 are presented in [6].

6. Future work. There is much room for improvement in the new approach discussed in this paper. Currently, the parallel search phase finds the pivot set and completes before the rank-m update begins. It would also be possible to pipeline the pivot search and update phase. The algorithm would create a dynamic sequence of parallel pivots which would be compatible only with those pivots which are also currently being processed. Also, the first pivot in the pivot set $\mathbf{S}$ is found with a sequential search, but it is also possible to use parallelism in this search, as was done by Zlatev and his parallel version of Y12M. Rather than creating a diagonal pivot set, another easily invertible pivot matrix could also be used (such as a triangular matrix).

Can some of the ideas of the clique tree and clique graph methods be used in the unsymmetric case, without resorting to forcing a symmetric pattern on the matrix during the analysis phase? When the reductions associated with a pivot are performed on a symmetric matrix, the result is a clique between all former neighbors of that node in the undirected elimination graph. For an unsymmetric matrix, the result is a small dense submatrix in the reduced $\mathbf{A}_k$. As the factorization progresses, such dense submatrices are formed for each previous pivot, and the dense submatrices for some pivots would subsume those of other pivots; the corresponding effect in symmetric matrices is the maximal clique. It might be possible to take advantage of this structure in order to reduce the parallel pivot search, and to take better advantage of the architecture of typical vector computers with the use of dense matrix kernels.

The nondeterministic parallel pivot search requires a single critical section to add the individual pivots in the pivot set $\mathbf{S}$, which will be a problem with higher numbers of processors. The single critical section could be replaced with a software combining tree, in which each processor is associated with a leaf of the tree. A potential pivot would be entered on a leaf of the tree, and would combine with other potential pivots at internal nodes to form compatible pivot subsets. The root node would represent the final compatible pivot set $\mathbf{S}$. The combining tree scheme could result in more effective parallelism in the pivot search and maps to a message-passing or distributed-memory multiprocessor. For more details, see [6].

Acknowledgments. We would like to thank Esmond Ng for providing a copy of Sparspak-C. Zahari Zlatev provided an enhanced version of Y12M on the Alliant FX/8, and many helpful discussions while he was visiting the Center for Supercomputing Research and Development (CSRD) in 1988–1989. Patrick Amestoy worked with the first author to compare D2 with MA37 at the European Center for Research and Advanced Training in Scientific Computation; an earlier version of MA37 was also provided by Iain Duff while he visited CSRD in 1987. We would like to thank these researchers, two anonymous referees, and others at CSRD and Oak Ridge National Laboratory, whose comments and critique improved the presentation of this paper.

REFERENCES

[1] G. ALAGHBAND AND H. F. JORDAN, *Sparse Gaussian elimination with controlled fill-in on a shared memory multiprocessor*, IEEE Trans. Comput., 38 (1989), pp. 1539–1557.

[2] P. R. AMESTOY, M. J. DAYDE, AND I. S. DUFF, *Use of level* 3 BLAS *kernels in the solution of full and sparse linear equations*, in High Performance Computing, J.-L. Delhaye and E. Gelenbe, eds., North-Holland, Amsterdam, 1989, pp. 19–31.

[3] P. R. AMESTOY AND I. S. DUFF, *Vectorization of a multiprocessor multifrontal code*, Internat. J. Supercomputer Appl., 3 (1989), pp. 41–59.

[4] D. A. CALAHAN, *Parallel solution of sparse simultaneous linear equations*, in Proc. of the 11th Annual Allerton Conference on Circuits and System Theory, 1973, pp. 729–735.

[5] A. R. CURTIS AND J. K. REID, *On the automatic scaling of matrices for Gaussian elimination*, J. Inst. Math. Appl., 10 (1972), pp. 118–124.

[6] T. A. DAVIS, *A parallel algorithm for sparse unsymmetric LU factorization*, Center for Supercomputing Research and Development, University of Illinois, Urbana, IL, Report 907, 1989.

[7] T. A. DAVIS AND E. S. DAVIDSON, *Pairwise reduction for the direct, parallel solution of sparse unsymmetric sets of linear equations*, IEEE Trans. Comput., 37 (1988), pp. 1648–1654.

[8] T. A. DAVIS AND P. C. YEW, *A stable parallel algorithm for general unsymmetric sparse LU factorization*, in Abstracts of the SIAM Symposium on Sparse Matrices, Gleneden Beach, OR, SIAM Activity Group on Linear Algebra, 1989.

[9] J. J. DONGARRA, J. R. BUNCH, C. B. MOLER, AND G. W. STEWART, LINPACK *Users Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1978.

[10] J. J. DONGARRA, J. DU CROZ, I. S. DUFF, AND S. HAMMARLING, *A set of level 3 basic linear algebra subprograms*, Argonne National Laboratory, Argonne, IL, Report TM-88, 1988; ACM Trans. Math. Software, to appear.

[11] I. S. DUFF, *Parallel implementation of multifrontal schemes*, Parallel Comput., 3 (1986), pp. 193–204.

[12] ———, *Multiprocessing a sparse matrix code on the Alliant FX/8*, J. Comput. Appl. Math., 27 (1989), pp. 229–239.

[13] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 1986.

[14] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.

[15] I. S. DUFF AND J. K. REID, *Some design features of a sparse matrix code*, ACM Trans. Math. Software, 5 (1979), pp. 18–35.

[16] ———, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[17] ———, *The multifrontal solution of unsymmetric sets of linear equations*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 633–641.

[18] A. GEORGE AND J. W. H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[19] ———, *A minimal storage implementation of the minimum degree algorithm*, SIAM J. Numer. Anal., 17 (1980), pp. 282–299.

[20] A. GEORGE, J. W. H. LIU, AND E. NG, *A data structure for sparse* QR *and* LU *factorizations*, SIAM J. Sci. Statist. Comp., 9 (1988), pp. 100–121.

[21] 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.

[22] ———, *Parallel sparse Gaussian elimination with partial pivoting*, Oak Ridge National Laboratory, Oak Ridge, TN, 1988.

[23] J. R. GILBERT, *An efficient parallel sparse partial pivoting algorithm*, Report 88/45052-1, Center for Computer Science, Chr. Michelsen Institute, Bergen, Norway, 1988.

[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] J. A. G. JESS AND H. G. M. KEES, *A data structure for parallel* L/U *decomposition*, IEEE Trans. Comput., C-31 (1982), pp. 231–239.

[26] KUCK AND ASSOCIATES INC., *Paralin User's Guide*, Kuck and Associates, Incorporated, Urbana, IL, 1988.

[27] D. J. KUCK, E. S. DAVIDSON, D. H. LAWRIE, AND A. H. SAMEH, *Parallel supercomputing today and the Cedar approach*, Science, 231 (1986), pp. 967–974.

[28] M. LEUZE, *Independent set orderings for parallel matrix factorization by Gaussian elimination*, Parallel Comput., 10 (1989), pp. 177–191.

[29] J. G. LEWIS, B. W. PEYTON, AND A. POTHEN, *A fast algorithm for reordering sparse matrices for parallel factorization*, Report ORNL/TM-11040, Oak Ridge National Laboratory, Oak Ridge, TN, 1989.

[30] J. W. H. LIU, *Modification of the minimum-degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.

[31] ———, *Computational models and task scheduling for parallel sparse Cholesky factorization*, Parallel Comput., 3 (1986), pp. 327–342.

[32] ———, *Equivalent sparse matrix reordering by elimination tree rotations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 424–444.

[33] L. W. NAGEL, SPICE2: *A computer program to simulate semiconductor circuits*, Electronics Research Laboratory, University of California, Berkeley, CA, 1975.

[34] 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, New York, 1973, pp. 183–217.

[35] D. J. ROSE AND R. E. TARJAN, *Algorithmic aspects of vertex elimination on directed graphs*, SIAM J. Appl. Math., 34 (1978), pp. 176–197.

[36] D. SMART AND J. WHITE, *Reducing the parallel solution time of sparse circuit matrices using reordered Gaussian elimination and relaxation*, in Proc. IEEE International Symposium Circuits and Systems, 1988.

[37] W. F. TINNEY AND J. W. WALKER, *Direct solutions of sparse network equations by optimally ordered triangular factorization*, Proc. IEEE, 55 (1967), pp. 1801–1809.

[38] O. WING AND J. W. HUANG, *A computation model of parallel solution of linear equations*, IEEE Trans. Comput., C-29 (1980), pp. 632–638.

[39] G. C. YANG, DSPack: *A direct sparse matrix software package for shared-memory parallel machines*, in Abstracts of the SIAM Symposium on Sparse Matrices, Gleneden Beach, OR, SIAM Activity Group on Linear Algebra, 1989.

[40] M. YANNAKAKIS, *Computing the minimum fill-in is* NP*-complete*, SIAM J. Algebraic Discrete Methods, 1 (1981), pp. 77–79.

[41] Z. ZLATEV, *On some pivotal strategies in Gaussian elimination by sparse technique*, SIAM J. Numer. Anal., 17 (1980), pp. 18–30.

[42] Z. ZLATEV, J. WASNIEWSKI, AND K. SCHAUMBURG, Y12M: *Solution of large and sparse systems of linear algebraic equations, lecture notes in computer science* 121, Springer-Verlag, Berlin, New York, 1981.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 403-429, July 1990

006

MULTILEVEL FILTERING ELLIPTIC PRECONDITIONERS*

C.-C. JAY KUO†, TONY F. CHAN‡¶, AND CHARLES TONG§¶

Abstract. A class of preconditioners for elliptic problems built on ideas borrowed from the digital filtering theory and implemented on a multilevel grid structure is presented. These preconditioners are designed to be both rapidly convergent and highly parallelizable. The digital filtering viewpoint allows for the use of filter design techniques for constructing elliptic preconditioners and also provides an alternative framework for understanding several other recently proposed multilevel preconditioners. Numerical results are presented to assess the convergence behavior of the new methods and to compare them with other preconditioners of multilevel type, including the usual multigrid method as preconditioner, the hierarchical basis method, and a recent method proposed by Bramble–Pasciak–Xu.

Key words. filtering, multigrid, multilevel, parallel computation, preconditioned conjugate gradient, preconditioners

AMS(MOS) subject classifications. 65N20, 65F10

1. Introduction.

Preconditioned conjugate gradient (PCG) methods have been a very popular and successful class of methods for solving large systems of equations arising from discretizations of elliptic partial differential equations. With the advent of parallel computers in recent years, there has been increased research into effectively implementing these methods on various parallel architectures. In this paper, we present a class of preconditioners for elliptic problems built on ideas from the digital filtering theory and implemented on a multilevel grid structure. Our goal is to work towards preconditioners that are both highly parellelizable and rapidly convergent.

The idea of preconditioning is a simple one, but it is now recognized as critical to the effectiveness of PCG methods. Suppose we would like to solve the symmetric positive definite linear system $Ax = b$, where A arises from discretizing a second-order self-adjoint elliptic partial differential operator. A good preconditioner for A is a matrix M that approximates A well (in the sense that the spectrum for the preconditioned matrix $M^{-1}A$ is clustered around 1 and has a small condition number), and for which the matrix vector product $M^{-1}v$ can be computed efficiently for a given vector v. With such a preconditioner, one then solves in principle the preconditioned system $\tilde{A}\tilde{x} = \tilde{b}$, where $\tilde{A} = M^{-1/2}AM^{-1/2}$, $\tilde{x} = M^{1/2}x$ and $\tilde{b} = M^{-1/2}b$, by the conjugate gradient method.

Since an effective preconditioner plays a critical role in PCG methods, many classical preconditioners have been proposed and studied, especially for second-order elliptic problems. Among these are the Jacobi preconditioner (diagonal scaling), the symmetric successive overrelaxation (SSOR) preconditioner [3], and the incomplete factorization

* Received by the editors September 7, 1989; accepted for publication (in revised form) December 27, 1989.

† Signal and Image Processing Institute and Department of Electrical Engineering Systems, University of Southern California, Los Angeles, California 90089-0272 (cckuo@brand.usc.edu). The work of this author was supported by the University of Southern California Faculty Research and Innovation Fund.

‡ Department of Mathematics, University of California at Los Angeles, Los Angeles, California 90024. (chan@math.ucla.edu)

§ Department of Computer Science, University of California at Los Angeles, Los Angeles, California 90024 (tong@cs.ucla.edu).

¶ The work of the second and third authors was supported in part by the National Science Foundation under contract NSF-DMS87-14612, and by the Army Research Office under contract DAAL03-88-K-0085. Part of this work was performed while they were visiting the Research Institute for Advanced Computer Science, National Aeronautics and Space Administration Research Center.

preconditioners (ILU [25] and MILU [15]). These preconditioners have been very successful, especially when implemented on sequential computers.

In the parallel implementation of PCG methods, the major bottleneck is often the parallelization of the preconditioner, since the rest of the PCG methods can be parallelized in a straightforward way (the only potential bottleneck is the need for innerproducts, but many parallel computers do support fast inner-product evaluations). Unfortunately, previous works [12], [16] have shown that for many of the classical preconditioners, there is a fundamental trade-off in the ease of parallelization and the rate of convergence. A principal obstacle to parallelization is the sequential manner in which many preconditioners traverse the computational grid—the data dependence implicitly prescribed by the method fundamentally limits the amount of parallelism available. Reordering the grid traversal (e.g., from natural to red-black ordering) or inventing new methods (e.g., polynomial preconditioners [2], [19]) to improve parallelization usually has an adverse effect on the rate of convergence [12], [23].

The fundamental difficulty can be traced to the global dependence of elliptic problems. An effective preconditioner must account for the global coupling inherent in the original elliptic problem. Preconditioners that use purely local information (such as red-black orderings and polynomial preconditioners) are fundamentally limited in their ability to improve the convergence rate. On the other hand, global coupling through a natural ordering grid traversal is not highly parallelizable. The fundamental challenge is therefore to construct preconditioners that maintain global coupling and are highly parallelizable. Ideas along this line have of course been explored in the development of multigrid methods as solution [10], [17] as well as preconditioning techniques [20], [21], and the more recently proposed hierarchical basis preconditioner [8], [29].

We are thus led to the consideration of preconditioners that share global information through a multilevel grid structure (ensuring a good convergence rate) but perform only local operations on each grid level (and hence highly parallelizable). Compared with a purely multigrid iteration, we have more flexibility in terms of the choice of inter- and intragrid level operators (such as interpolation, projection, and smoothing), since we are using the multilevel iteration within an outer conjugate gradient iteration. One preconditioner of this type has been proposed recently by Bramble, Pasciak, and Xu [9] and Xu [28]. The methods that we propose in this paper are quite similar to their preconditioner, and our digital filtering framework can be looked at as providing an alternative view of their method. It also allows the flexibility in deriving several variants. The approach taken in this paper and that of Bramble, Pasciak, and Xu differs from that of multigrid methods in that the smoothing operation in multigrid methods is replaced by a simple scaling operation. Other types of multilevel preconditioners have been studied by Vassilevski [27], Axelsson and Vassilevski [6], [7], Kuznetsov [24] and Axelsson [4].

The outline of the paper is as follows. In § 2, we describe our framework for deriving multilevel filtering preconditioners for a model problem on a single discretization grid. The basic framework is then extended to the multigrid discretization case in § 3. In § 4, we briefly survey several other preconditioners of the multilevel type. Numerical results for (model, variable coefficient, and discontinuous coefficient) problems in two and three dimensions are presented in § 5, comparing the performance of several multilevel preconditioners, including the usual multigrid method as a preconditioner, the hierarchical basis preconditioner, and the method of Bramble–Pasciak–Xu. Some brief concluding remarks are given in § 6.

We note that the main emphasis of the present paper is on the convergence behavior of these multilevel preconditioners—no attempt is made to assess their parallel efficiency. That will be the subject of a forthcoming paper.

2. Multilevel filtering preconditioners: Fundamentals.

2.1. Motivation. Consider the one-dimensional discrete Poisson equation on [0, 1] with zero boundary conditions on a uniform grid Ω_h,

$$\left(-\frac{1}{2}E+1-\frac{1}{2}E^{-1}\right)u_n=f_n, \qquad n=1,\cdots,N-1, \tag{2.1}$$

where $N = h^{-1} = 2^L$, with integer $L > 1$, and E is the shift operator on Ω_h. We denote the above system by

$$Au=f,$$

where A, u, and f correspond, respectively, to the discrete Laplacian, solution, and forcing functions. Clearly, A is a tridiagonal matrix with diagonal elements $-\frac{1}{2}$, 1 and $-\frac{1}{2}$. It is well known that the matrix A can be diagonalized as

$$A = W^T\Lambda_A W, \tag{2.2}$$

where Λ_A is a diagonal matrix

$$\text{diag}\,(\lambda_1,\cdots,\lambda_k,\cdots,\lambda_{N-1}), \qquad \lambda_k = 1-\cos\,(k\pi h),$$

and W is an order $(N-1)^2$ orthogonal matrix whose kth row is

$$w_k^T=\left(\frac{2}{N}\right)^{1/2}(\sin\,(k\pi h),\cdots,\sin\,(k\pi nh),\cdots,\sin\,(k\pi(N-1)h)). \tag{2.3}$$

The diagonalization of the matrix A can be interpreted as the decomposition of the driving and solution functions into their Fourier components, i.e.,

$$\hat{u}_k=\left(\frac{2}{N}\right)^{1/2}\sum_{n=1}^{N-1}u_n\sin\,(k\pi nh), \qquad \hat{f}_k=\left(\frac{2}{N}\right)^{1/2}\sum_{n=1}^{N-1}f_n\sin\,(k\pi nh),$$

$$k=1,2,\cdots,N-1.$$

One can easily verify that $\hat{u}_k$ and $\hat{f}_k$ are related via

$$\hat{A}(k)\hat{u}_k=\hat{f}_k, \qquad k=1,2,\cdots,N-1,$$

where

$$\hat{A}(k)=\lambda_k=1-\cos\,(k\pi h), \tag{2.4}$$

is known as the spectrum of the discrete Laplacian.

In order to invert A, we can make use of (2.2) and obtain a fast Poisson solver:

$$A^{-1} = W^T\Lambda_A^{-1}W. \tag{2.5}$$

The above procedure also serves as the general framework for fast Poisson solvers in cases of higher dimension. However, fast Poisson solvers are not generally applicable for nonseparable elliptic operators and irregular domains. Instead, we want to find good approximations to this solution procedure that are extensible to more general problems and then use them as preconditioners. The fundamental idea is to avoid the use of fast Fourier transform (FFT) and to use instead a sequence of filtering operations to approximate the desired spectral decomposition. This explains the motivation and the name of the multilevel filtering (MF) preconditioner proposed in this paper.

Our main idea for deriving the MF preconditioner for A is to divide all admissible wavenumbers into bands and to approximate the spectrum $\hat{A}(k)$ at each band with some

constant. To be more precise, consider the following piecewise constant function in the wavenumber domain

$$\hat{P}(k) = c_l, \qquad k \in B_l, \qquad 1 \leqq l \leqq L,$$

where

$$B_l = \{k : 2^{l-1} \leqq k < 2^l \quad \text{and} \quad k \in I\},$$

is the lth wavenumber band. Let Λ_P be the diagonal matrix with $\hat{P}(k)$ as the kth diagonal element, i.e.,

$$\Lambda_P = \operatorname{diag}(\hat{P}(1), \hat{P}(2), \cdots, \hat{P}(N-1)),$$

and $P = W^T \Lambda_P W$. Then, the P-preconditioned Laplacian becomes

$$P^{-1}A = W^T \Lambda_{P^{-1}A} W,$$

where

$$\Lambda_{P^{-1}A} = (\Lambda_P)^{-1} \Lambda_A = \operatorname{diag}\left(\frac{\lambda_1}{c_1}, \frac{\lambda_2}{c_2}, \frac{\lambda_3}{c_2}, \cdots, \frac{\lambda_{2^{l-1}}}{c_l}, \cdots, \frac{\lambda_{2^l-1}}{c_l}, \cdots, \frac{\lambda_{N-1}}{c_L}\right).$$

The question is how to choose appropriate c_l's to reduce the condition number $\kappa(P^{-1}A)$.

Suppose that we can find c_l's so that

$$C_1 \leqq \frac{\lambda_k}{c_l} \leqq C_2, \qquad k \in B_l, \qquad 1 \leqq l \leqq L,$$

where C_1 and C_2 are positive constants independent of h. Then, P and A are spectrally equivalent. There are many ways to achieve this goal. For example, we can choose any eigenvalue λ within band B_l to be the constant c_l. For the following discussion, let us consider the choice,

$$c_l = 4^{-(L-1)}. \tag{2.6}$$

The ratio of $\hat{A}(k)$ and $\hat{P}(k)$ is then bounded by

$$4^{L-1}[1 - \cos(2^{-L+l-1}\pi)] \leqq \hat{P}^{-1}(k)\hat{A}(k) < 4^{L-l}[1 - \cos(2^{-L+l}\pi)],$$

for $k \in B_l$. The largest and smallest values of $\hat{P}^{-1}(k)\hat{A}(k)$ for $k \in B$ are bounded. They are, respectively,

$$\lambda_{\max}(P^{-1}A) = \max_k \hat{P}^{-1}(k)\hat{A}(k) < \max_{1 \leqq l \leqq L} 4^{L-l}[1 - \cos(2^{-L+l}\pi)] < \frac{\pi^2}{2},$$

and

$$\lambda_{\min}(P^{-1}A) = \min_k \hat{P}^{-1}(k)\hat{A}(k) \geqq \min_{1 \leqq l \leqq L} 4^{L-l}[1 - \cos(2^{-L+l-1}\pi)] \geqq 1.$$

Note that the last inequalities in the equations above hold independent of L, or equivalently, the grid size h. Thus, the condition number κ of the preconditioned operator $P^{-1}A$ is bounded by a constant

$$\kappa(P^{-1}A) < \frac{\pi^2}{2} \approx 4.93.$$

We plot the spectra $\hat{A}(k)$, $\hat{P}^{-1}(k)$, and $\hat{P}^{-1}(k)A(k)$ in Fig. 2.1 for $N = h^{-1} = 256$ with c_l defined in (2.6).

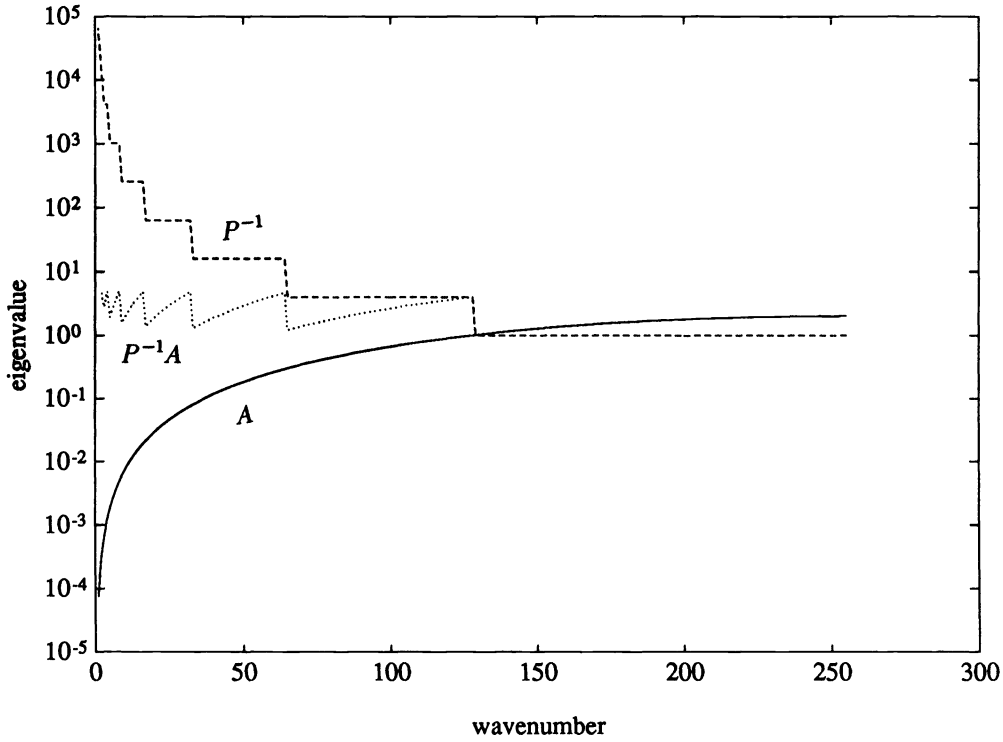


FIG. 2.1. *Spectra of* A, P^{-1} *and* $P^{-1}A$.

2.2. Decomposition and synthesis based on filtering. The preconditioning procedure

$$P^{-1}r = W^T \Lambda_P^{-1} W r, \tag{2.7}$$

consists of three building blocks: decomposition, scaling, and synthesis. The construction of these building blocks with ideal digital filters will be discussed in this section.

Let us rewrite (2.7) as

$$P^{-1}r = \left(\sum_{l=1}^{L} \frac{1}{c_l} W_l^T W_l \right) r, \tag{2.8}$$

where W_l, $1 \leqq l \leqq L$, are $(N-1)^2$ square matrices which have the same 2^{l-1} to $2^l - 1$ rows as W and zero vectors for remaining rows. If we implement W_l and W_l^T in decomposition and synthesis respectively, FFT and inverse FFT are needed. This is due to the fact that W_l is a mapping from the space domain to the wavenumber domain, whereas W_l^T is a mapping from the wavenumber domain to the space domain. By performing $P^{-1}r$ according to (2.8), we are led to an algorithm similar to the fast Poisson solver (2.5).

Let $F_l = W_l^T W_l$. Then, F_l is a mapping from the space domain to the space domain. In addition, we have

$$F_l = W^T \Lambda_{F_l} W,$$

where Λ_{F_l} is a diagonal matrix whose kth element is

$$\hat{F}_l(k) = \begin{cases} 1, & k \in B_l \\ 0, & \text{otherwise.} \end{cases}$$

The spectral property of F_l is characterized by $\hat{F}_l(k)$. A digital filter is a mapping from the space domain to the space domain satisfying a certain spectral property. Since F_l passes Fourier components in band B_l and blocks components in other bands, it is called a bandpass filter. We might perform the preconditioning (2.8) by implementing F_l's with digital filters in decomposition and a simple addition operation in synthesis. However, the resulting scheme loses a certain symmetrical property in decomposition and synthesis. This turns out to be important in the multigrid context (see § 3).

This motivates us to write (2.7) in another form as

$$P^{-1}r=\left(\sum_{l=1}^{L}\frac{1}{c_l}F_l^TF_l\right)r, \tag{2.9}$$

where bandpass filters $F_l\ (=F_l^T)$ are implemented in both decomposition and synthesis building blocks. In the context of multirate signal processing [13], the separation of a function into several components, each of which is confined to a narrow wavenumber band, is known as the *filter bank analyzer* and the reverse process is the *filter bank synthesizer*. Although there exist many ways to implement the filter bank analyzer and synthesizer, a simple design illustrated by the block diagram of Fig. 2.2 will be sufficient for our purpose. This design, called the single-grid multilevel filtering (SGMF) preconditioner, is based on the cascade of a sequence of elementary filters H_L, H_{L-1}, $\cdots$, H_2, where the function of H_l is to preserve Fourier components contained in bands B_1, $\cdots$, B_{l-1} and to eliminate Fourier components contained in band B_l. In terms of mathematics, we define

$$H_l=W^T\Lambda_{H_l}W, \tag{2.10a}$$

where Λ_{H_l} is a diagonal matrix with the kth element

$$\hat{H}_l(k)=\begin{cases}1, & k\in B_1\cup\cdots\cup B_{l-1}\\ 0, & k\in B_l.\end{cases} \tag{2.10b}$$

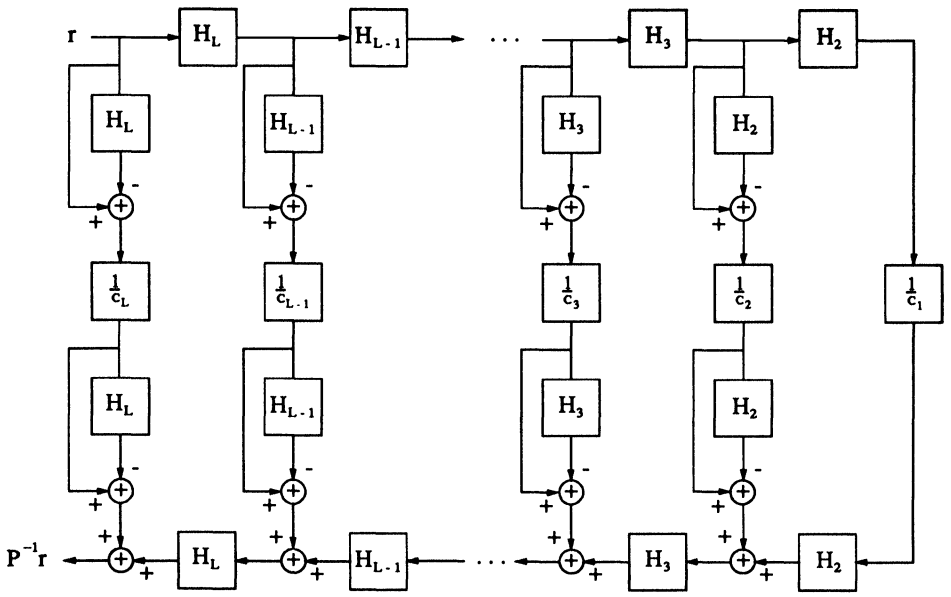


FIG. 2.2. *Block diagram of the* MF *preconditioner with a single discretization grid* (SGMF).

From Fig. 2.2, we see that the filters F_l are related to the filters H_l via

$$F_L = I - H_L, \tag{2.11a}$$

$$F_l = (I - H_l)\left[\prod_{p=l+1}^{L} H_p\right], \qquad 2 \leqq l \leqq L-1, \tag{2.11b}$$

$$F_1 = \prod_{p=2}^{L} H_p. \tag{2.11c}$$

It is easy to verify that F_l's satisfy the desired bandpass characteristics by pre- and post-multiplying (2.11) with W and W^T, respectively. Note also that the values of $\hat{H}_l(k)$ for $k \in B_{l+1} \cup \cdots \cup B_L$ do not influence the bandpass feature of F_l's. This observation simplifies the design of H_l's (see § 2.3).

To save computational work, we can further simplify the SGMF preconditioner in Fig. 2.2 by deleting the paths and the associated work corresponding to $I - H_l$. As given in Fig. 2.3, we have the modified SGMF preconditioner

$$Q^{-1}r = \left(\sum_{l=1}^{L} \frac{1}{d_l} G_l^T G_l\right) r, \tag{2.12}$$

where

$$G_L = I,$$

$$G_l = \prod_{p=l+1}^{L} H_p, \quad \text{for} \quad 1 \leqq l \leqq L-1.$$

Note that bandpass filters F_l in the preconditioner P have been replaced by lowpass filters G_l in the preconditioner Q. By choosing d_l's appropriately, we can make Q behave the same as P. With the preconditioner Q, Fourier components of band B_l exist in the first $L - l + 1$ levels and these components are multiplied by $d_L^{-1}, \cdots, d_l^{-1}$, respectively. Therefore, the scaling constants d_l's are implicitly defined via

$$\sum_{i=l}^{L} \frac{1}{d_i} = \frac{1}{c_l}. \tag{2.13}$$

Solving (2.13) for d_l gives

$$d_L = c_L \quad \text{and} \quad d_l = \frac{1}{c_l^{-1} - c_{l+1}^{-1}}, \quad l = L-1, \cdots, 1. \tag{2.14}$$

However, we observe from numerical experiments that the parameter sets $\{c_l\}$ and $\{d_l\}$

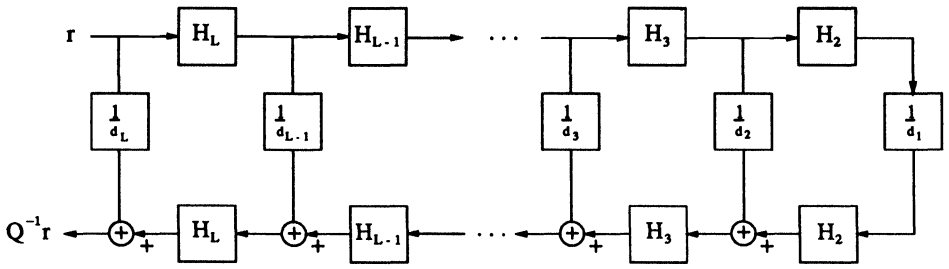


FIG. 2.3. *Block diagram of the modified* SGMF *preconditioner.*

used in Fig. 2.3 give about the same convergence rate. This can be explained by the observation that, for small l, $d_l \approx c_l$, since $c_l^{-1} \gg c_{l+1}^{-1}$.

2.3. Design of elementary filters. Consider the design of the filter H_L appearing at the first stage. The H_L have the following ideal lowpass characteristic,

$$\hat{H}_L(k) = \begin{cases} 1, & 0 \leqq k < 2^{L-1} \\ 0, & 2^{L-1} \leqq k \leqq 2^L. \end{cases} \tag{2.15}$$

From (2.10), we find that H_L is an $(N-1)^2$ full matrix. Thus, the operation $H_L v$ for an arbitrary vector v has a complexity proportional to $O(N^2)$. This is too expensive to perform. Therefore, we seek the approximation of the ideal lowpass filter H_L with a nonideal lowpass filter $H_{L,J}$, which is a symmetric band matrix of bandwidth $O(J)$ with the spectral property $\hat{H}_{L,J}(k) \approx \hat{H}_L(k)$ for $1 \leqq k \leqq N-1$. Consequently, the operation $H_{L,J} v$ only has a complexity proportional to $O(JN)$.

Let us write the nonideal lowpass filter of the form

$$H_{L,J} = a_0 + \sum_{j=1}^{J} a_j (E^j + E^{-j}), \tag{2.16}$$

where the coefficients a_0 and a_j's are to be determined. In order to define the operation

$$H_{L,J} v_n = a_0 + \sum_{j=1}^{J} a_j (v_{n+j} + v_{n-j})$$

for any vector v_n appropriately, the odd-periodic extension of v_n is assumed,

$$v_{-n} = -v_n \quad \text{and} \quad v_{n+2pN} = v_n, \quad \text{for integer } p.$$

This implies that $H_{L,J}$ corresponds to a circulant matrix. The above odd-periodic assumption is used only for analyzing and designing $H_{l,J}$'s in this section. The actual implementation of the MF preconditioner with a multigrid discretization described in § 3 does not rely on this assumption.

There are numerous ways to determine the coefficients a_0 and a_j's depending on what approximation criteria are to be used. The operator $H_{L,J}$ has the eigenfunction $\sin(k\pi nh)$ with the eigenvalue

$$\hat{H}_{L,J}(k) = a_0 + 2 \sum_{j=1}^{J} a_j \cos(k\pi jh).$$

Here we consider a class of lowpass filters based on the following two criteria:

(1) $\hat{H}_{L,J}\left(\frac{N}{2}\right) = \frac{1}{2}$ and $\hat{H}_{L,J}(k) - \frac{1}{2} = -\left[\hat{H}_{L,J}(N-k) - \frac{1}{2}\right]$,

(2) $\hat{H}_{L,J}(0) = 1$ and the first jth derivatives ($1 \leqq j \leqq J$) of $\hat{H}_{L,J}(0)$ are all zero.

The first criterion implies that the function $\hat{H}_{L,J}(k) - \frac{1}{2}$ is odd symmetric with respect to $k = N/2$. A direct consequence of this criterion is that

$$a_0 = \frac{1}{2} \quad \text{and} \quad a_j = 0, \; j \text{ positive even.}$$

The second criterion, called the *maximally flat* criterion [18], requires the approximation at the origin to be as accurate as possible. It is used to determine a_j with odd j. In Table 2.1, we list coefficients a_j for $J = 1, 3, 5$ obtained according to criteria (1) and (2) and plot their spectra in Fig. 2.4 with $N = 2^8 = 256$. The larger J becomes, the better the approximation is.

TABLE 2.1
Coefficients of a class of nonideal lowpass filters.

J	a_0	a_1	a_3	a_5
1	$\frac{1}{2}$	$\frac{1}{4}$	0	0
3	$\frac{1}{2}$	$\frac{9}{32}$	$\frac{-1}{32}$	0
5	$\frac{1}{2}$	$\frac{150}{512}$	$\frac{-25}{512}$	$\frac{3}{512}$

As illustrated in Figs. 2.2 and 2.3, the low wavenumber band of the function r is used as the input to the filter H_{L-1} at the next stage. The filter H_{L-1} can be constructed with the same set of coefficients used by H_L, i.e.,

$$H_{L-1,J} = a_0 + \sum_{j=1}^{J} a_j (E^{2j} + E^{-2j}). \tag{2.17}$$

Comparing (2.16) and (2.17), we see that the only difference between $H_{L,J}$ and $H_{L-1,J}$ is the position of grid points used for averaging. For the first-stage filter $H_{L,J}$, local averaging is used. For the second-stage filter $H_{L-1,J}$, we consider averaging between points separated by $2h$. This design is due to the following reason. From (2.17), we see that the filter $H_{L-1,J}$ has the spectrum

$$\hat{H}_{L-1,J}(k) = a_0 + 2 \sum_{j=1}^{J} a_j \cos(k\pi j 2h),$$

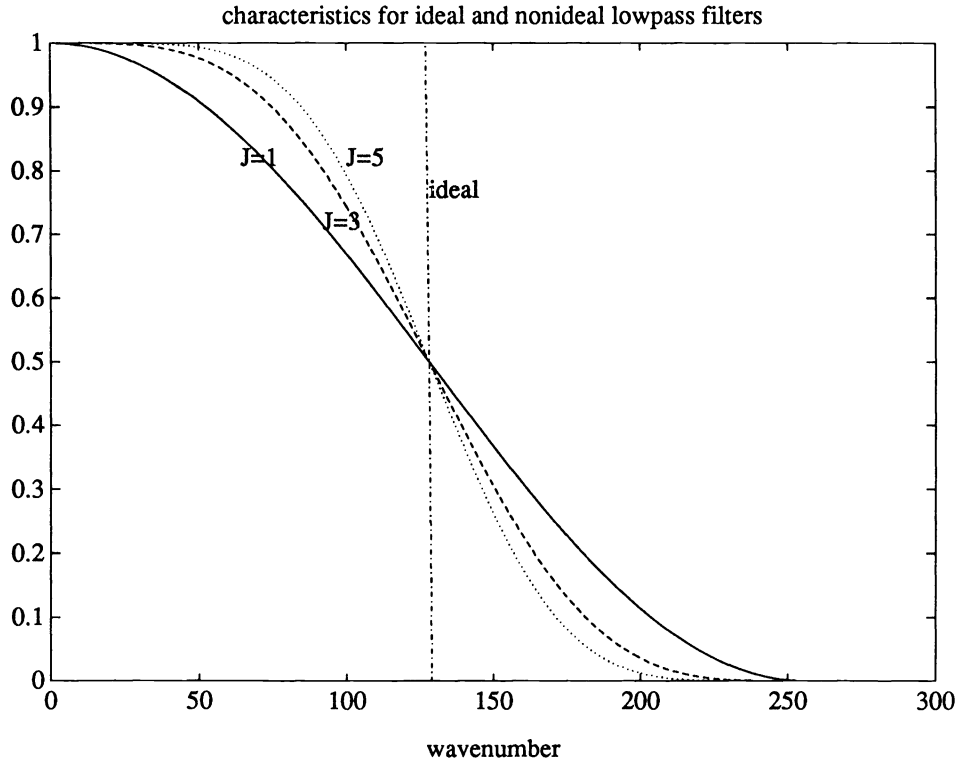


FIG. 2.4. *Spectra of maximally flat lowpass filters $H_{L,J}$ with $J = 1, 3, 5$.*

and that $\hat{H}_{L-1,J}(k)$ is related to $\hat{H}_{L,J}(k)$ via

$$\hat{H}_{L-1,J}(k) = \hat{H}_{L,J}(2k).$$

Consequently, for functions consisting only of components in low wavenumber region $1 \leqq k < 2^{L-1}$, $\hat{H}_{L-1}$ behaves like a lowpass filter, which preserves components in the region $1 \leqq k < 2^{L-2}$ and filters out components in the region $2^{L-2} \leqq k < 2^{L-1}$. However, note that H_l, $l < L$ is not a lowpass filter with respect to the entire wavenumber band.

By applying the same procedure recursively, we can approximate the general elementary filter H_l on a uniform infinite grid as

$$H_{l,J} = a_0 + \sum_{j=1}^{J} a_j (E^{2^{L-l}j} + E^{-2^{L-l}j}), \qquad 2 \leqq l \leqq L, \tag{2.18}$$

where the coefficients a_j's are listed in Table 2.1. The spectrum of $H_{l,J}$ is

$$\hat{H}_{l,J}(k) = a_0 + 2 \sum_{j=1}^{J} a_j \cos(k\pi j 2^{L-l} h), \qquad 2 \leqq l \leqq L. \tag{2.19}$$

According to (2.11), we can construct nonideal bandpass filters $F_{l,J}$ with nonideal elementary filters $H_{l,J}$,

$$F_{L,J} = I - H_{L,J}, \tag{2.20a}$$

$$F_{l,J} = (I - H_{l,J}) \left(\prod_{p=l+1}^{L} H_{p,J} \right), \qquad 2 \leqq l \leqq L-1, \tag{2.20b}$$

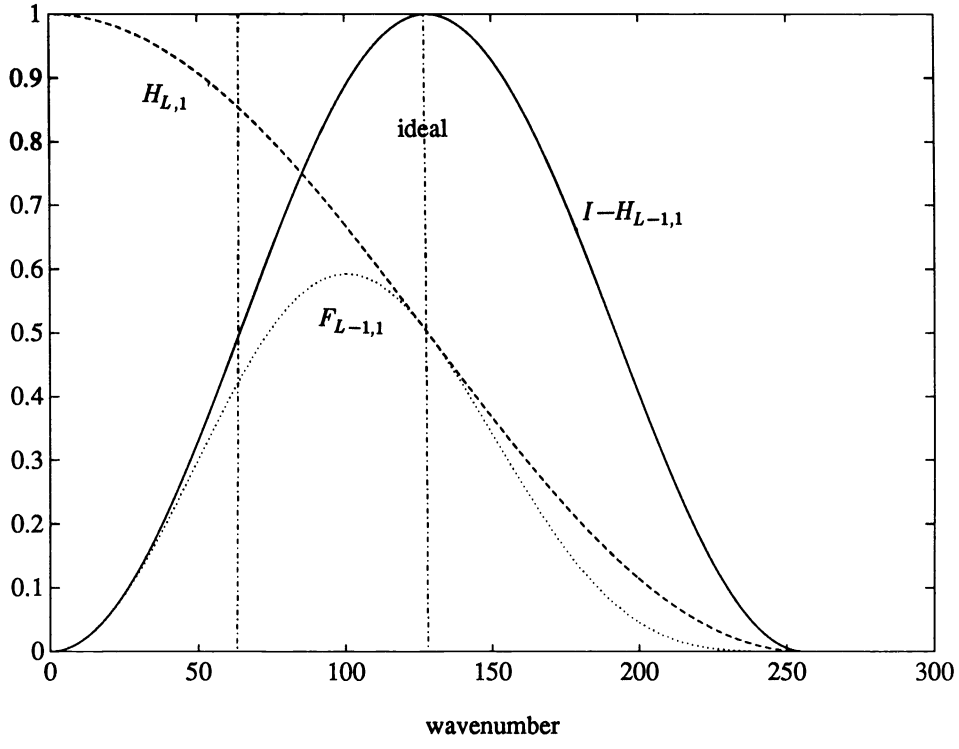


FIG. 2.5. *Spectra of* $H_{L,J}$, $I - H_{L-1,J}$ *and* $F_{L-1,J}$ *with* $J = 1$.

$$F_{1,J} = \prod_{p=2}^{L} H_{p,J}. \tag{2.20c}$$

To give an example, the construction of $F_{L-1,J}$ with $J = 1$ is illustrated in Fig. 2.5. Note that the elementary filter $H_{l,J}$ given by (2.18) is symmetric. So is the bandpass filter $F_{l,J}$. Finally, we obtain the nonideal MF-preconditioner

$$P_J^{-1} r = \left(\sum_{l=1}^{L} \frac{1}{c_l} F_{l,J}^T F_{l,J} \right) r, \tag{2.21}$$

which approximates the ideal MF-preconditioner P given by (2.9).

It is worthwhile to summarize the similarities and differences between the fast Poisson solver (2.5) and the SGMF preconditioning (2.21). They are both based on spectral decomposition. The fast Poisson solver decomposes a function into its Fourier components through the FFT, whereas the MF preconditioner approximately decomposes it into a certain number of bands through filtering. The filtering operations, which correspond to local averaging processes, can be easily adapted to irregular grids and domains and variable coefficients. In contrast, the FFT is primarily applicable to constant coefficient problems with regular grids and domains. Besides, for the fast Poisson solver we usually require detailed knowledge of the spectrum. But for the MF preconditioner we have only to estimate how the spectrum varies from one band to another.

2.4. Fourier analysis and higher-dimensional cases. Since the MF preconditioner P_J and the Laplacian A share the same eigenvectors, i.e., Fourier sine functions, the spectrum and condition number of the MF-preconditioned Laplacian can be analyzed conveniently by Fourier analysis. From (2.20), we have the following spectral relationship

$$\hat{F}_{L,J}(k) = 1 - \hat{H}_{L,J}(k), \tag{2.22a}$$

$$\hat{F}_{l,J}(k) = (1 - \hat{H}_{l,J}(k)) \left(\prod_{p=l+1}^{L} \hat{H}_{p,J}(k) \right), \qquad 2 \leqq l \leqq L-1, \tag{2.22b}$$

$$\hat{F}_{1,J}(k) = \prod_{p=2}^{L} \hat{H}_{p,J}(k), \tag{2.22c}$$

where $\hat{H}_{l,J}(k)$, $1 \leqq l \leqq L$, are given by (2.19). Using (2.4), (2.6), and (2.22), we can determine the eigenvalues of $P_J^{-1}A$,

$$\lambda(P_J^{-1}A) = \hat{P}_J^{-1}(k)\hat{A}(k) = \sum_{l=1}^{L} \frac{1}{c_l} \hat{F}_{l,J}^T(k) \hat{F}_{l,J}(k) \hat{A}(k).$$

The eigenvalues $\lambda(P_J^{-1}A)$ are plotted as a function of k with $J = 1$, 3, 5 and $h^{-1} = 256$ in Fig. 2.6. We should compare these spectra with that in Fig. 2.1 based on the ideal filtering assumption. All of them have one common feature. That is, eigenvalues are redistributed in such a way that there exist many local maxima and minima. The condition numbers for $J = 1$, 3, 5 are 2.50, 1.88, and 1.93, respectively. Note that these numbers are in fact smaller than the condition number 4.93 obtained with ideal filtering. The precise reason for this phenomenon is still not clear to us. It might be related to the smoothness of the eigenvalue distribution curves. The eigenvalue distribution for $P^{-1}A$ in Fig. 2.1 has many keen edges. However, these edges are smoothed by nonideal digital filters as shown in Fig. 2.6.

The generalization of the MF preconditioner to two- or three-dimensional problems on square or cube domains can be done straightforwardly. For example, we may construct

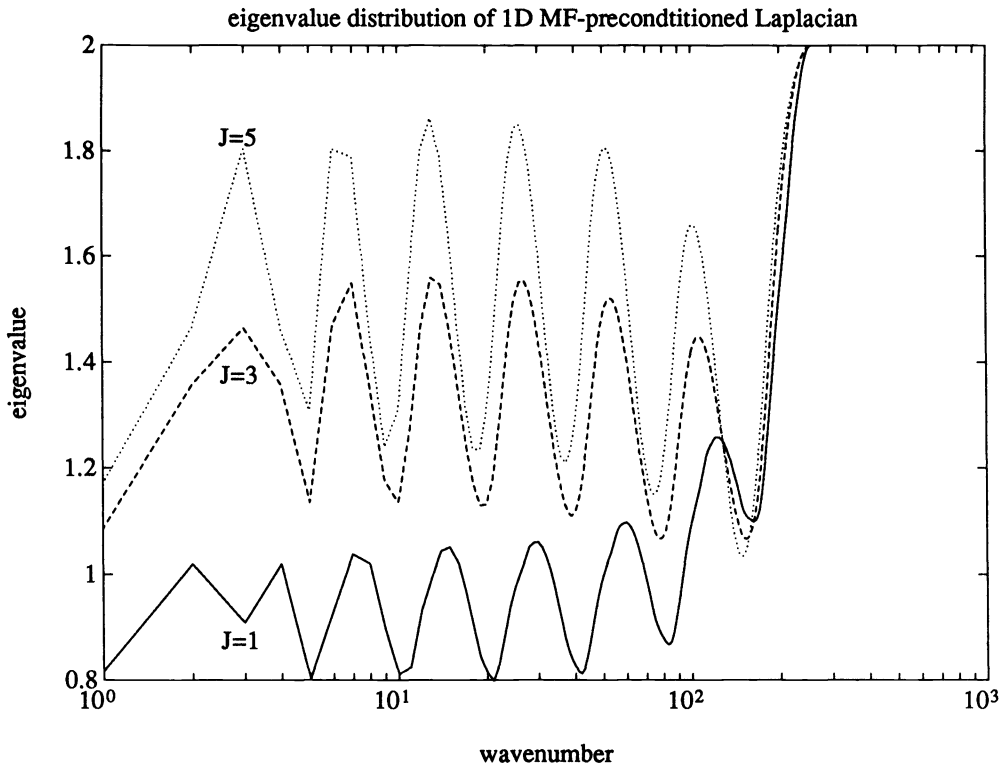


FIG. 2.6. *Eigenvalues of* $P_J^{-1}A$ *with* $J = 1, 3, 5$.

the two-dimensional elementary filter by the tensor product of one-dimensional elementary filters along the x- and y-directions,

$$H_{l,J} = \left(a_0 + \sum_{j=1}^{J} a_j (E_x^{2^{L-l_j}} + E_x^{-2^{L-l_j}}) \right) \times \left(a_0 + \sum_{j=1}^{J} a_j (E_y^{2^{L-l_j}} + E_y^{-2^{L-l_j}}) \right),$$

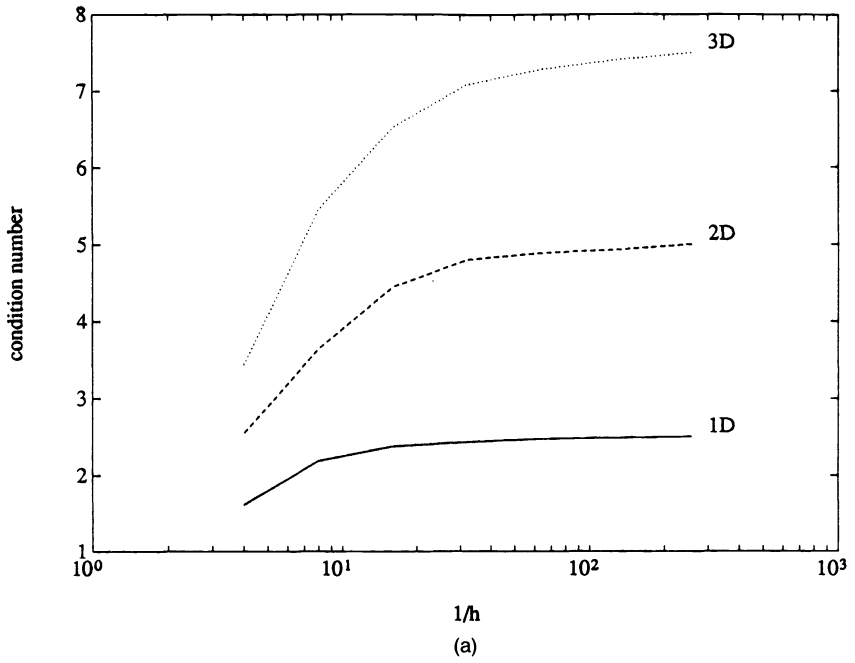
which can be further simplified by using operator algebra [14]. For example, the coefficients for $H_{L,1}$ can be written in stencil form as

$$H_{L,1} : \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}. \tag{2.23}$$

Similarly, the three-dimensional elementary filter can be obtained by the tensor product of three one-dimensional filters along the x-, y- and z-directions.

The condition numbers of one-, two-, and three-dimensional MF-preconditioned Laplacians with two types of nonideal filters ($J = 1$ and $J = 3$) are computed and plotted as functions of the grid size h in Figs. 2.7 (a) and (b). These figures show that P_J and A are spectrally equivalent.

The discussion in § 2.3 is based on the odd-periodic property of the sequence v_n. However, this may not be easily implementable for general multidimensional problems with nonrectangular domains. The difficulty arises when the size of $H_{l,J}$ is so large that it operates on points outside the domain. There are two possible solutions. It may be preferable to construct filters of larger size by the repeated application of filters of smaller size. For example, we can apply the filter $H_{L,J}$ (2.16) with $J = 1$ twice. This is equivalent



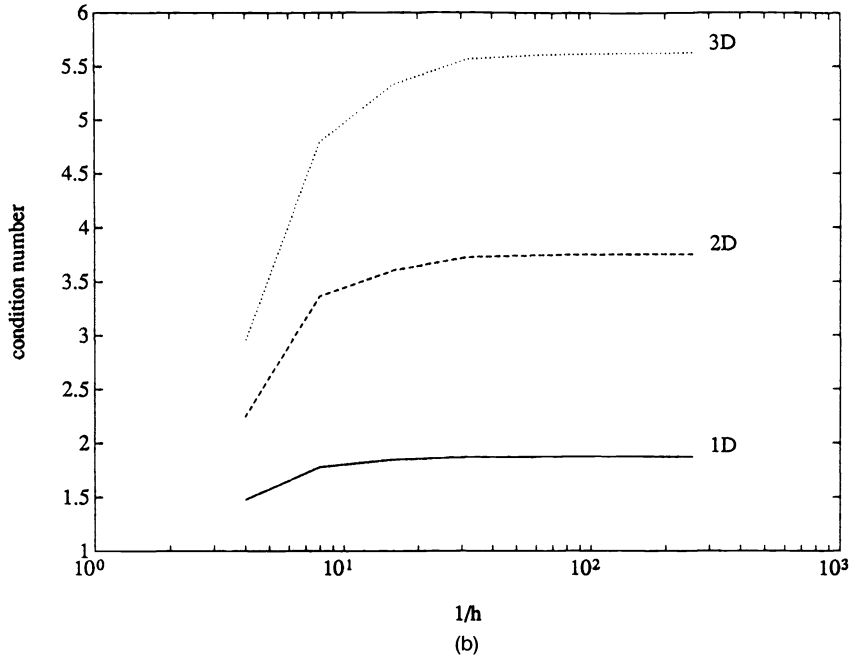


FIG. 2.7. *Condition numbers of the* MF-*preconditioned Laplacian with* (a) $J = 1$ *and* (b) $J = 3$.

to a filter of size 5,

$$H_{L,1}^2 = \left(\frac{1}{4}E^{-1} + \frac{1}{2} + \frac{1}{4}E\right)^2 = \frac{1}{16}E^{-2} + \frac{1}{4}E^{-1} + \frac{3}{8} + \frac{1}{4}E + \frac{1}{16}E^2.$$

Another possibility is to apply smaller filters at points close to boundaries and larger

filters at points far away from boundaries. Note also that, for fixed J, the size of the elementary filter $H_{l,J}$ increases as l decreases. However, this problem can be resolved by incorporating the multigrid discretization structure into the above multilevel filtering framework as described in § 3.

3. Multigrid multilevel filtering (MGMF) preconditioners. In § 2, we discussed the construction of the MF preconditioner for the model Poisson problem based on a single discretization grid. This section will discuss the generalization of this preconditioning technique so that it can be implemented more efficiently and applied to more general self-adjoint elliptic partial differential equation (PDE) problems.

The filtering operation described above is performed at every grid point at all levels $2 \leqq l \leqq L$. Since there are $O(\log N)$ levels and $O(JN)$ operations per level, where N and J denote, respectively, the order of unknowns and the filter size, the total number of operations required is proportional to $O(JN \log N)$. However, since waveforms consisting only of low wavenumber components can be well represented on coarser grids, we can use the multigrid philosophy [10], [17] and incorporate the multigrid discretization structure into the filtering framework described in § 2. That is, we construct a sequence of grids Ω_l of sizes $h_l = O(2^{-l})$, $1 \leqq l \leqq L$, to represent the decomposed components. Then, the total number of unknowns is $O(N)$ and consequently the total number of operations per MF preconditioning step is $O(JN)$. Note that J is a constant independent of N.

The block diagram of the multigrid multilevel filtering (MGMF) preconditioner is depicted in Fig. 3.1. It is obtained by inserting down-sampling (I_l^{l-1}) and up-sampling (I_{l-1}^l) operators into the SGMF preconditioner. With the notation commonly used in the multigrid literatures, the down-sampling and up-sampling operators for grids Ω_l ($h_l = 2^{L-l}h$) and Ω_{l-1} ($h_{l-1} = 2^{L-l+1}h$) can be defined as

$$I_l^{l-1}: \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{vmatrix}_{l}^{l-1}, \qquad I_{l-1}^{l}: \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{vmatrix}_{l-1}^{l}.$$

It is easy to verify that a lowpass filter followed by a down-sampling operator is the same as the restriction operator in MG methods, whereas an up-sampling operator followed by a lowpass filter is equivalent to the interpolation operator [22].

Given a sequence of grids Ω_l, $1 \leqq l \leqq L$, down-sampling (I_{l+1}^l) and up-sampling (I_l^{l+1}) operators between grids Ω_l and Ω_{l+1}, and appropriate elementary filters H_l defined on Ω_l, the algorithm corresponding to the block diagram given by Fig. 3.1 can be summarized as in Table 3.1.

TABLE 3.1
Computation of $M^{-1}r$.

Decomposition:
  $v_L := r$,
  for $l = L-1, \cdots, 1$
  $v_l := I_{l+1}^l H_{l+1} v_{l+1}$,
Scaling:
  for $l = L, \cdots, 1$
  $w_l := v_l d_l^{-1}$
Synthesis:
  $s_1 := w_1$,
  for $l = 2, \cdots, L$
  $s_l := w_l + H_l I_{l-1}^l s_{l-1}$
  $M^{-1}r := s_L$

This is the MGMF algorithm implemented in §5.

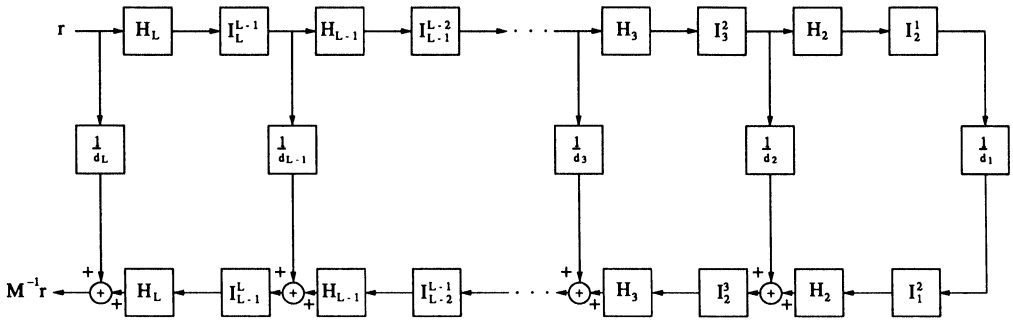


FIG. 3.1. *Block diagram of the modified* MGMF *preconditioner.*

The preconditioning $M^{-1}r$ can be viewed as a degenerate multigrid method, for which we have a sequence of restriction and interpolation operations but where the error smoothing at each grid level is replaced by an appropriate scaling. This observation leads us to generalize the MF preconditioner to the case of nonuniform grids commonly obtained from the finite-element discretization. That is, one can view projection as decomposition and interpolation as synthesis and any multigrid method can be used as an MGMF preconditioner if we replace the potentially more expensive error smoothing by a simple scaling. It is well known that the eigenvalue λ_k in band B_l (see § 2.1) behaves like $O(h_l^{-2})$, where h_l describes approximately the grid spacing for level l [9]. Therefore, a general rule for selecting the scaling constant c_l at grid level l is

$$c_l = O(h_l^{-2}).$$

This generalized version is closely related to the preconditioner by Bramble, Pasciak, and Xu [9]. They derived their preconditioner in the finite-element context discretized with the nested triangular elements. From our filtering framework, the corresponding elementary filter H_L takes the form

$$H_{L,BPX}: \frac{1}{8}\begin{vmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{vmatrix}, \tag{3.1}$$

which is different from $H_{L,1}$ given by (2.23). We can derive other filters from (3.1) by applying it more than once. For example, by applying it twice, we get

$$H_{L,TBPX}: \frac{1}{64}\begin{vmatrix} 0 & 0 & 1 & 2 & 1 \\ 0 & 2 & 6 & 6 & 2 \\ 1 & 6 & 10 & 6 & 1 \\ 2 & 6 & 6 & 2 & 0 \\ 1 & 2 & 1 & 0 & 0 \end{vmatrix}. \tag{3.2}$$

In order to eliminate the directional preference, we can apply (3.1) in alternating directions to give a symmetric filter:

$$H_{L,ADBPX}: \frac{1}{64}\begin{vmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \\ 2 & 6 & 8 & 6 & 2 \\ 1 & 4 & 6 & 4 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{vmatrix}. \tag{3.3}$$

The MF preconditioner is designed to capture the spectral property (or h-dependency) of a discretized elliptic operator but not the variation of its coefficients. This is

also true for the hierarchical basis and BPX preconditioners. In order to take badly scaled variable coefficients into account, we use the MF preconditioner in association with diagonal scaling in our experiments [16]. The diagonal scaling is often used for cases where the diagonal elements of the coefficient matrix A vary for a wide range. Suppose that the coefficient matrix can be written as

$$A = D^{1/2} \tilde{A} D^{1/2},$$

where we choose D to be a diagonal matrix with positive elements in such a way that the diagonal elements of $\tilde{A}$ are of the same order, say, $O(1)$. Then, in order to solve $Au = f$, we can solve an equivalent problem $\tilde{A}\tilde{u} = \tilde{f}$, where $\tilde{u} = D^{1/2}u$ and $\tilde{f} = D^{-1/2}f$, with the MF preconditioner. There exist other ways to incorporate the coefficient information into preconditioners of the multilevel type, say, to use the Gauss-Seidel smoothing suggested by Bank, Dupont, and Yserentant [8].

4. Brief survey of multilevel preconditioners. In this section, we very briefly survey other multilevel preconditioners that have been proposed in the literature and their relationships to one another.

4.1. Multigrid preconditioner (MG). A natural choice for a multilevel preconditioner is to use a fixed number of cycles of a conventional multigrid method. This approach was explored early on in the development of multigrid methods [20], [21]. The basic operations on each grid are interpolation, projection, and smoothing operations, each of which can be easily designed to be highly parallelizable. For example, in the V-cycle strategy, each grid is visited exactly twice in each preconditioning step, once going from fine to coarse grids and once coming back from coarse to fine. However, for highly irregular problems, such as singularities in the solutions due to reentrant corners and highly discontinuous coefficients, it is not clear how to choose the smoothing operations and the performance can deteriorate.

4.2. Hierarchical basis preconditioner (HB). Another preconditioning technique of multilevel type is the hierarchical basis method [8], [29]. The name refers to the space of hierarchical basis functions defined on the grid hierarchy. The usual nodal basis functions are used except that those defined at grid points on a given level which also belong to coarser levels are omitted. Let the hierarchical basis functions be denoted by ψ_j^l, where l denotes the grid level and j the index of the basis function on that level. Then, the action of the inverse of the hierarchical basis preconditioner M on a function v can be written as,

$$M^{-1}v = \sum_l \sum_j (v, \psi_j^l)\psi_j^l,$$

which takes the discretized form $SS^T v_h$ and can be computed by a V-cycle with the matrix S^T corresponding to a fine-to-coarse grid traversal and S to a coarse-to-fine traversal. On each level, only local operations are performed. In two dimensions, the condition number of the preconditioned system can be shown to grow like $O(\log^2 h^{-1})$, which is very slow. Unfortunately, this nice property is lost in three dimensions, where the growth is $O(h^{-1})$ [26], [29]. However, these theoretical results are proven under much weaker regularity assumptions than for the multigrid methods. Moreover, the computational work per step is $O(h^{-1})$ even for highly nonuniform and refined meshes. For numerical experiments on parallel computers, see [1], [16].

4.3. Method by Bramble–Pasciak–Xu (BPX). Very recently, Bramble–Pasciak–Xu [9], [28] proposed the following preconditioner for second-order elliptic problems

in R^d:

$$M^{-1}v = \sum_l h_l^{2-d} \sum_j (v, \phi_j^l)\phi_j^l,$$

where ϕ_j^l are the nodal basis functions and h_l is the measure of the mesh size at grid level l. Since the form of their preconditioner is similar to that for the hierarchical basis preconditioner, the computations can be arranged in a similar way via a V-cycle. They proved that the condition number of the preconditioned operator can be bounded by $O(\log h^{-1})$ for problems with smooth solutions, by $O(\log^2 h^{-1})$ for problems with crack type singularities, and by $O(\log^3 h^{-1})$ for problems with discontinuous coefficients. In 3D, this is a significant improvement over the hierarchical basis preconditioner.

4.4. Algebraic multilevel preconditioners (AMP). Vassilevski [27] proposed a different approach to derive multilevel preconditioners. He used the standard nodal basis functions and a multilevel ordering of the nodes of the discretization, in which nodes at a given level belonging to a coarser grid are ordered after the other nodes. He then considered an approximate block factorization of the stiffness matrix in this ordering, in which the Schur complement at a given grid level is approximated by iteration with the preconditioner of the stiffness matrix recursively defined at the current level. He showed that, with one iteration at each level, the condition number of the preconditioned system can be bounded by $O(\log h^{-1})$. A similar method has been proposed by Kuznetsov [24]. Later, Axelsson-Vassilevski [6], [7] improved this bound to $O(1)$ by carrying out recursively more (Chebyshev) iterations with the preconditioner at each level. Axelsson [4] also showed that the same technique can be applied when hierarchical basis functions are used instead of the nodal basis. Note that when the number of iterations at each level exceeds 1, the grid traversal differs from all the previously mentioned V-cycle based methods. At this time, we have not included non-V-cycle type preconditioners in our numerical comparisons but plan to do so in the future.

4.5. Relationship among multilevel preconditioners. As can be seen from the discussion above, there are similarities among various multilevel preconditioners. Most of the multilevel preconditioners are in the form of a multigrid V-cycle (MG, HB, BPX, and MF, but not AMP). The MF preconditioner is very similar to the BPX method. The MF method allows some flexibility in the choice of filters (basically any multigrid residual averaging operator can be used) and does not depend on the use of a finite-element discretization with nested nodal basis functions. It also allows a single grid (i.e., nonmultigrid) version which may better suit massively parallel architecture computers. On the other hand, the finite-element framework allows an elegant proof of the asymptotic convergence behavior for rather general problems as is done in [9], [28], whereas the filtering framework is rigorously provable for constant coefficient model problems only (although much more detailed information can be obtained for them).

Finally, it is interesting to compare these preconditioners with the conventional multigrid method. Several of the preconditioners have the same form of a conventional multigrid cycle, except that the smoothing operations are omitted. For less regular problems where a good smoothing operator is hard to derive and could be quite expensive, one step of these preconditioners can be substantially less expensive than a corresponding step of the multigrid iteration. In a sense, one can view these preconditioners as efficiently capturing mesh size-dependent part of the ill-conditioning of the elliptic operator and leaves the other sources of ill-conditioning (e.g., discontinuous coefficients) to the conjugate gradient iteration. The combination of multigrid and conjugate gradient holds the promise of being both robust and efficient. However, to get a spectrally equivalent pre-

conditioner, it seems that one must go beyond the V-cycle and perform more iterations on each grid as in the AMP method.

5. Numerical experiments. In this section, we present numerical results for two- and three-dimensional test problems to compare the convergence behavior and the amount of work needed for various preconditioners. The preconditioners implemented are:

HB: hierarchical basis preconditioner using linear elements for two-dimensional and trilinear elements for three-dimensional problems,

MG(i, i): multigrid preconditioner with one V-cycle, where i is the number of pre- and post-smoothings,

BPX1: the BPX preconditioner for two-dimensional problems (H_L given by (3.1)),

BPX2: a modified version of BPX preconditioner by filtering twice for two-dimensional problems (H_L given by (3.2)),

BPX3: another modified version of BPX preconditioner by filtering twice but using linear elements of different orientations for two-dimensional problems (H_L given by (3.3)),

MGMF1: the MGMF preconditioner with the 9-point (2.23) or 27-point filter for two- and three-dimensional problems, respectively,

MGMF2: a modified version of MGMF preconditioner in which the 9-point (or 27-point) filter is applied twice,

MGMF3: another modified version of MGMF preconditioner in which the 9-point (or 27-point) filter is applied once at the finest grid level (to have a smaller amount of work compared to MGMF2) and twice at other grid levels (to achieve a faster convergence rate compared to MGMF1),

RIC: the relaxed incomplete Cholesky preconditioner [5] is included for the purpose of comparison. For the relaxation factor, we use the optimal value $\omega = 1 - 8 \sin^2(\pi h/2)$ from [11]. The number of iterations required for RIC can be bounded by $O(n^{1/2})$.

The preconditioning operation counts for each method, for two- and three-dimensional problems are given in Tables 5.1 and 5.2, respectively. These operation counts include addition, multiplication, and division (each is counted as one operation), but exclude overhead such as condition checking and data copying. The non-preconditioning operation counts required per PCG step for two-dimensional problems are $21N$, which include $6N$ for three inner products (one more inner product than the basic CG step, since we use the relative residual norm for convergence check), $6N$ for three SAXPY

TABLE 5.1
Work per iteration for preconditioners (2D).

Preconditioner	Operation count per iteration
RIC	9 N
HB	7 N
MG(1.1)	38 N
BPX1	8 N
BPX2	26 N
BPX3	26 N
MGMF1	9 N
MGMF2	27 N
MGMF3	15 N

TABLE 5.2
Work per iteration for preconditioners (3D).

Preconditioner	Operation count per iteration
RIC	13 N
HB	8 N
MGMF1 (BPX1)	9 N
MGMF2 (BPX2)	32 N
MGMF3	12 N

operations, and $9N$ for one matrix vector product. Similarly, the non-preconditioning operation counts per PCG step for three-dimensional problems are $25N$.

From Table 5.1, we observe that the operation counts per iteration for BPX1 and MGMF1 are much less than that of the MG(1, 1) preconditioners, because the former preconditioners do not need smoothing, which is expensive. In general, for two-dimensional problems, MG(i, i) preconditioner takes $(38 + 32 \times (i - 1))N$ operations. For example, MG(3, 3) preconditioning requires $102N$ operations. Also note that the application of filtering twice requires about three times the work of filtering once. This is because by filtering twice the filter stencil is extended from 9-point to 25-point (about three times as many points).

For three-dimensional problems, the operation count for BPX1 (BPX2) preconditioning using trilinear elements is the same as for the MGMF1 (MGMF2) preconditioning as shown in Table 5.2. The MG preconditioner has not yet been implemented for three-dimensional problems.

For all test problems, we use the standard 5- (or 7-) point stencil on a square (or cubic) uniform mesh with $h = n^{-1}$ and $N = (n - 1)^2$ (or $N = (n - 1)^3$), zero boundary conditions and zero initial guesses. Experimental results are given for different values of h and the stopping criterion is $\|r^k\| / \|r^0\| \leqq 10^{-6}$. Diagonal scaling is always used except for RIC. The six test problems are:

(1) the two-dimensional model problem with solution $u = x(x - 1)y(y - 1)e^{xy}$,

$$\Delta u = f, \qquad \Omega = (0,1)^2, \tag{5.1}$$

(2) a two-dimensional variable coefficient problem with solution $u = xe^{xy} \sin \pi x \times \sin \pi y$,

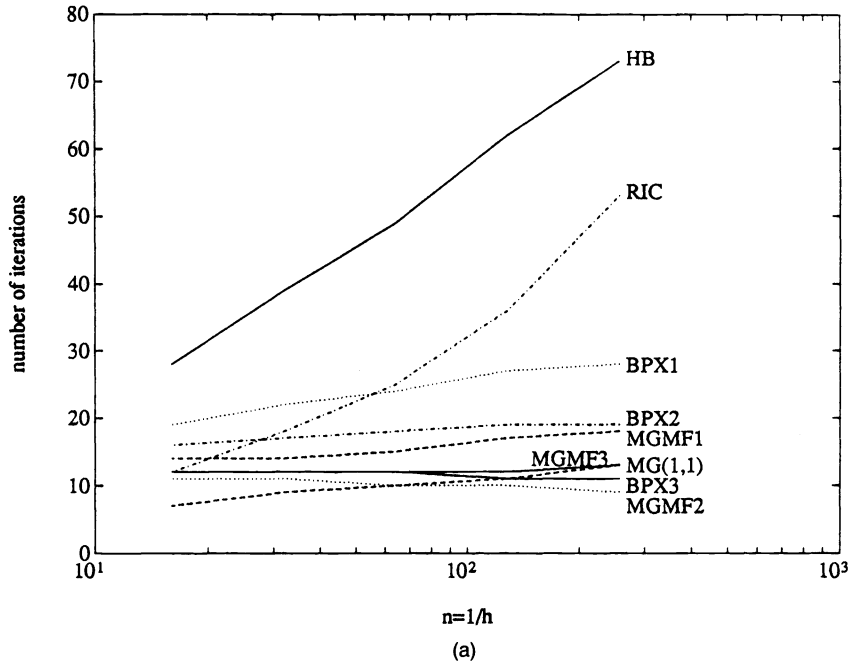
$$\frac{\partial}{\partial x}\left(e^{-xy}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(e^{xy}\frac{\partial u}{\partial y}\right) = f, \qquad \Omega = (0,1)^2, \tag{5.2}$$

(3) a two-dimensional problem with discontinuous coefficients with $f = 2x(1 - x) + 2y(1 - y)$,

$$\frac{\partial}{\partial x}\left(\rho(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\rho(x,y)\frac{\partial u}{\partial y}\right) = f, \qquad \Omega = (0,1)^2, \tag{5.3}$$

where

$$\rho(x,y) = \begin{cases} 10^4 & x > 0.5 \ y \leqq 0.5, \\ 10^{-4} & x \leqq 0.5 \ y > 0.5, \\ 1 & \text{elsewhere.} \end{cases}$$



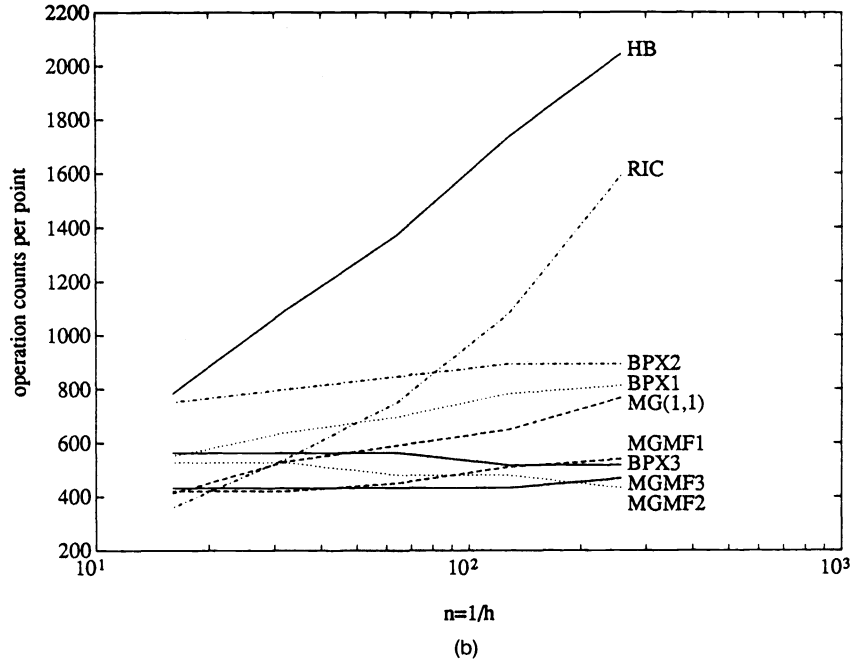


FIG. 5.1. (a) *Iteration and* (b) *operation counts for Test Problem* 1.

(4) the three-dimensional model problem with solution

$$u = x(1-x)y(1-y)z(1-z)e^{xyz},$$

$$\Delta u = f, \qquad \Omega = (0,1)^3, \tag{5.4}$$

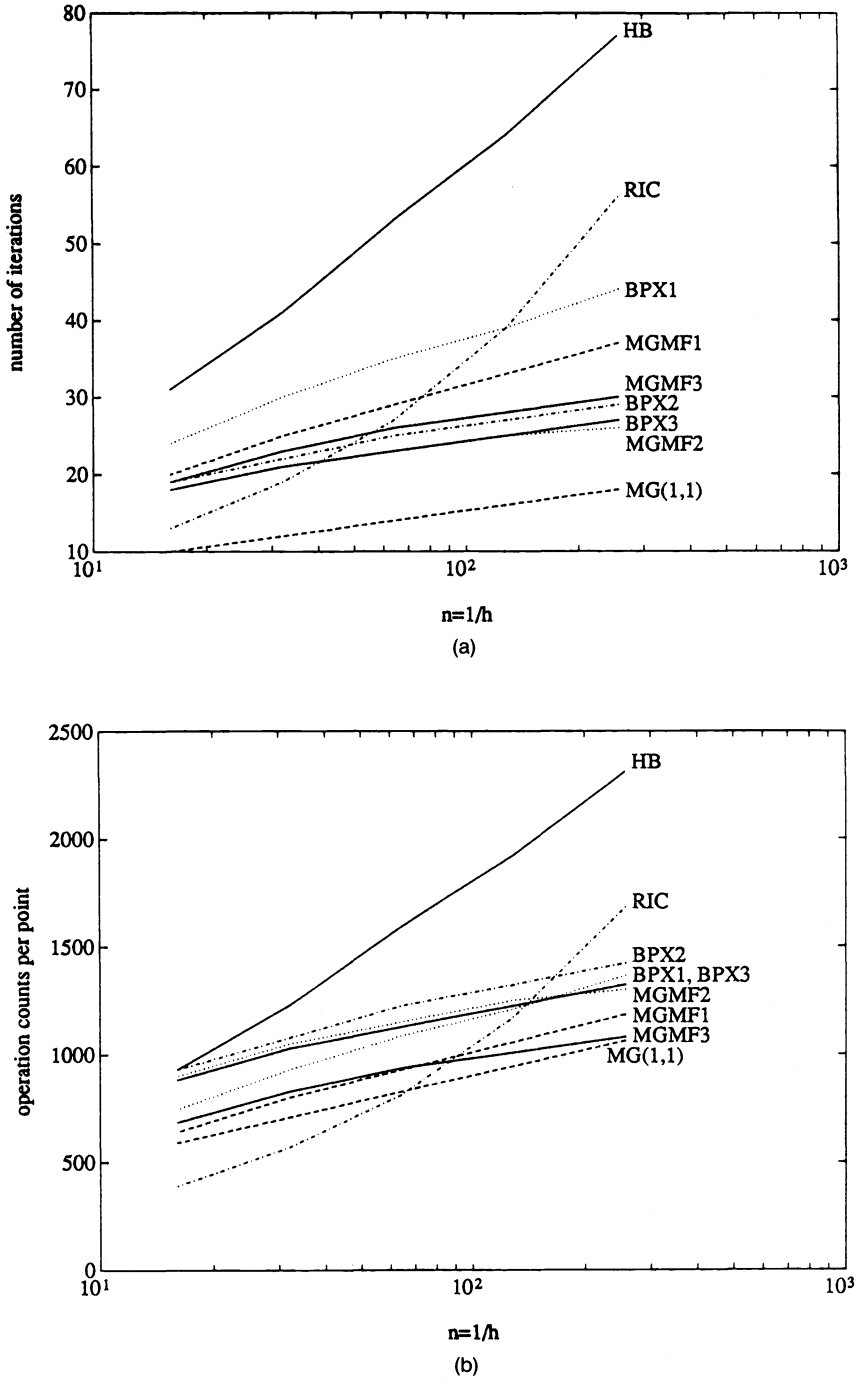


FIG. 5.2. (a) *Iteration and* (b) *operation counts for Test Problem* 2.

(5) a three-dimensional variable coefficient problem with solution $u = e^{xyz} \sin \pi x \times \sin \pi y \sin \pi z$,

$$\frac{\partial}{\partial x}\left(e^{-xyz}\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(e^{xyz}\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(e^{-xyz}\frac{\partial u}{\partial z}\right)=f, \qquad \Omega=(0,1)^3, \tag{5.5}$$

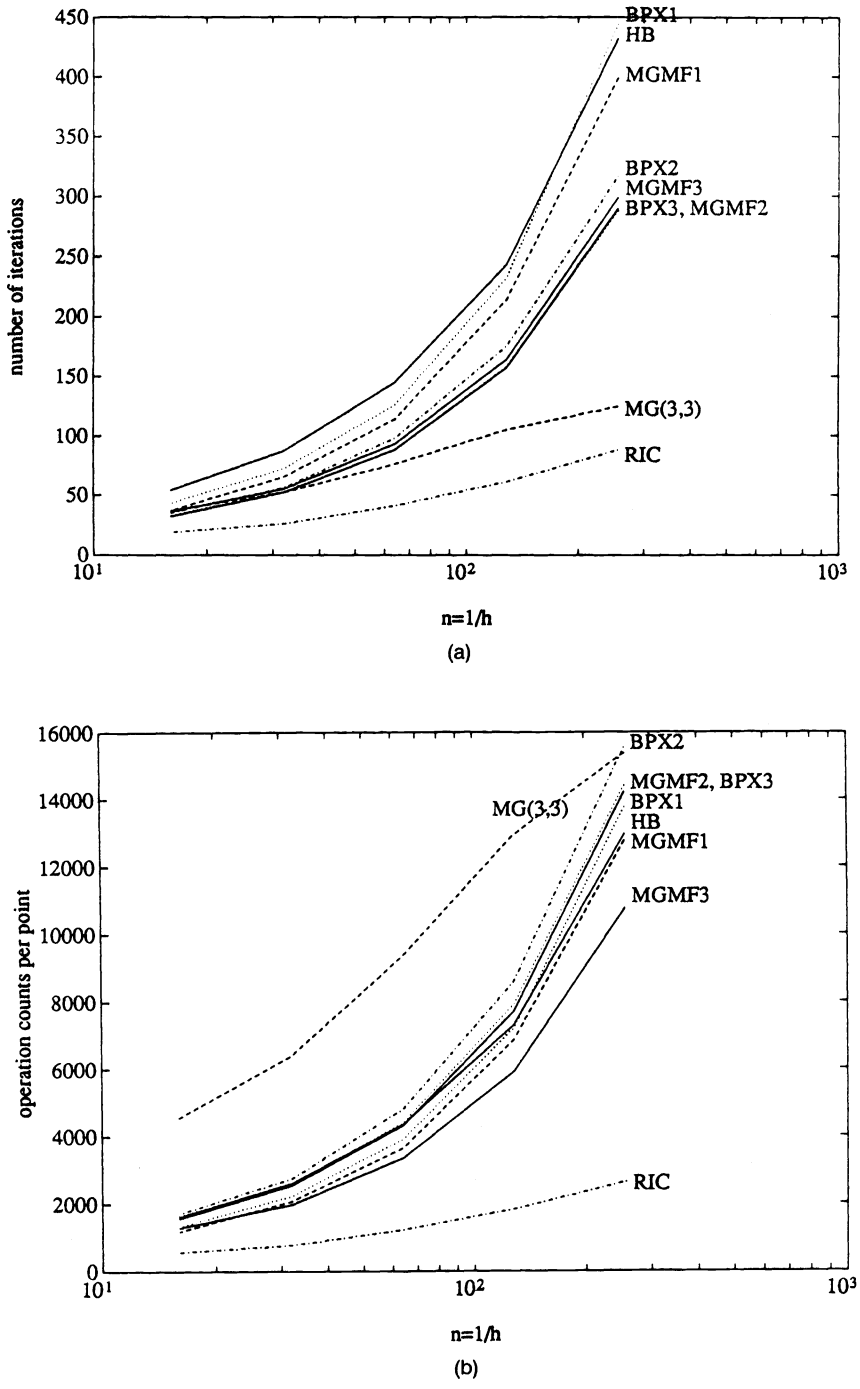


FIG. 5.3. (a) *Iteration and* (b) *operation counts for Test Problem* 3.

(6) a three-dimensional problem with discontinuous coefficients with $f = 2x(1-x) + 2y(1-y) + 2z(1-z)$,

$$\frac{\partial}{\partial x}\left(\rho(x,y,z)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\rho(x,y,z)\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(\rho(x,y,z)\frac{\partial u}{\partial z}\right)=f, \qquad \Omega=(0,1)^3, \tag{5.6}$$

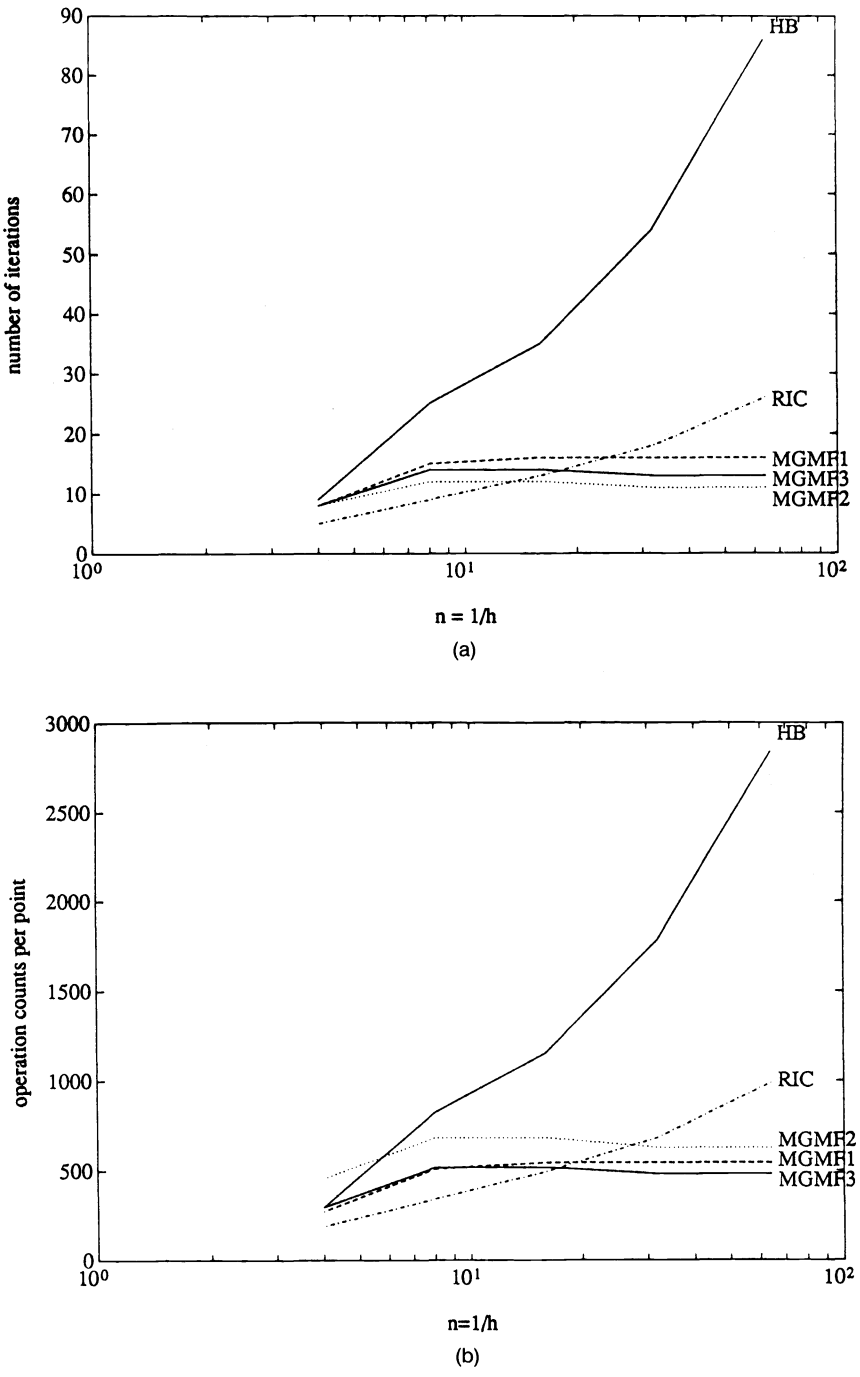


FIG. 5.4. (a) *Iteration and* (b) *operation counts for Test Problem* 4.

where

$$\rho(x,y,z)=\begin{cases} 10^{-4} & x>0.5 \text{ with } y\leqq 0.5, z\leqq 0.5 \text{ or } y>0.5, z>0.5, \\ 10^{4} & x\leqq 0.5 \text{ with } y>0.5, z\leqq 0.5 \text{ or } y\leqq 0.5, z>0.5, \\ 1 & \text{elsewhere.} \end{cases}$$

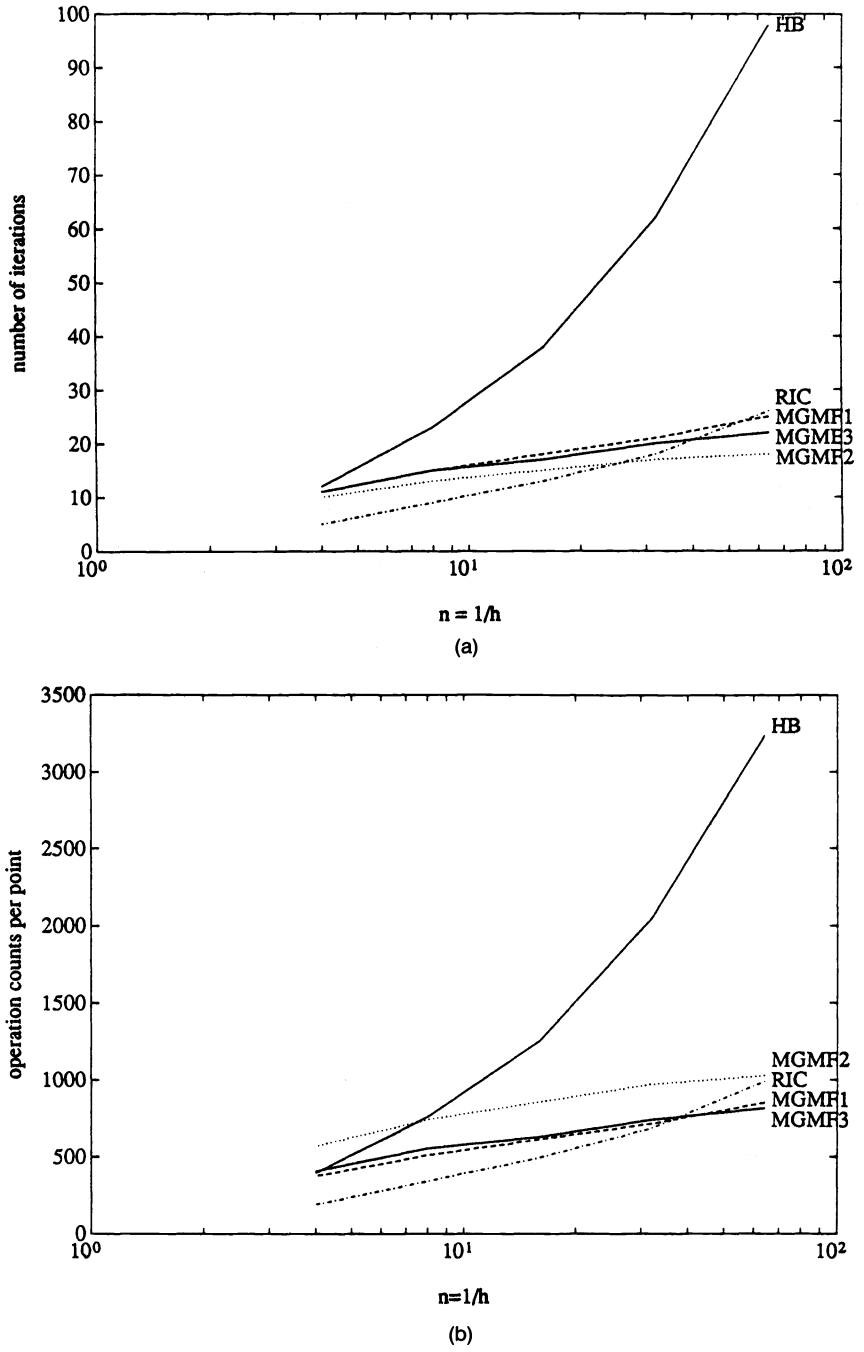


FIG. 5.5. (a) *Iteration and* (b) *operation counts for Test Problem* 5.

The number of iterations and operation counts per grid point are plotted in Figs. 5.1–5.6 (a) and (b), respectively. We can make the following observations from these figures.

(1) The BPX and MGMF preconditioners have better convergence behavior than the HB preconditioner, especially for three-dimensional problems. The HB method is competitive with the other multilevel methods only for the discontinuous coefficient problem in two dimensions.

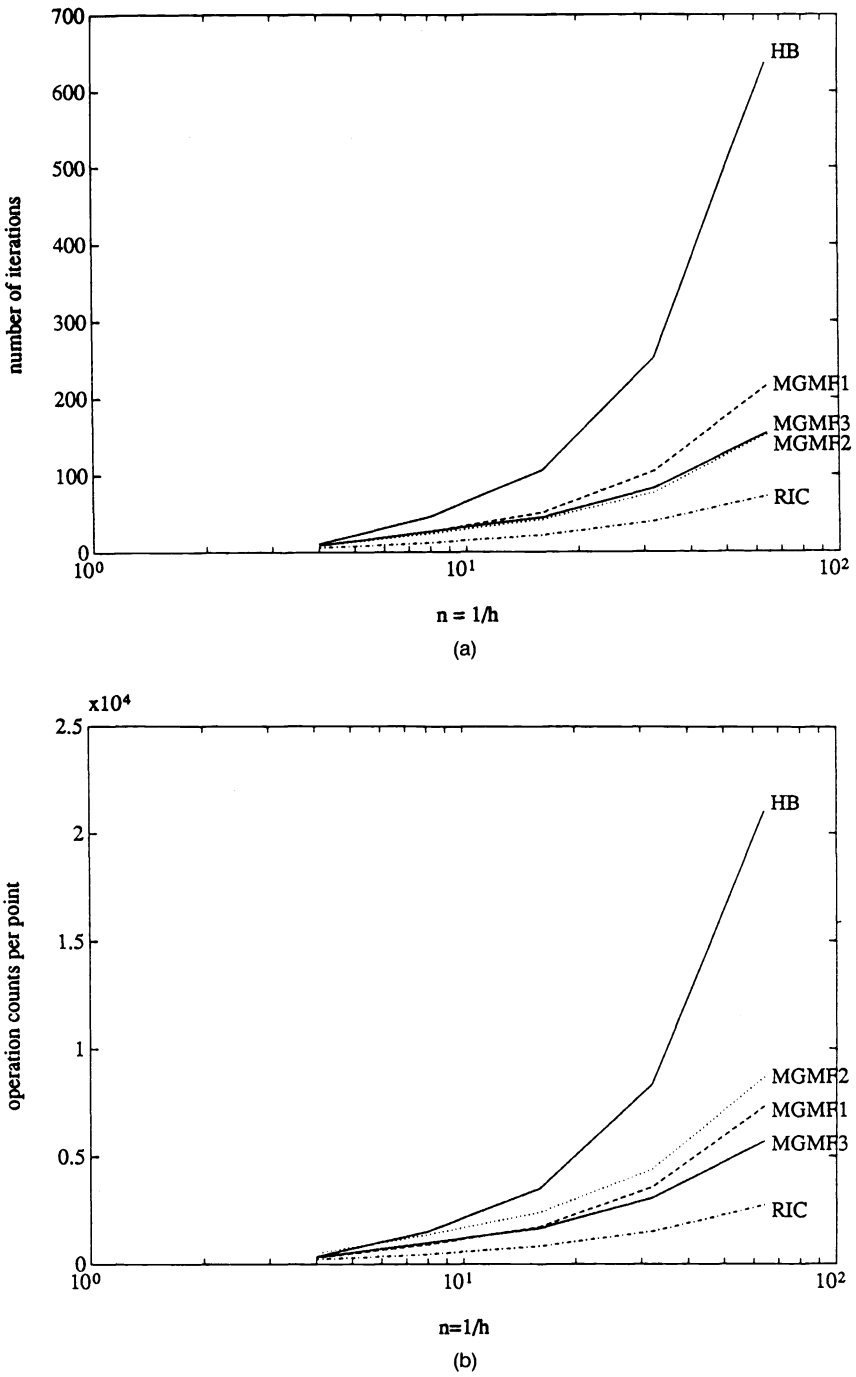


FIG. 5.6. (a) *Iteration and* (b) *operation counts for Test Problem* 6.

(2) The $O(\log^{\alpha} n)$ convergence rate for all the multilevel methods is evident, except for the three-dimensional HB method. The three-dimensional HB method behaves like $O(h^{-0.59})$ and $O(h^{-0.70})$ for problems (5.4) and (5.5), which are close to the predicted theoretical result $O(h^{-0.5})$. However, for the discontinuous coefficient problem (5.6), it converges more slowly, like $O(h^{-1.26})$.

(3) In general, the MGMF methods perform slightly better than the corresponding BPX methods. Recall that the only difference between the two methods is the choice of the elementary filters.

(4) Filtering twice (BPX2, BPX3, and MGMF2) does improve the convergence rates for the model Poisson problem in either two or three dimensions (the MGMF2 and BPX3 preconditioners appear to be spectrally equivalent.) For variable and discontinuous coefficient problems, filtering twice does not seem to improve the convergence rates enough to compensate for the extra work involved.

(5) The MGMF3 method is designed to incorporate the desired features of MGMF1 and MGMF2, i.e., the good convergence property due to filtering twice and the smaller amount of work due to filtering once at the finest grid level. It turns out that it works very well. MGMF3 behaves better than MGMF1 but worse than MGMF2 in the number of iterations required. However, in terms of amount of work, MGMF3 is better than MGMF1 and MGMF2.

(6) For small n (<100), the RIC method is competitive with all the multilevel methods. In fact, for the discontinuous coefficient problems, none of the multilevel preconditioners gives a better convergence rate than the RIC preconditioner. It appears that the RIC preconditioner captures the variation of the coefficients especially well. Its performance deteriorates as n gets large, as predicted by its inferior asymptotic convergence rate.

(7) The MG preconditioner is among the most efficient methods for problems with smooth coefficients. However, it has some difficulties with problems with discontinuous coefficients. In fact, for Problem (5.3), MG(1, 1) requires too many iterations to fit on the plot. Instead we show the results for the MG(3, 3) method, which converges in a reasonable number of iterations but still requires the most work of all the methods. We have noticed that the performance of the multigrid methods are somewhat sensitive to the initial guess. In experiments with random initial guesses, we have observed that the performance of the multigrid methods is significantly improved. This may be due to the extra smoothing operations in the multigrid methods which are more adept at annihilating the high frequency errors inherent in the random initial guess.

6. Conclusions. The experimental results show that the class of multilevel filtering preconditioners compares favorably with the hierarchical basis and the RIC preconditioners, at least for problems with smooth coefficients and quasi-uniform grids such as used in our experiments. For these types of problems, the multilevel filtering and the BPX methods behave quite similarly to the multigrid preconditioner. What these new methods offer is the saving of smoothing operations which are difficult to make effective for irregular problems, while preserving the nice asymptotic convergence rates of multigrid preconditioners. The relative performance of the hierarchical basis method should improve for irregular problems on highly nonuniform and refined meshes. Even though the RIC preconditioner shows better convergence rates for strongly discontinuous coefficient problems, it has a low degree of parallelism. The multilevel filtering preconditioners are also similar to the BPX method. What the filtering framework provides is the flexibility of filter design, which can lead to more efficient methods.

Acknowledgment. The authors thank the referees for their help in improving the presentation of this paper.

REFERENCES

[1] L. M. ADAMS AND M. E. G. ONG, *A comparison of preconditioners for* GMRES *on parallel computers*, in Parallel Computations and Their Impact on Mechanics, A. K. Noor, ed., pp. 171–186, The American Society of Mechanical Engineers, New York, 1987.

[2] S. F. ASHBY, *Polynomial preconditioning for conjugate gradient methods*, Ph.D. thesis, 1987, Department of Computer Science, University of Illinois, Urbana, IL 61801.
[3] O. AXELSSON, *A generalized* SSOR *method*, BIT, 13 (1972), pp. 443–467.
[4] ———, *An algebraic framework for multilevel methods*, Report 8820, Department of Mathematics, Catholic University, The Netherlands, 1988.
[5] O. AXELSSON AND G. LINDSKOG, *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math., 48 (1986), pp. 479–498.
[6] O. AXELSSON AND P. VASSILEVSKI, *Algebraic multilevel preconditioning methods*, I, Report 8811, Department of Mathematics, Catholic University, The Netherlands, 1988.
[7] ———, *Algebraic multilevel preconditioning methods*, II, Report 1988-15, Institute for Scientific Computation, University of Wyoming, Laramie, WY, 1988.
[8] R. E. BANK, T. F. DUPONT, AND H. YSERENTANT, *The hierarchical basis multigrid method*, Numer. Math., 52 (1988), pp. 427–458.
[9] J. H. BRAMBLE, J. E. PASCIAK, AND J. XU, *Parallel multilevel preconditioners*, Math. Comp., to appear.
[10] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.
[11] T. F. CHAN, *Fourier analysis of Relaxed Incomplete Cholesky factorization preconditioners*, CAM report 88-34, University of California, Los Angeles, CA, 1988.
[12] T. F. CHAN, C.-C. J. KUO, AND C. TONG, *Parallel elliptic preconditioners: Fourier analysis and performance on the Connection Machine*, Comput. Phys. Comm., 53 (1989), pp. 237–252.
[13] R. E. CROCHIERE AND L. R. RABINER, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[14] G. DAHLQUIST AND A. BJÖRCK, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[15] T. DUPONT, R. P. KENDALL, AND H. H. RACHFORD, JR., *An approximate factorization procedure for solving self-adjoint difference equations*, SIAM J. Numer. Anal., 5 (1968), pp. 559–573.
[16] A. GREENBAUM, C. LI, AND H. Z. CHAO, *Parallelizing preconditioned conjugate gradient algorithms*, Comput. Phys. Comm. 53 (1989), pp. 295–309.
[17] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, New York, 1985.
[18] O. HERRMANN, *On the approximation problem in nonrecursive digital filter design*, IEEE Trans. Circuit Theory, CT-18 (1971), pp. 411–413.
[19] O. G. JOHNSON, C. A. MICCHELLI, AND G. PAUL, *Polynomial preconditioning for conjugate gradient calculations*, SIAM J. Numer. Anal., 20 (1983), pp. 362–376.
[20] R. KETTLER, *Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., pp. 502–534, Springer-Verlag, Berlin, New York, 1982.
[21] R. KETTLER AND J. A. MEIJERINK, *A multigrid method and a combined multigrid-conjugate gradient method for elliptic problems with strongly discontinuous coefficients in general domain*, Shell publication 604, KSEPL, Rijswijk, the Netherlands.
[22] C.-C. J. KUO AND B. C. LEVY, *Two-color Fourier analysis of the multigrid method with red/black Gauss-Seidel smoothing*, Appl. Math. Comput., 29 (1989), pp. 69–87.
[23] C.-C. J. KUO AND T. F. CHAN, *Two-color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 000–000.
[24] Y. A. KUZNETSOV, *Multigrid domain decomposition methods for elliptic problems*, in Proceedings VIII International Conference on Computational Methods for Applied Science and Engineering, Vol. 2, pp. 605–616, 1987.
[25] 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.
[26] M. E. G. ONG, *The* 3D *linear hierarchical basis preconditioner and its shared memory parallel implementation*, Preprint, Department of Applied Mathematics, University of Washington, Seattle, WA, 1988.
[27] P. VASSILEVSKI, *Iterative methods for solving finite element equations based on multilevel splitting of the matrix*, Preprint, Bulgarian Academy of Science, Sofia, Bulgaria, 1987.
[28] J. XU, *Theory of multilevel methods*, Ph.D. thesis, Department of Mathematics, Cornell University, Ithaca, NY, 14853, 1989.
[29] H. YSERENTANT, *On the multi-level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 430–452, July 1990

007

PARTITIONING SPARSE MATRICES WITH EIGENVECTORS OF GRAPHS*

ALEX POTHEN†, HORST D. SIMON‡, AND KANG-PU LIOU§

Abstract. The problem of computing a small vertex separator in a graph arises in the context of computing a good ordering for the parallel factorization of sparse, symmetric matrices. An algebraic approach for computing vertex separators is considered in this paper. It is shown that lower bounds on separator sizes can be obtained in terms of the eigenvalues of the Laplacian matrix associated with a graph. The Laplacian eigenvectors of grid graphs can be computed from Kronecker products involving the eigenvectors of path graphs, and these eigenvectors can be used to compute good separators in grid graphs. A heuristic algorithm is designed to compute a vertex separator in a general graph by first computing an edge separator in the graph from an eigenvector of the Laplacian matrix, and then using a maximum matching in a subgraph to compute the vertex separator. Results on the quality of the separators computed by the spectral algorithm are presented, and these are compared with separators obtained from other algorithms for computing separators. Finally, the time required to compute the Laplacian eigenvector is reported, and the accuracy with which the eigenvector must be computed to obtain good separators is considered. The spectral algorithm has the advantage that it can be implemented on a medium-size multiprocessor in a straightforward manner.

Key words. graph partitioning, graph spectra, Laplacian matrix, ordering algorithms, parallel orderings, sparse matrix, vertex separator

AMS(MOS) subject classifications. 65F50, 65F05, 65F15, 68R10

1. Introduction.

In the solution of large, sparse, positive definite systems on parallel computers, it is necessary to compute an ordering of the matrix such that it can be factored efficiently in parallel. Several algorithms have been developed recently for computing good parallel orderings: for instance [39], [40]. For large problems, the storage required for the structure of the matrix may exceed the storage capacities of a single processor, and the ordering itself will need to be computed in parallel. One strategy to compute a good parallel ordering is to employ the divide-and-conquer paradigm: Find a set of vertices in the adjacency graph of the matrix, whose removal disconnects the graph into two nearly equal parts. Number the vertices in the separator last, and recursively number the vertices in the two parts by the same strategy. This strategy is employed in several algorithms which order sparse matrices for factorization; e.g., the Sparspak nested dissection algorithm [27].

In computing an ordering by the above approach, at each step, the following *partitioning problem* needs to be solved: Given an adjacency graph G of a sparse matrix, find a vertex separator S such that S has few vertices and S disconnects $G \setminus S$ into two parts A, B with nearly equal numbers of vertices. In the context of the ordering problem, since a separator S becomes a clique in the factor matrix (filled matrix), a small S controls the fill incurred by the ordering. The requirement that the parts A and B be

* Received by the editors August 15, 1989; accepted for publication (in revised form) December 27, 1989.

† Computer Science Department, Pennsylvania State University, Whitmore Lab, University Park, Pennsylvania 16802 (pothen@shire.sys.cs.psu.edu, na.pothen@na-net.stanford.edu.). A part of this work was done while the author was at the University of Wisconsin, Madison. The research of this author was supported by National Science Foundation grant CCR-8701723, National Science Foundation Equipment grant CCR-8705110, and Air Force Office of Scientific Research grant AFOSR-88-0161.

‡ Numerical Aerodynamic Simulation (NAS) Systems Division, Mail-Stop 258-5, NASA Ames Research Center, Moffett Field, California 94035 (simon@orville.nas.nasa.gov.). The work of this author was supported through National Aeronautics and Space Administration contract NAS2-12961.

§ Computer Science Department, Pennsylvania State University, Whitmore Lab, University Park, Pennsylvania 16802.

roughly equal is a simple way of maintaining load balance in parallel computation, since the submatrix represented by each part will be mapped to a subset of half the processors.

In this paper, we consider a spectral algorithm for solving the partitioning problem. We associate with the given sparse, symmetric matrix (and its adjacency graph), a matrix called the Laplacian matrix. We compute a particular eigenvector of the Laplacian matrix and use its components to initially partition the vertices into two sets A', B'. The set of edges joining A' and B' is an edge separator in the graph G. A vertex separator S is computed from the edge separator by a matching technique.

The use of spectral methods to compute edge separators in graphs was first considered by Donath and Hoffman [16], [17], and since then spectral methods for computing various graph parameters have been considered by several others. A discussion of some of this work is included in § 2.

The spectral algorithm for computing vertex separators considered in this paper has three features that distinguish it from previous algorithms that are worthy of comment.

First, previous algorithms for computing separators, such as the level-structure separator algorithm in Sparspak or the Kernighan–Lin algorithm make use of *local* information in the graph, viz. information about the neighbors of a vertex, to compute separators. The spectral method employs *global* information about the graph, since it computes a separator from eigenvector components. Thus the spectral method has the potential of finding separators in the graph that are qualitatively different from the separators obtained by previous approaches.

Second, we can view the spectral method as an approach in which a vertex in the graph makes a *continuous* choice, with a weight between $+1$ and -1, about which part in the initial partition it is going to belong to. All vertices with weights below the median weight form one part, and the rest, the other part. In the Kernighan–Lin method, each vertex makes a discrete choice (zero or one) to belong to one set. The weights in the spectral method can be used to move a few vertices from one part to the other, if a slightly different partition is desired in the course of the separator algorithm.

Third, the dominant computation in the spectral method is an eigenvector computation by a Lanczos or similar algorithm. This distinguishes the new algorithm from standard graph-theoretical algorithms computationally. Most of the computation is based on standard vector operations on floating point numbers. Because of its algebraic nature, the algorithm is parallelizable in a fairly straightforward manner on medium-grain multiprocessors used in scientific computing. Furthermore, since most of the computations are also vector floating point operations, this algorithm is well suited for vector supercomputers used for large scale scientific computing.

This paper is organized as follows. We include background material on the spectral properties of Laplacian matrices and their relevance to graph partitioning in § 2. We also review earlier work on computing edge separators from the eigenvectors of the adjacency matrix in this section. In § 3, we obtain lower bounds on the size of the smallest vertex separators of a graph in terms of the eigenvalues of the Laplacian matrix. Two different techniques for proving lower bounds are illustrated: One uses the Courant–Fischer–Poincaré minimax criterion, and the second employs an inequality from the proof of the Wielandt–Hoffman theorem. We then show that the spectra of rectangular and square grid graphs can be computed explicitly from the spectra of path graphs by employing suitable graph products and Kronecker products in § 4. We proceed to show how good edge and vertex separators in the grid graphs can be computed from the spectral information. In § 5, we describe our heuristic spectral algorithm to compute vertex separators in general graphs. The algorithm initially computes an edge separator, and then uses a maximum matching in a subgraph to compute the vertex separator. Results about the

quality of the separators computed by the algorithm are presented in § 6. In this section, we also compare the spectral separators with separators computed in the first step of the Sparspak nested dissection ordering algorithm and the Kernighan–Lin algorithm, as well as with results obtained recently by Liu [42] and Leiserson and Lewis [38]. The time required to compute the Laplacian eigenvectors with the Lanczos algorithm and the accuracy needed in the eigenvector to obtain good separators are addressed in § 7. The final § contains our conclusions and some directions for future work.

2. Background. Let $G = (V, E)$ be an undirected graph on $|V| = n$ vertices. The $n \times n$ adjacency matrix $A = A(G)$ has element $a_{v,w}$ equal to one if $(v, w) \in E$, and zero otherwise. By convention, $a_{v,v}$ is zero for all $v \in V$. The rows and columns of the matrices associated with a graph are indexed by the vertices of the graph, their order being arbitrary. Let $d(v)$ denote the degree of a matrix, and define D to be the $n \times n$ diagonal matrix with $d_{v,v} = d(v)$. The matrix $Q = Q(G) = D - A$ is the *Laplacian matrix* of G.

Let the edges of the graph G be directed arbitrarily, and let C denote the vertex-edge incidence matrix of the directed graph. The $|V| \times |E|$ matrix C has elements

$$c_{v,e} = \begin{cases} +1 & \text{if } v \text{ is the head of } e, \\ -1 & \text{if } v \text{ is the tail of } e, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to verify that $Q(G) = CC^t$, and that Q is independent of the direction of the edges in C. Biggs [11] contains a good discussion of the techniques from algebraic graph theory that are used here.

The spectral properties of Q have been studied by several authors [4], [23]. Since

$$\underline{x}^t Q \underline{x} = \underline{x}^t C C^t \underline{x} = (C^t \underline{x})^t (C^t \underline{x}) = \sum_{(v,w) \in E} (x_v - x_w)^2,$$

Q is positive semidefinite. Let the eigenvalues of Q be ordered $\lambda_1 = 0 \leqq \lambda_2 \leqq \cdots \leqq \lambda_n$. An eigenvector corresponding to λ_1 is $\underline{e}$, the vector of all ones. The multiplicity of the zero eigenvalue is equal to the number of connected components of the graph. If G is connected, then the second smallest eigenvalue λ_2 is positive. We call an eigenvector $\underline{y}$ corresponding to λ_2 a *second eigenvector*.

Fiedler [23], [24] has studied the properties of the second eigenvalue λ_2 and a corresponding eigenvector $\underline{y}$. He calls λ_2 the *algebraic connectivity*, and relates it to the vertex and edge connectivities of a graph. He has also investigated the partitions of G generated by the components of the eigenvector $\underline{y}$. One of his results of interest in this paper can be rephrased as follows.

THEOREM 2.1. *Let G be a connected graph, and let $\underline{y}$ be an eigenvector corresponding to λ_2. For a real number $r \geqq 0$, define $V_1(r) = \{v \in V : y_v \geqq -r\}$. Then the subgraph induced by $V_1(r)$ is connected. Similarly, for a real number $r \leqq 0$, the subgraph induced by the set $V_2(r) = \{v \in V : y_v \leqq |r|\}$ is also connected.*

In both sets V_1 and V_2, it is necessary to include all vertices with zero components for the theorem to hold. The role played by these latter vertices in the connectedness of the two subgraphs has been investigated at greater length by Powers [54], [55].

A corollary to this result is that if $y_v \neq 0$ for all $v \in V$, then each of the subgraphs induced by $P = \{v \in V : y_v > 0\}$ and $N = \{v \in V : y_v < 0\}$ is a connected subgraph of G.

The eigenvectors of the adjacency matrix corresponding to its algebraically largest eigenvalues have also been used to partition graphs. It is of interest to ask if a similar theorem holds for an eigenvector corresponding to the second largest eigenvalue of the adjacency matrix.

Let $\underline{x}$, $\underline{y}$ denote eigenvectors corresponding to the algebraically largest and second largest eigenvalues, respectively, of the adjacency matrix of G. By the Perron–Frobenius theory, it is known that all components of $\underline{x}$ are positive. Fiedler's theorem states that if α is a nonnegative number, then the subgraph induced by

$$V_1 = \{ v \in V : y_v + \alpha x_v \geqq 0 \}$$

is connected. Similarly, if α is a nonpositive number, then the subgraph induced by $V_2 = \{ v \in V : y_v - |\alpha| x_v \leqq 0 \}$ is also connected.

Alon [1] and Mohar [44] have studied the relationship of the second Laplacian eigenvalue to the *isoperimetric number*, $i(G)$. If U is a subset of the vertices of the graph G, and δU denotes the set of edges with one endpoint in U and the other in $V \setminus U$, then

$$i(G) = \min_{|U| \leqq n/2} \frac{|\delta U|}{|U|}.$$

Clearly $i(G)$ is related to the problem of computing good edge separators.

Alon, Galil, and Milman [2], [3] have related the second Laplacian eigenvalue to the expansion properties of graphs. The relationship of the Laplacian spectrum to several other graph properties has been considered by several authors; two recent survey articles are by Mohar [45] and Bien [10].

Spectral methods for computing edge separators have been considered by several researchers: Donath and Hoffman [16], [17], Barnes [7], [8], Barnes and Hoffman [9], Boppana [12]. An algorithm for coloring a graph by employing the eigenvectors of the adjacency matrix has been considered by Aspvall and Gilbert [6] and a spectral algorithm for finding a pseudoperipheral node has been described by Grimes, Pierce, and Simon [33]. A spectral algorithm for envelope reduction is considered in [53].

Algorithms that make use of flows in networks to compute separators have been designed by Bui et al. [13], and Leighton and Rao [37]. The former describes a bisection algorithm with good average-case behavior for degree-regular random graphs, and the latter describes an approximation algorithm for minimum quotient edge separators.

3. Lower bounds. We obtain lower bounds on the sizes of vertex separators in terms of the eigenvalues of the Laplacian matrix $Q(G)$ in this section. The lower bounds hold for *any* vertex separator in the graph; in particular, these bounds apply to a smallest separator in the graph. We assume that the graph G is connected.

Let $G = (V, E)$ denote a graph on $|V| = n$ vertices, and let A be a subset of its vertices. Denote by $\rho(v, A)$ the distance of a vertex v from A, i.e., the fewest number of edges in a shortest path from v to a vertex in A. Let S denote the set of vertices which are at a distance of less than $\rho \geqq 2$ from A, and not belonging to A. Hence

$$S = \{ v \in V \setminus A : \rho(v, A) < \rho \}.$$

Define $B = V \setminus (A \cup S)$; if $B \neq \varnothing$, then the distance between A and B, $\rho(A, B) = \rho$. If $\rho > 2$, the set S is a *wide separator* that separates A from B. If $\rho = 2$, we get the commonly used notion of separators. Wide separators were first used in sparse matrix algorithms by Gilbert and Schreiber [28].

Let E_A denote the set of edges with both endpoints in A, and E_{AS} denote the set of edges with one endpoint in A, and the other in S. The sets E_B, E_S, and E_{BS} are defined similarly. In the following, it will be convenient to work with the fractional sizes $a \equiv |A|/n$, $b \equiv |B|/n$, and $s \equiv |S|/n$. The degree of a vertex v will be denoted by $d(v)$, and Δ will denote the maximum degree of vertices in G.

The first result is a lower bound on the size of a wide separator separating any pair of vertex disjoint sets A and B that are at a distance ρ from each other. As will be described later, it generalizes a result of Alon, Galil, and Milman [2].

THEOREM 3.1. *Let A, B be disjoint subsets of vertices of G that are at a distance $\rho \geqq 2$ from each other. Let S denote the set of vertices not belonging to A that are at a distance less than ρ from A. Then*

$$s^2 + \beta s - \rho^2 a(1-a) \geqq 0, \quad \text{where } \beta = (\Delta/\lambda_2) + \rho^2 a - 1.$$

Proof. Let $\underline{e}$, $\underline{0}$ be the vector of all ones and all zeros, respectively. The Courant–Fischer–Poincaré minimax principle states that

$$\lambda_2 = \min_{\substack{\underline{x} \neq \underline{0} \\ \underline{e}^t \underline{x} = 0}} \frac{\underline{x}^t Q \underline{x}}{\underline{x}^t \underline{x}} = \min_{\substack{\underline{x} \neq \underline{0} \\ \underline{e}^t \underline{x} = 0}} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_{i=1}^{n} x_i^2}.$$

Using the Lagrange identity in the above equation, Fiedler [24] derived the following inequality, which is valid for all real n-vectors.

$$n \sum_{(i,j) \in E} (x_i - x_j)^2 \geqq \lambda_2 \sum_{\substack{i,j \in V \\ i<j}} (x_i - x_j)^2. \tag{1}$$

We prove the result by making an appropriate choice of $\underline{x}$ in the above inequality.

Choose the vth component of $\underline{x}$ to be $x_v = 1 - (2/\rho) \min \{\rho, \rho(v, A)\}$. If $v \in A$, then $x_v = 1$; if $v \in B$, then $x_v = -1$; and if $v \in S$, then $-1 + (2/\rho) \leqq x_v \leqq 1 - (2/\rho)$. Also, if v, w are adjacent vertices, then $|x_v - x_w| \leqq 2/\rho$.

The left-hand side of (1) has nonzero contributions from three terms, and it can be bounded from above as follows:

$$\begin{aligned} \sum_{(i,j) \in E} (x_i - x_j)^2 &= \left(\sum_{\substack{(i,j) \in E \\ i \in A, j \in S}} + \sum_{\substack{(i,j) \in E \\ i \in B, j \in S}} + \sum_{\substack{(i,j) \in E \\ i \in S, j \in S}} \right) (x_i - x_j)^2 \\ &\leqq \frac{4}{\rho^2} (|E_{AS}| + |E_{BS}| + |E_S|) \qquad (2) \\ &\leqq \frac{4}{\rho^2} n s \Delta. \end{aligned}$$

Similarly, nonzero contributions to the right-hand side of (1) also come from three terms, and we obtain a lower bound as shown:

$$\begin{aligned} \sum_{\substack{i,j \in V \\ i<j}} (x_i - x_j)^2 &= \left(\sum_{i \in A, j \in S} + \sum_{i \in A, j \in B} + \sum_{i \in B, j \in S} + \sum_{\substack{i \in S, j \in S \\ i<j}} \right) (x_i - x_j)^2 \\ &\geqq \left(\sum_{i \in A, j \in S} + \sum_{i \in A, j \in B} + \sum_{i \in B, j \in S} \right) (x_i - x_j)^2 \\ &\geqq \left(1 - \left(1 - \frac{2}{\rho}\right)\right)^2 n^2 as + (1 - (-1))^2 n^2 ab + \left(-1 - \left(-1 + \frac{2}{\rho}\right)\right)^2 n^2 bs \qquad (3) \\ &= \frac{4n^2}{\rho^2} ((a+b)s + \rho^2 a(1-a-s)) \\ &= \frac{4n^2}{\rho^2} ((1-s)s + \rho^2 a(1-a-s)). \end{aligned}$$

Using inequalities (2) and (3) in Fiedler's inequality (1), and canceling common terms, we obtain

$$s\Delta \geqq \lambda_2((1-s)s+\rho^2 a(1-a-s)).$$

This last inequality yields the desired result after some rearrangement. □

Fiedler [23] showed that $\lambda_2 \leqq (n/(n-1)) \min\{d(v): v \in V\}$. Mohar [44] proved that for all graphs except the complete graphs K_n, $\Delta \geqq \lambda_2$. Thus for all graphs except the complete graphs, the ratio $\Delta/\lambda_2 \geqq 1$, and β is a positive number. Indeed, the ratio Δ/λ_2, and hence β, is much larger than one, for all the adjacency graphs of sparse matrices that we have computed partitions.

COROLLARY 3.2. *If* $\beta \geqq \rho$, *then*

$$s \geqq \frac{\rho^2 a(1-a)}{\beta} = \frac{\rho^2 a(1-a)}{(\Delta/\lambda_2)+\rho^2 a-1}.$$

Proof. Let s_1, s_2 be the roots of the quadratic equation corresponding to the inequality in Theorem 3.1, with $s_1 \leqq s_2$. Then $s \geqq s_2$, and

$$s_2 = \frac{1}{2}(-\beta+(\beta^2+4\rho^2 a(1-a))^{1/2}).$$

If $\beta \geqq 2\rho(a(1-a))^{1/2}$, then expanding the right-hand side in power series yields the result.

It remains to verify the condition of the corollary. Since $(a(1-a))^{1/2}$ has its maximum value $\frac{1}{2}$ when $0 \leqq a \leqq 1$, the power series expansion is valid when $\beta \geqq \rho$. □

The corollary exhibits the dependence of vertex separator sizes on λ_2: the smaller the second eigenvalue, the larger the ratio Δ/λ_2, and the smaller the lower bound on the vertex separator size. The corollary also shows the dependence of the lower bound on the distance ρ and the fractional size of the set A.

The common situation of a separator corresponds to $\rho = 2$. In this case, the quadratic inequality becomes $s^2 + \beta s - 4a(1-a) \geqq 0$, with $\beta = (\Delta/\lambda_2) + 4a - 1$. After some simplification, it can be seen that the inequality in Theorem 2.1 of Alon, Galil, and Milman [2] is equivalent to the above inequality. In this case, when $\beta \geqq 2$, we obtain the lower bound

$$s \geqq \frac{4a(1-a)}{(\Delta/\lambda_2)+4a-1}.$$

Mohar [43, Lem. 2.4] has obtained a lower bound on vertex separators in terms of λ_n and λ_2. Lower bounds on edge separators can also be obtained by this technique.

A second lower bound. We now obtain a lower bound that exhibits another factor influencing the size of vertex separators. The technique used is derived from the Wielandt–Hoffman theorem, and has been previously used by Donath and Hoffman [17] to obtain lower bounds on edge separators.

Let S be a vertex separator that separates the graph G into two sets A and B, with $|A| \geqq |B| \geqq |S|$. Let $d(v)$ denote the degree of a vertex v, and let $i(v)$ denote the "internal" degree of v, i.e., the number of edges incident on v with the other endpoint in the same set as v.

Recall that the eigenvalues of Q are ordered as $\lambda_1 = 0 < \lambda_2 \leqq \lambda_3 \cdots \leqq \lambda_n$. Let the $n \times n$ matrix $J = \operatorname{diag}(J_a, J_b, J_c)$, where J_a is the $na \times na$ matrix of all ones, and J_b, J_c are similarly defined. The eigenvalues of J are $\mu_1 = na \geqq \mu_2 = nb \geqq \mu_3 = ns > \mu_4 = \cdots = \mu_n = 0$.

THEOREM 3.3. *Let S be a vertex separator that divides a graph G into two parts, A, B, with $|A| \geq |B| \geq |S|$. Then*

$$s \geq \frac{(1-a)\lambda_2}{2\Delta - (\lambda_3 - \lambda_2)}.$$

Proof. From the proof of the Wielandt–Hoffman theorem [34] (see also [17]),

$$\text{trace}\,(QJ) \geq \sum_{i=1}^{n} \lambda_i \mu_i. \tag{4}$$

We now compute both sides of the above inequality.

The right-hand side is

$$\sum_{i=1}^{n} \lambda_i \mu_i = na \cdot 0 + nb \cdot \lambda_2 + ns \cdot \lambda_3 = n(1-a-s)\lambda_2 + ns\lambda_3.$$

To evaluate the left-hand side, we partition the symmetric matrix Q to conform to J:

$$Q = \begin{pmatrix} Q_{aa} & 0 & Q_{as} \\ 0 & Q_{bb} & Q_{bs} \\ Q_{as}^t & Q_{bs}^t & Q_{ss} \end{pmatrix}.$$

$$\begin{aligned} \text{trace}\,(QJ) &= \text{trace}\,(Q_{aa}J_a) + \text{trace}\,(Q_{bb}J_b) + \text{trace}\,(Q_{ss}J_s) \\ &= \left(\sum_{v \in A} + \sum_{v \in B} + \sum_{v \in S} \right) d(v) - i(v) \\ &= 2(|E| - |E_A| - |E_B| - |E_S|) \qquad (5) \\ &\leq 2(|E| - |E_A| - |E_B|) \\ &\leq 2ns\Delta. \end{aligned}$$

Substituting the inequalities (3) and (5) in (4), we obtain

$$2ns\Delta \geq n(1-a-s)\lambda_2 + ns\lambda_3.$$

This yields the final result after some rearrangement. □

This last lower bound on a vertex separator size shows as before that the magnitude of λ_2 influences the lower bound; it also shows that the "gap" between λ_3 and λ_2 has an effect.

A word of caution is in order about these lower bounds. These bounds should be considered the same way one treats an upper bound on the error in an a priori roundoff error analysis [58]. The lower bounds obtained are not likely to be tight, except for particular classes of graphs. They do illustrate, however, that a large λ_2, with an accompanying small Δ/λ_2, will result in large sizes for the best separators in a graph.

4. Partitions of grid graphs. In this section we show that the second eigenvector of the Laplacian matrix can be used to find good vertex separators in grid graphs, which are model problems in sparse matrix computations. The separators obtained are identical to the separators used by George [26] at the first step in a nested dissection ordering of grid graphs.

To compute separators by this technique, we need to first compute the eigenvectors of grids. The Laplacian spectra of grid graphs can be explicitly computed in terms of the

Laplacian spectra of path graphs. Some of this material is well known in spectral graph theory [15], but such treatments consider only eigenvalues and not eigenvectors. Further, the nine-point grid needs to be modified before its spectrum can be explicitly computed. The techniques used are quite general, and can be used to compute the spectra of several other classes of graphs which can be expressed in terms of graph products of simpler graphs.

The path graph. Let P_n denote the path graph on n vertices. We assume in the following discussion that $n \geqq 2$ is even. We number the vertices of the path from 1 to n in the natural order from left to right.

The Laplacian matrix of P_n is tridiagonal, and hence its spectrum is easily computed. Let $\phi_n \equiv \pi/n$. We denote the elements of a vector $\underline{x}$ by writing its ith component as (x_i).

LEMMA 4.1. *The Laplacian spectrum of P_n is*

$$\lambda_{k,n} = 4 \sin^2\left(\frac{1}{2}(k-1)\phi_n\right),$$

$$\underline{x}_{k,n} = (\cos((i-1/2)(k-1)\phi_n)), \quad \text{for } k = 1, \cdots, n,\ i = 1, \cdots, n. \quad \square$$

As k ranges from 1 to n, the angle $\frac{1}{2}(k-1)\phi_n$ varies from zero to $\pi/2$; hence the eigenvalues are ordered as $\lambda_{1,n} \leqq \lambda_{2,n} \cdots \leqq \lambda_{n,n}$. Note that $\lambda_{1,n} = 0$, $\underline{x}_{1,n} = \underline{1}$, and $\lambda_{2,n} = 4 \sin^2(\phi_n/2)$, and $\underline{x}_{2,n} = (\cos((i - \frac{1}{2})\phi_n))$. The components of $\underline{x}_{2,n}$ plotted against the vertices of P_{30} decrease monotonically from left to right.

Let x_l denote the median ($n/2$th largest) component of the second eigenvector, and partition the vertices of the path into two sets, one set consisting of all vertices with components less than or equal to the median component, and the other consisting of all vertices with components larger than the median component. This partitions the path into subsets of vertices of equal size, one consisting of the vertices with positive eigenvector components, and the other consisting of vertices with negative components.

Graph products. We can compute the spectra of grid graphs from the spectra of the path graph. We require the concepts of graph products and the Kronecker products of matrices. One notation for graph products is from Cvetkovic, Doob, and Sachs [15], and a good discussion of Kronecker products may be found in Fiedler [25].

For $i = 1, 2$, let $G_i = (V_i, E_i)$ be graphs. The *Cartesian sum* $G_1 + G_2$ is the graph $(V_1 \times V_2, E)$, where vertices (i_1, j_1) and (i_2, j_2) are joined by an edge if either $i_1 = i_2$ and $\{j_1, j_2\}$ is an edge in G_2, or $j_1 = j_2$ and $\{i_1, i_2\}$ is an edge in G_1. The *Cartesian product* $G_1 \cdot G_2$ is the graph $(V_1 \times V_2, F)$, where vertices (i_1, j_1) and (i_2, j_2) are joined by an edge if $\{i_1, i_2\}$ is an edge in G_1 and $\{j_1, j_2\}$ is an edge in G_2. The *strong sum* $G_1 \oplus G_2$ is the graph $(V_1 \times V_2, E \cup F)$; thus it contains the edges in both the Cartesian sum and the Cartesian product.

It is easy to verify that the Cartesian sum $P_n + P_m$ is the five-point $m \times n$ grid graph, and that the strong sum $P_n \oplus P_m$ is the nine-point $m \times n$ grid graph.

Since the grid graphs can be obtained from appropriate graph products of the path graph, the Laplacian matrices of the grid graphs can be obtained from Kronecker products involving the Laplacian matrices of the path graph. If C is a $p \times q$ matrix, and D is $r \times s$, recall that the *Kronecker product* $C \otimes D$ is the $pr \times qs$ matrix with each element d_{ij} of D replaced by the submatrix (Cd_{ij}).

The five-point grid. We consider the $m \times n$ five-point grid, and without loss of generality consider $m \leqq n$. Initially we consider the case when n is even, and $m < n$. At the end of this section, we discuss how the results are modified when n is odd, or $m = n$. We draw the $m \times n$ grid with n vertices in each row and m vertices in each column.

Let Q denote the Laplacian matrix of the five-point $m \times n$ grid graph, R_n denote the Laplacian matrix of the path graph on n vertices, and I_n be the identity matrix of order n. Recall that $\lambda_{k,n}$, $\underline{x}_{k,n}$ denotes the kth eigenpair (when eigenvalues are listed in increasing order) of the path graph with n vertices. The following result is well-known; we include a proof for completeness, and because we wish to indicate how a similar result is obtained for the Laplacian spectrum of a modified nine-point grid.

THEOREM 4.2. *The Laplacian spectrum of the $m \times n$ five-point grid is*

$$\mu_{k,l} = \lambda_{k,n} + \lambda_{l,m},$$

$$\underline{y}_{k,l} = \underline{x}_{k,n} \otimes \underline{x}_{l,m}, \qquad k = 1, \cdots, n, \; l = 1, \cdots, m.$$

Proof. It is easy to verify that the Laplacian matrix of the five-point grid can be expressed in terms of the Laplacian matrix of the path graph as $Q = R_n \otimes I_m + I_n \otimes R_m$. The first term in the sum creates m copies of the path on n vertices, and the second term adds the "vertical" edges, which join neighboring vertices in each column of the grid.

We show that $\mu_{k,l}$, $\underline{y}_{k,l}$ is an eigenpair of Q.

$$\begin{aligned} Q\underline{x}_{k,n} \otimes \underline{x}_{l,m} &= (R_n \otimes I_m)(\underline{x}_{k,n} \otimes \underline{x}_{l,m}) + (I_n \otimes R_m)(\underline{x}_{k,n} \otimes \underline{x}_{l,m}) \\ &= (R_n \underline{x}_{k,n}) \otimes (I_m \underline{x}_{l,m}) + (I_n \underline{x}_{k,n}) \otimes (R_m \underline{x}_{l,m}) \\ &= \lambda_{k,n} \underline{x}_{k,n} \otimes \underline{x}_{l,m} + \underline{x}_{k,n} \otimes \lambda_{l,m} \underline{x}_{l,m} \\ &= (\lambda_{k,n} + \lambda_{l,m})\, \underline{x}_{k,n} \otimes \underline{x}_{l,m}. \end{aligned}$$

The transformation from the first line to the second line uses the associativity of the Kronecker product. □

The smallest eigenvalue $\mu_{1,1} = \lambda_{1,n} + \lambda_{1,m}$ is zero. The next smallest eigenvalue is $\mu_{2,1} = 4 \sin^2 (\phi_n/2)$, and the corresponding eigenvector is

$$\underline{y}_{2,1} = \underline{x}_{2,n} \otimes \underline{x}_{1,m} = \left(\cos \left(i - \frac{1}{2} \right) \phi_n \right) \otimes \underline{1}.$$

The components of $\underline{y}_{2,1}$ are constant along each column of m vertices, and the components decrease from left to right across a row. Columns numbered 1 to $n/2$ have positive components, and the rest of the columns have negative components. The components of this eigenvector of the $m \times n$ five-point grid are plotted in Fig. 1.

These results show that the second eigenvector of the grid can be used to compute good edge separators and vertex separators. Let $\underline{y}$ denote this eigenvector in the following discussion, and let y_l denote the median component ($(mn/2)$th largest component out of mn). Let y_v denote the eigenvector component corresponding to vertex v.

COROLLARY 4.3. *Let V denote the set of vertices of the five-point $m \times n$ grid ($m < n$, n even), and let V be partitioned by its second eigenvector as follows:*

$$A' = \{ v : y_v \leqq y_l \}, \qquad B' = V \backslash A.$$

If E' denotes the set of edges joining A' to B', then E' is an edge separator of size m which separates the grid into two parts each with $(mn/2)$ vertices. Further, if S denotes the set of endpoints of E' which belong to B', then S is a vertex separator of size m which separates the grid into two parts of $(mn/2)$ and $m((n/2) - 1)$ vertices.

The corollary follows from noting that A' consists of vertices in the columns 1 to $n/2$ of the grid, and B' is the remaining set of columns. The edge separator E' consists of the m edges of the grid which join vertices in column $n/2$ to column $(n/2) + 1$. Finally, the vertex separator S consists of vertices in column $(n/2) + 1$. Note that the vertex separator is the same as the separator at the first step of a nested dissection ordering

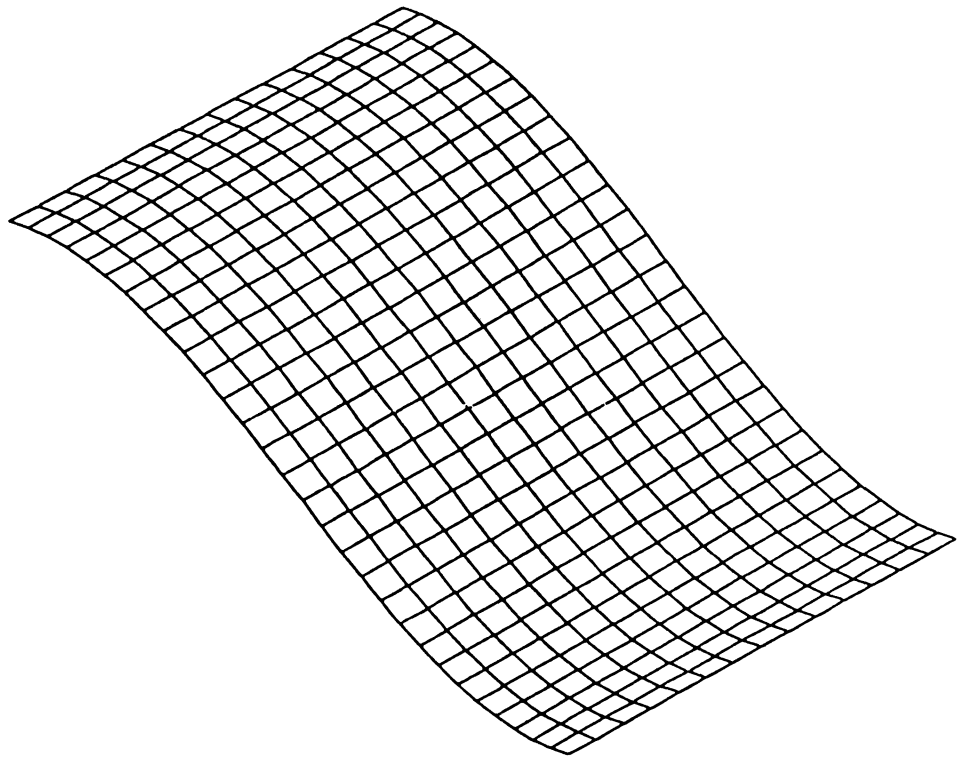


FIG. 1. *The second Laplacian eigenvector of the five-point grid.*

described by George [26]. Buser [14] has shown that the edge separator E' yields the optimal isoperimetric number for grid graphs.

We now consider the case when n is odd or $m = n$. When n is odd, the only difference is that vertices in the middle column ($((n+1)/2$th column) have eigenvector components equal to zero. Columns numbered less than the middle column have positive components, and columns numbered higher have negative components. The middle column can be chosen as a vertex separator. The second case corresponds to a square grid, $m = n$. Then $\mu_{2,1} = \mu_{1,2}$, and the second smallest eigenvalue of Q has geometric multiplicity two. The two linearly independent eigenvectors obtained by the graph product approach are $\underline{y}_{2,1} = \underline{x}_{2,n} \otimes \underline{x}_{1,n}$, and $\underline{y}_{1,2} = \underline{x}_{1,n} \otimes \underline{x}_{2,n}$. The eigenvector $\underline{y}_{2,1}$ has components as described earlier for the rectangular case. The eigenvector $\underline{y}_{1,2}$ has components constant across each row, and decreasing from bottom to top along each column. From these two independent eigenvectors, we obtain a middle column and a middle row as the vertex separators.

Note that when the Lanczos algorithm is used to compute an eigenvector corresponding to the second eigenvalue of the square grid, the eigenvector obtained will be some linear combination of the two eigenvectors $\underline{y}_{1,2}$ and $\underline{y}_{2,1}$. This will lead to a larger vertex separator than the ones above. We report computational results on separators of square grids obtained from the Lanczos algorithm in § 6.

The nine-point grid. Let Q' denote the Laplacian matrix of the nine-point grid, and let D_n be the $n \times n$ diagonal degree matrix of the n-vertex path. As before, let R_n denote the Laplacian matrix of the n-vertex path, and I_n the identity matrix of order n. It is again not difficult to verify that $Q' = R_n \otimes I_m + I_n \otimes R_m + R_n \otimes D_m + D_n \otimes R_m -$

$R_n \otimes R_m$. Unfortunately, the spectrum of Q' cannot be expressed in terms of the spectra of the path graphs, as for the five-point grid.

However, we can first embed the nine-point grid graph in a modified grid, whose Laplacian spectrum is computable in terms of the spectra of the path graphs, and then partition the modified grid. We use the partition of the modified grid to partition the nine-point grid.

The necessary modification to the nine-point grid is as follows. Replace each boundary edge of the $m \times n$ grid by *two edges* joining the same endpoints. Let Q denote the Laplacian of the resulting multigraph.

THEOREM 4.4. *The spectrum of Q is*

$$\mu_{k,l} = 3(\lambda_{k,n} + \lambda_{l,m}) - \lambda_{k,n}\lambda_{l,m},$$

$$\underline{y}_{k,l} = \underline{x}_{k,n} \otimes \underline{x}_{l,m}, \quad \text{for } k = 1, \cdots, n,\ l = 1, \cdots, m.$$

Proof. It is easy to show that $Q = 3(R_n \otimes I_m + I_n \otimes R_m) - R_n \otimes R_m$. A direct computation, as in Theorem 4.2, shows that $\mu_{k,l}, \underline{y}_{k,l}$ is an eigenpair of Q. □

Note that the eigenvectors of the modified nine-point grid are the same as the eigenvectors of the five-point grid, and hence the partitions of the modified nine-point grid are exactly the same as those of the five-point grid.

Finally, we remark that the adjacency spectra of the grids can also be explicitly computed in terms of the adjacency spectra of the path graphs.

5. A spectral partitioning algorithm. In this section we describe an algorithm for finding a vertex separator of a graph by means of its Laplacian matrix. Recall that we require the separator to partition the graph into two parts with nearly equal numbers of vertices in each part, and also that the size of the vertex separator be small.

The algorithm uses a second eigenvector of the Laplacian matrix to compute the partition. We compute x_l, the median value of the components of the eigenvector. Let A' be the set of vertices whose components are less than or equal to x_l, and let B' be the remaining set of vertices. If there is a single vertex with the component corresponding to x_l, then A' and B' differ in size by at most one. If there are several vertices with components equal to x_l, arbitrarily assign such vertices to A' or B' to make these sets differ in size by at most one.

This initial partition of G gives an edge separator in the graph. Let A_1 denote the vertices in A' that are adjacent to some vertex in B', and similarly let B_1 be the set of vertices in B' that are adjacent to some vertex in A'. Let E_1 be the set of edges of G with one endpoint in A_1 and the other in B_1. Then E_1 is an edge separator of G. Note that the subgraph $H = (A_1, B_1, E_1)$ is bipartite.

We require a vertex separator of G, which can be obtained from the edge separator E_1 by several methods. The simplest method is to choose the smaller of the two endpoint sets A_1 and B_1. Gilbert and Zmijewski [29] have computed vertex separators from edge separators in this manner in the context of a parallel Kernighan–Lin algorithm. However, there is a way to choose a *smallest* vertex separator, which can be computed from the given edge separator E'.

The idea is to choose a set S consisting of some vertices from *both* sets of endpoints A_1 and B_1, such that every edge in E_1 is incident on at least one of the vertices in S. The set S is a vertex separator in the graph G, since the removal of these vertices causes the deletion of all edges incident on them, and this latter set of edges contains the edge separator E_1. The set S is a *vertex cover* (*cover*) of the bipartite graph H.

A cover of smallest cardinality is a *minimum cover*. A minimum cover S of the graph H is a smallest vertex separator of G corresponding to the edge separator E_1. It is

well known [36], [47] that a minimum cover of a bipartite graph H can be computed by finding a maximum matching, since these are dual concepts.

In general, S will consist of vertices from both A_1 and B_1. Let A_s and B_s denote the vertices of S that belong to A_1 and B_1, respectively. Then S separates G into two subgraphs with vertex sets $A = A' \setminus A_s$, $B = B' \setminus B_s$. Usually the structure of H permits some freedom in the choice of the sets A_s and B_s; only the sum $|A_s| + |B_s|$ is invariant. This freedom can be used to make the two sets A and B less unequal in size. The sets A_s and B_s may be computed from a canonical decomposition of bipartite graphs called the Dulmage–Mendelsohn decomposition, which is induced by a maximum matching. An implementation of this decomposition is described in [52].

The *Spectral Partitioning Algorithm* is summarized in Fig. 2.

Complexity of the algorithm. In finite precision arithmetic, how accurately must the components of a Laplacian eigenvector be computed to ensure that the vertices are correctly partitioned with respect to the median component? Since the eigenvector components are algebraic numbers, it follows from a discussion in Aspvall and Gilbert [6] that only a polynomial number of bits are needed to order the components of a second eigenvector correctly. In theory, this can be computed in polynomial time by any algorithm that is at least linearly convergent.

In practice, we will have to be content with eigenvector components that are accurate to a fixed number of digits. Since the Lanczos algorithm is an iterative algorithm, the number of Lanczos steps required to approximately compute a second eigenvector will depend on the accuracy desired in the eigenvector. In exact arithmetic, the distribution of the eigenvalues of the Laplacian matrix Q is the primary factor which influences the number of steps required to approximate a second eigenvector (§ 12.4, Parlett [48, § 12.4]. We will assume that the number of iterations of the Lanczos algorithm required to compute a second eigenvector to a small number of digits (say, four) is bounded by a constant. Our experiments in § 7 indicate that this is a reasonable assumption. Each iteration of the Lanczos algorithm costs $O(e)$ flops, and by our assumption, a second eigenvector can also be approximated to a few digits in $O(e)$ flops.

The median component of the eigenvector can be obtained by an algorithm that selects the kth element out of n. This can be done in $O(n)$ time in the worst case by a well-known algorithm of Blum, Floyd, Pratt, Rivest, and Tarjan. This algorithm finds the desired element by repeatedly partitioning a subarray with respect to a pivot element, without sorting the array.

1. Compute the eigenvector $\underline{x}_2$ and the median value x_l of its components;
2. Partition the vertices of G into two sets:
   $A' = \{\text{vertices with } x_v \leqq x_l\}$;
   $B' = V \setminus A$;
   If $|A'| - |B'| > 1$, move enough vertices with components equal to x_l from A' to B' to make this difference at most one;
3. Let A_1 be the set of vertices in A' adjacent to some vertex in B';
   Let B_1 be the set of vertices in B' adjacent to some vertex in A';
   Compute $H = (A_1, B_1, E_1)$, the bipartite subgraph induced by the vertex sets A_1, B_1;
4. Find a minimum vertex cover S of H by a maximum matching;
   Let $S = A_s \cup B_s$, where $A_s \subseteq A_1$, $B_s \subseteq B_1$;
   S is the desired vertex separator, and separates G into subgraphs with vertex sets $A = A' \setminus A_s$, $B = B' \setminus B_s$.

FIG. 2. *The Spectral Partitioning Algorithm.*

The partition into the sets A and B can be done in $O(n)$ time. The bipartite graph H can be generated in $O(e)$ time, by examining the adjacency list of each vertex at most once. Let m be the smaller of $|A'|$ and $|B'|$, and let $e' \equiv |E'|$. A maximum matching and a minimum cover S can be obtained in $O(\sqrt{m}e') = O(\sqrt{n}e)$ time by an algorithm of Hopcroft and Karp. Thus the worst-case time complexity of the *Spectral Partitioning Algorithm* is $O(\sqrt{n}e)$.

Some comment is necessary about the above analysis. In practice, the matching is obtained quite fast. Several matching algorithms have been efficiently implemented in [18], [20], [52], and these algorithms exhibit $O(n + e)$ time complexity in practice. Also, we used a less sophisticated median-finding algorithm, which is $O(n)$ in the average-case, and $O(n^2)$ in the worst-case. In practice, the dominant step in the *Spectral Partitioning Algorithm* is the computation of a second eigenvector by the Lanczos algorithm.

6. Results. In this section, we report computational results obtained from the *Spectral Partitioning Algorithm* and provide comparisons with several other separator algorithms: a modified level-structure separator algorithm implemented in Sparspak, the *Kernighan–Lin algorithm*, the *Fiduccia–Mattheyses algorithm* as implemented by Leiserson and Lewis [38], and the separator algorithm of Liu [42] based on the Multiple Minimum Degree algorithm. We implemented the spectral algorithm, the modified Sparspak separator algorithm, and the Kernighan–Lin algorithm; results for the last two algorithms were obtained from Lewis (personal communication) and Liu's paper [42]. Several sparse matrices from the Boeing–Harwell collection [19] and five- and nine-point grids are partitioned using these algorithms.

Our primary goal in this paper is to establish that the spectral algorithm computes separators that compare favorably with separators computed by previous algorithms. Thus in this section, we report statistics about the quality of the separators computed by the various algorithms. In the next § 7, we report the time required to compute the second Laplacian eigenvector (the dominant computation in the spectral algorithm) for a few representative problems.

In current work, we are implementing a parallel Lanczos algorithm for computing the second eigenvector *in parallel*. This algorithm will be used to compute the separators in parallel. The parallel separator algorithm will then be used to recursively find separators and thereby to compute, in parallel, orderings appropriate for parallel factorizations.

Arioli and Duff [5] have reported results on generating bordered block triangular forms of unsymmetric matrices by finding separators in a directed graph associated with the matrix. Their goal was to use the bordered block triangular form for the parallel solution of large, sparse systems of equations.

The spectral algorithm. We computed vertex and edge separators using the *Spectral Partitioning Algorithm* from the second Laplacian eigenvector. The Lanczos algorithm was terminated either when the approximate eigenvector satisfied the eigenvalue equation to a residual of 10^{-6} or when 300 Lanczos steps were performed. The partitions obtained with the *Spectral Partitioning Algorithm* are tabulated in Table 1. In this table, we list the edge separator first and the vertex separator next, since the former is computed first, and the latter is computed from the former. The edge separator E_1 separates the graph into two parts A' and B'. The sizes of these sets are shown in the first group of three columns in the table. We show two vertex separators obtained from E_1: the first vertex separator is chosen to be the smaller endpoint set of E_1; in the table, this set is denoted A_1. The second vertex separator S includes subsets of vertices from both endpoint sets, and is computed by means of a maximum matching to be a minimum vertex cover of the bipartite graph induced by E_1.

TABLE 1
Partitions using median component of the second Laplacian eigenvector.

	Edge separator			Vertex separators					
				Endpoint set			Matching		
Key	$\|E_1\|$	$\|A'\|$	$\|B'\|$	$\|A_1\|$	$\|A'\| - \|A_1\|$	$\|B'\|$	$\|S\|$	$\|A\|$	$\|B\|$
BCSPWR09	34	862	861	22	840	861	20	857	846
BCSPWR10	44	2,650	2,650	35	2,615	2,650	31	2,623	2,646
BCSSTK13	3,585	1,002	1,001	295	707	1,001	236	862	905
CAN 1072	165	536	536	53	483	536	33	525	514
DWT 2680	85	1,340	1,340	29	1,311	1,340	28	1,313	1,339
JAGMESH	50	468	468	26	442	468	26	442	468
LSHP3466	121	1,733	1,733	61	1,672	1,733	61	1,672	1,733
NASA1824	740	912	912	103	809	912	102	839	883
NASA2146	934	1,073	1,073	96	977	1,073	74	1,036	1,036
NASA4704	1,324	2,352	2,352	185	2,167	2,352	172	2,266	2,266
GRD61.101.5	61	3,111	3,050	61	3,050	3,050	61	3,050	3,050
GRD61.101.9	181	3,111	3,050	61	3,050	3,050	61	3,050	3,050
GRD80.80.5	80	3,200	3,200	80	3,120	3,200	80	3,120	3,200
GRD80.80.9	238	3,200	3,200	80	3,120	3,200	80	3,120	3,200

For six of the Boeing–Harwell problems, the matching method computes vertex separators that are almost the same size as the smaller endpoint set. However, on the CAN 1072 problem, the separator from the matching method is almost 40 percent smaller. On the average problem in this test set, matching finds a separator that is about 11 percent smaller than the separator obtained from the endpoint set. Further, since there are two choices for the minimum cover, a good choice also makes the two part sizes less different. Thus the use of matching techniques seems to be recommended in this context.

The edge separators obtained are small relative to the total number of edges in each graph, except for the BCSSTK13 problem, which has a high average degree. For all problems, except two, the vertex separators obtained are also relatively small (fractional separator size $s < 0.04$) in comparison to the parts generated by the separators. The exceptions are BCSSTK13 and NASA1824. Both these problems have large second eigenvalue λ_2. For BCSSTK13, $\lambda_2 \approx 0.65$; in contrast, for the 80×80 nine-point grid, which has good separators, $\lambda_2 \approx 4.6 \times 10^{-3}$.

For the grid graphs, good vertex separators can be computed by explicitly computing the second eigenvector by the methods in § 4. Here, we investigate the partitions obtained by the spectral algorithm with the eigenvector computed by the Lanczos algorithm. We partitioned the 61×101 grids initially into two sets with 3050 (50 columns) and 3111 (51 columns) vertices. The edge separator obtained joins vertices in the fiftieth column to vertices in the fifty-first column. The vertex separator computed is the middle (fifty-first) column.

In the square grids, the second eigenvalue has geometric multiplicity two, and there are two linearly independent eigenvectors. The eigenvectors in § 4, $\underline{y}_{2,1}$ and $\underline{y}_{1,2}$, obtained by the Kronecker products of the Laplacian eigenvectors of the path, can be used to compute two sets of edge separators. One edge separator joins vertices in the fortieth column to vertices in the forty-first column, and the other joins vertices in the fortieth row to the forty-first row. In general, the Lanczos algorithm will compute a linear combination of the two eigenvectors described above, leading to a different (and large) edge separator. However, for the starting vector we used, the Lanczos algorithm converged to

the eigenvector $y_{1,2}$, and the latter edge separator was computed. (The choice of the start vector is described in § 7.)

We now compare the quality of the separators computed by the spectral algorithm with separators computed from several other algorithms.

The modified level-structure separator algorithm. The separator routine in Sparspak, FNDSEP, finds a pseudoperipheral vertex in the graph, and generates a level structure from it. It then chooses the median level in the level structure as the vertex separator. However, this choice may separate the graph into widely disparate parts. We modified this routine such that the vertex separator is chosen to be the smallest level k such that the first k levels together contain more than half the vertices. A vertex separator is obtained by removing from the vertices in level k those vertices that are not adjacent to any vertex in level $k+1$. By the construction of the level structure, the removed vertices are adjacent to vertices in level $k-1$, and hence these are added to the part containing vertices in the first $k-1$ levels. The other part has vertices in levels $k+1$ and higher. We can also obtain two edge separators using the level structure from the set of edges joining the vertex separator to the two parts A and B.

Statistics about the edge and vertex separators computed by this technique are shown in Table 2. In this table, the vertex separator is listed first and then the edge separator since the former is computed first and the latter is obtained from the former.

The *Spectral Partitioning Algorithm* computes smaller vertex separators than the Sparspak separator algorithm; on the average problem in the Boeing–Harwell test set, the spectral vertex separator is about half the size of the Sparspak vertex separator. The spectral algorithm also succeeds in keeping the part sizes less disparate than the latter algorithm. The average difference in the part sizes is about 7 percent for the Sparspak separator, but there are problems for which this difference is greater than 20 percent.

For most problems, the spectral algorithm also finds smaller edge separators in the graph than the Sparspak level-structure separator algorithm. There are a few problems where the best edge separator obtained by the latter algorithm is smaller than that obtained by the spectral algorithm, but the former edge separators separate the graph into parts with widely differing sizes. In the spectral algorithm, equal part sizes can be obtained by

TABLE 2
Partitions from automated nested dissection.

	Vertex separator			Edge separators					
Key	$\|S\|$	$\|A\|$	$\|B\|$	$\|E_1\|$	$\|A\|$	$\|B \cup S\|$	$\|E_2\|$	$\|A \cup S\|$	$\|B\|$
BCSPWR09	68	762	893	80	762	961	130	830	893
BCSPWR10	169	2,421	2,710	209	2,421	2,879	317	2,590	2,710
BCSSTK13	302	764	937	3,035	764	1,239	4,792	1,066	937
CAN 1072	64	478	530	108	478	594	342	542	530
DWT 2680	28	1,327	1,325	84	1,327	1,353	84	1,355	1,325
JAGMESH	26	455	455	50	455	481	50	481	455
LSHP3466	59	1,711	1,696	118	1,711	1,755	116	1,770	1,696
NASA1824	137	839	848	910	839	985	1,347	976	848
NASA2146	131	1,008	1,007	1,473	1,008	1,138	1,569	1,139	1,007
NASA4704	296	2,245	2,163	2,134	2,245	2,459	2,424	2,541	2,163
GRD61.101.5	61	3,050	3,050	121	3,050	3,111	121	3,111	3,050
GRD61.101.9	111	3,025	3,025	327	3,025	3,131	333	3,131	3,025
GRD80.80.5	80	3,160	3,160	158	3,160	3,240	158	3,240	3,160
GRD80.80.9	113	3,136	3,151	333	3,136	3,264	339	3,249	3,151

partitioning with respect to the median eigenvector component; any other choice of part sizes can also be obtained by partitioning with respect to the appropriate component. Since edge separators are computed in the Sparspak algorithm by means of a level structure, part sizes cannot be controlled as effectively.

The Kernighan–Lin algorithm. The *Kernighan–Lin algorithm* is a heuristic algorithm for computing small edge separators. We investigated the use of this algorithm separately and in conjunction with the *Spectral Partitioning Algorithm*, to compute edge and vertex separators.

The Kernighan–Lin algorithm begins with an initial partition of the graph into two subsets A', B', which differ in their sizes by at most one. At each iteration, the algorithm chooses two subsets of equal size to swap between A and B, thereby reducing the number of edges that join A to B. We refer the reader to Kernighan and Lin [35], or Gilbert and Zmijewski [29] for a detailed description of how the algorithm chooses the subsets to be swapped. The algorithm terminates when it is no longer possible to decrease the size of the edge separator by swapping subsets. In our implementation, each iteration could require $O(n^3)$ time, though in practice, often the running time is $O(n^2 \log n)$, the time required for n sorts.

One initial partition we could use is the edge partition obtained from the *Spectral Partitioning Algorithm*, and a second choice is to use a randomly computed initial partition. We consider the four graphs with the largest edge separators from Table 1, and report the sizes of the edge and vertex separators obtained with the Kernighan–Lin algorithm in Table 3. An edge separator was computed first, and then a vertex separator was obtained as before by matching methods. The column labeled "SP" corresponds to the output of the spectral algorithm, "SP, KL" corresponds to the Kernighan–Lin algorithm with initial partition from the spectral algorithm, and "KL" corresponds to the Kernighan–Lin algorithm with a random initial partition.

The application of the Kernighan–Lin algorithm with the spectral partition as input succeeds in reducing the sizes of the edge separator considerably for two of the four problems. Thus if one is primarily concerned with small edge separators, applying the Kernighan–Lin algorithm to the partition produced by spectral algorithm could be

TABLE 3

Partitions from the Kernighan–Lin algorithm. The first table describes the edge separators, and the second, vertex separators.

			$\lvert E_1 \rvert$		
Key	$\lvert A' \rvert$	$\lvert B' \rvert$	SP	SP, KL	KL
BCSSTK13	1,002	1,001	3,585	2,880	3,550
NASA1824	912	912	740	739	739
NASA2146	1,073	1,073	934	870	870
NASA4704	2,352	2,352	1,324	1,313	1,525

	SP			SP, KL			KL		
Key	$\lvert S \rvert$	$\lvert A \rvert$	$\lvert B \rvert$	$\lvert S \rvert$	$\lvert A \rvert$	$\lvert B \rvert$	$\lvert S \rvert$	$\lvert A \rvert$	$\lvert B \rvert$
BCSSTK13	236	862	905	250	870	883	284	772	947
NASA1824	103	839	883	102	830	892	102	830	892
NASA2146	74	1,036	1,036	74	1,036	1,036	74	1,036	1,036
NASA4704	172	2,266	2,266	172	2,266	2,266	204	2,163	2,337

worthwhile. However, the size of the vertex separator is not improved. For two of the problems, the size remains the same; for a third, it decreases by one, and the size increases for a fourth problem. Also, for two of the four problems, the spectral algorithm by itself finds better vertex separators than those obtained by the Kernighan–Lin algorithm alone.

Gilbert and Zmijewski [29] have observed that the quality of the partition found by the Kernighan–Lin algorithm strongly depends on the quality of the initial partition. They show for a grid graph that it is possible to choose a bad initial partition for the Kernighan–Lin algorithm such that the algorithm will not find a minimum edge separator.

Edge separators obtained from the Kernighan–Lin algorithm with initial spectral partition are better than those obtained from the application of the Kernighan–Lin algorithm with random initial partitions for two of the four problems. Use of the initial partition from spectral algorithm also helps the Kernighan–Lin algorithm to converge faster. On these four problems, the Kernighan–Lin algorithm ran on the average about 3.2 times faster when the spectral partition was used. Thus the spectral algorithm could be used to generate initial partitions of high quality for the Kernighan–Lin algorithm.

The Leiserson–Lewis and Liu algorithms. In [38] Leiserson and Lewis have used the *Fiduccia–Mattheyses algorithm* [22] to compute vertex separators and then to order sparse matrices. Liu [42] uses the Multiple Minimum Degree ordering algorithm to compute vertex separators, and then improves the separator (by decreasing its size and making the parts less unequal) by a matching technique. He uses his separator algorithm in [41] to compute a good ordering for parallel factorization. In both implementations sparse matrices from the Boeing–Harwell collection are used, so we are able to give a direct comparison of the first level vertex separator. The data in Table 4 are obtained directly from Liu's report [42] and from Lewis (personal communication). In both cases we have added small disconnected components, which were created by the vertex separators, to the smaller of the two sets $|A|$ or $|B|$.

The results in Table 4 show that the Leiserson–Lewis implementation and Liu's algorithm find separators which are smaller than the spectral separators for the two power network problems. The reason for this seems to be that for these problems, the spectral algorithm computes a partition from an eigenvector that has converged to fewer than two correct digits. The accuracy of the computed eigenvectors is discussed in greater detail in § 7. For the other four problems, the Leiserson–Lewis and the spectral separators are almost the same size. Liu's algorithm finds a larger separator than the spectral algorithm for the BCSSTK13 problem. The Leiserson–Lewis algorithm does a good job of keeping the part size roughly equal. There is greater difference between the part sizes in Liu's algorithm. However, neither the Leiserson–Lewis algorithm nor Liu's algorithm offers any easy prospect for a parallel implementation. A factor which cannot be evaluated in

TABLE 4
Vertex separators from the Leiserson–Lewis and the Liu algorithms.

	Leiserson–Lewis			Liu		
Key	$\|S\|$	$\|A\|$	$\|B\|$	$\|S\|$	$\|A\|$	$\|B\|$
BCSPWR09	7	858	854	8	1,026	689
BCSPWR10	18	2,641	2,634	19	2,661	2,620
BCSSTK13	242	892	869	298	941	764
CAN1072	34	522	516	38	665	368
DWT2680	28	1,339	1,313	26	1,369	1,283
LSHP3466	57	1,708	1,701	61	1,727	1,678

this comparison is the relative execution time of the algorithms, since these algorithms were implemented on different computers.

7. Convergence. The dominant computation in the *Spectral Partitioning Algorithm* is the computation of the second eigenvector of the Laplacian matrix by the Lanczos algorithm. Since the Lanczos algorithm is an iterative algorithm, the number of iterations and the time required to compute this eigenvector is dependent on the number of correct digits needed in the eigenvector components. In this section, we describe the details of an implementation of the Lanczos algorithm for computing this eigenvector, and study how the quality of computed separators depends on the accuracy in the second eigenvector.

The Lanczos algorithm. The most efficient algorithm for computing a few eigenvalues and eigenvectors of large, sparse symmetric matrices is the Lanczos algorithm. Since the Lanczos algorithm is discussed extensively in the textbook literature [30], [48], we do not include a detailed description of the standard algorithm here. The convergence of the Lanczos algorithm depends critically on the distribution of the eigenvalues of the underlying matrix. Usually the extreme eigenpairs, i.e., the largest and smallest, are found first. However it is also known that for operators such as the discrete Laplacian for a grid problem, or more generally for positive definite finite element matrices which are approximations to elliptic operators, the Lanczos algorithm converges in most cases to the extreme right, i.e., the very large eigenvalues, before delivering good approximations to the eigenvalues close to zero. This behavior can be explained with the so-called Kaniel–Paige–Saad theory (see [48]). When computing the smallest positive eigenvalue of the Laplacian matrix Q, one faces exactly the same situation: the Lanczos algorithm delivers very good approximations to the large eigenvalues before converging to the desired second smallest eigenvalue. Thus the Lanczos algorithm potentially requires long runs before it computes an approximation to the second eigenpair.

A potential modification which can be incorporated in the Lanczos algorithm for faster computation of the second eigenvector would be to apply the shifted and inverted operator, i.e., to consider the eigenvalue problem

$$(Q-\sigma I)^{-1}\underline{u}=\mu\underline{u}.$$

This is a standard technique in finite element applications [31], and it has been used very successfully in a variety of implementations of the Lanczos algorithm [21], [32], [49], [56]. In the situation here, a shift σ chosen near zero would result in rapid convergence to the eigenvalue λ_2. This approach cannot be taken here, since it requires the factorization of the matrix $Q-\sigma I$, which is a large sparse symmetric matrix with the same sparsity structure as M. Our original goal, however, is to find an efficient reordering of M, so to be able to factor it efficiently. Hence the "shift and invert" approach would require us to factor a matrix closely related to M, and thus cannot be considered in this application.

Reorthogonalization has also been used in the Lanczos algorithm to improve both its reliability and computational efficiency [49], [50], [57]. However, in this application we do not require reorthogonalization techniques in their full generality. Only a limited amount of reorthogonalization is necessary for the computation of the second eigenpair. No reorthogonalizations are performed at the right end of the spectrum, with respect to the large eigenvalues, since there is no interest in the accurate computation of eigenvalues at this end. Also it is unlikely that preserving orthogonality at the right end will have any impact on the convergence of the Lanczos algorithm towards the second smallest eigenvalue, which is at the left end of the spectrum. The first eigenvector $\underline{x}_1$ of Q is $\underline{e}$, the vector of all ones, and this vector can be used for reorthogonalization at the left end of

the spectrum. At each step we explicitly orthogonalize the current Lanczos vector against $\underline{e}$. This is effectively a deflation of the problem and now the eigenpair λ_2, $\underline{x}_2$ will be computed as the first eigenpair at the left end of the spectrum.

Another important consideration for the Lanczos algorithm is the choice of a starting vector. In the absence of any other information, a random starting vector is appropriate. However, many practical matrix problems are presented already in an ordering relevant to the formulation of the problem, sometimes even in an ordering which is close to a good band or envelope ordering. In this case it is desirable to transmit this ordering information to the Lanczos algorithm. This was accomplished by setting the starting vector in the Lanczos algorithm to $\underline{r}$, with $\underline{r}_i = i - (n + 1)/2$. This choice also makes the starting vector orthogonal to $\underline{e}$. In most cases this resulted in faster convergence to the second eigenvector.

Finally, another point needs to be mentioned. Considering the simple structure of the Laplacian matrix Q, and the seeming simplicity of the task of computing just one eigenpair at the left end of the spectrum, one might be inclined to avoid the complexities of the Lanczos algorithm and attempt to solve this problem with a simple shifted power method with a deflation procedure analogous to the one described above. This was tried as a first attempt at the computation of a second eigenvector, but with very poor results. The power method converged exceedingly slowly, in many cases exhibiting the phenomenon of *misconvergence* [51]. This meant that the power method settled down at an eigenvalue of Q, which was not the Fiedler value, and whose eigenvector correspondingly delivered a very poor reordering. The results here support the claims of [51] that even in the simplest cases the Lanczos algorithm is the method of choice, when computing eigenvalues of large, sparse, symmetric matrices.

Figure 3 contains a description of the specialized Lanczos algorithm for computing the second Laplacian eigenvector. In this algorithm, we have assumed that the Laplacian $Q(G)$ is irreducible, or equivalently that the graph G is connected. Many of the sparse matrices from the Boeing–Harwell collection have disconnected adjacency graphs. If a graph has k connected components, the first k eigenvectors correspond to the multiple eigenvalue zero, and the $k + 1$th eigenvector is used to partition the graph. A simple modification to the above algorithm can be used to compute this eigenvector.

Convergence and quality of separators. We now present our results on the number of iterations and the time required by the Lanczos algorithm as the second eigenvector is computed to a set of different tolerances. The tolerance criterion, *tol*, is the 2-norm of the residual vector $Q\underline{u} - \lambda\underline{u}$, where λ, $\underline{u}$ are the computed quantities at the current step in the algorithm. We also study the quality of the vertex separators obtained from these approximate eigenvectors.

We report results for a few representative problems from the Boeing–Harwell collection and for two grid problems in Table 5. The iteration numbers reported are multiples of twelve, since we checked for convergence in the Lanczos algorithm by an eigendecomposition of the tridiagonal matrix only after every twelve iterations. Times are in seconds

1. Given the sparsity structure of a matrix M, form the Laplacian matrix Q.
2. Pick a starting vector $\underline{r}$, with $r_i = i - (n + 1)/2$.
3. Carry out a Lanczos iteration with the matrix Q and starting vector $\underline{r}$. At each step orthogonalize the Lanczos vector against the vector $\underline{e}$. Stop when a second eigenvector has been determined to sufficient accuracy.

FIG. 3. *The Lanczos algorithm for computing the second Laplacian eigenvector.*

TABLE 5
Convergence results. Times are in seconds on a Cray Y-MP. *A blank entry in the separator column indicates that the separator is unchanged from the row above it.*

Key	*tol*	Iterns	Time	$\|S\|$	$\|A\|$	$\|B\|$
NASA4704	10^{-1}	24	0.27	172	2,266	2,266
	10^{-2}	60	0.65			
	10^{-3}	72	0.80			
	10^{-4}	96	1.10			
	10^{-5}	108	1.30			
	10^{-6}	120	1.50			
BCSSTK13	10^{-1}	36	0.23	236	905	862
	10^{-2}	36	0.23			
	10^{-3}	48	0.30			
	10^{-4}	60	0.39			
	10^{-5}	72	0.49			
	10^{-6}	84	0.60			
BCSPWR10	10^{-1}	24	0.24	171	2,619	2,510
	10^{-2}	84	0.92	72	2,642	2,586
	10^{-3}	252	7.20	34	2,643	2,623
	10^{-4}	300	11.90	31	2,646	2,623
GRD61.101.5	10^{-2}	12	0.15	101	3,050	3,010
	10^{-3}	36	0.42	61	3,050	3,050
	10^{-4}	96	1.26			
	10^{-5}	108	1.47			
	10^{-6}	120	1.67			
GRD61.101.9	10^{-1}	12	0.16	101	3,030	3,030
	10^{-2}	24	0.30	61	3,050	3,050
	10^{-3}	108	1.53			
	10^{-4}	120	1.76			
	10^{-5}	144	2.38			
	10^{-6}	156	2.70			

on a Cray Y-MP, using our vectorized Lanczos code. For each value of *tol*, we report the size of the vertex separator and the corresponding part sizes computed by the *Spectral Partitioning Algorithm*. Blank entries in the separator columns mean that the separator computed is the same as the one obtained with the previous tolerance.

For most of the problems that we have computational results, it is only necessary to compute the second eigenvector to a tolerance of about 10^{-2}, to obtain the best separator obtained by the spectral algorithm. This accuracy requires only a modest number of Lanczos iterations, and can be obtained reasonably fast. One class of notable exceptions is the power network problems, illustrated by BCSPWR10 in the table. For these problems, the average degree of a vertex is small (about 1.5 for BCSPWR10), and the diameter of the graph is large; hence computing eigenvector components (which represent global information about the graph) is relatively slow. A large number of iterations are thus necessary to compute the second eigenvector accurately. In the BCSPWR10 problem, after 300 iterations, the norm of the residual in the eigenvalue equation was about 10^{-4}. In this problem, the vertex separator decreases in size as the eigenvector becomes more accurate.

8. Conclusions. We have considered an algebraic approach for computing vertex separators and have shown that the eigenvalues of the Laplacian matrix can be used to obtain lower bounds on the sizes of the separators. We have described a heuristic algorithm for computing vertex separators from the second eigenvector of the Laplacian. Thus the

spectral algorithm uses global information about the graph to compute separators. It is enough to compute the eigenvector to low accuracy to obtain good separators for most problems. Our results show that the spectral separators compare quite favorably with separators computed by previous algorithms. The spectral algorithm has an advantage over these algorithms in that its dominant computation is an eigenvector computation (which involves mainly dense and sparse vector operations), and is fairly straightforward to compute efficiently on medium-size multiprocessors used in scientific computing. For previous algorithms, it is either not clear how to implement them in parallel or the amount of parallelism is not high. Since the spectral algorithm involves mainly floating point computations, we expect it to be attractive over primarily combinatorial algorithms on machines like the Cray Y-MP, where floating point arithmetic is considerably faster than integer arithmetic.

The computation of good separators is useful in many divide-and-conquer algorithms. Several of the new parallel algorithms that have been reported to date make use of divide and conquer, and hence the spectral separator algorithm will have applications in parallel algorithm design. The spectral algorithm may also be useful in VLSI layout problems, since good edge separators are needed in this context.

But our immediate intent was to use the spectral separator algorithm to compute good orderings for parallel sparse factorizations. More work remains to be done in order to accomplish this goal. First, we intend to compute and study the quality of orderings obtained by the recursive application of the spectral algorithm. Second, we are working on the fast sequential and parallel computation of the second Laplacian eigenvector. The latter algorithm will enable us to compute the separators (and thereby orderings for parallel factorizations) in parallel. We are investigating the Lanczos algorithm and the generalized Davidson's algorithm of Morgan and Scott [46] in this regard.

Finally, much remains to be understood about the theoretical underpinnings of the spectral separator algorithm. It will be useful to obtain results on the quality of the partitions computed by the Laplacian eigenvector components. It will also be helpful to identify classes of graphs that are partitioned well by the spectral algorithm.

REFERENCES

[1] N. ALON, *Eigenvalues and expanders*, Combinatorica, 6 (1986), pp. 83–96.

[2] N. ALON, Z. GALIL, AND V. D. MILMAN, *Better expanders and superconcentrators*, J. Algorithms, 8 (1987), pp. 337–347.

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

[4] W. N. ANDERSON AND T. D. MORLEY, *Eigenvalues of the Laplacian of a graph*, Linear and Multilinear Algebra, 18 (1985), pp. 141–145. (Originally published as University of Maryland Tech. Report TR-71-45, 1971).

[5] M. ARIOLI AND I. S. DUFF, *Experiments in tearing large sparse systems*, Tech. Report CSS 217, Computer Science and Systems Division, Harwell Lab, Harwell, UK, January 1988.

[6] B. ASPVALL AND J. R. GILBERT, *Graph coloring using eigenvalue decomposition*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 526–538.

[7] E. R. BARNES, *An algorithm for partitioning the nodes of a graph*, SIAM J. Algebraic Discrete Methods, 3 (1982), pp. 541–550.

[8] ———, *Partitioning the nodes of a graph*, in Graph Theory with Applications to Algorithms and Computer Science, Y. Alavi, G. Chartrand, L. Lesniak, D. R. Lick, and C. E. Wall, eds., John Wiley, 1985, pp. 57–72.

[9] E. R. BARNES AND A. J. HOFFMAN, *Partitioning, spectra, and linear programming*, in Progress in Combinatorial Optimization, W. E. Pulleyblank, ed., Academic Press, New York, 1984, pp. 13–25.

[10] F. BIEN, *Constructions of telephone networks by group representations*, Notices Amer. Math. Soc., 36 (1989), pp. 5–22.

[11] N. L. BIGGS, *Algebraic Graph Theory*, Cambridge University Press, Cambridge, 1974.
[12] R. B. BOPPANA, *Eigenvalues and graph bisection: an average case analysis*, in 28th Annual Symposium on Foundations of Computer Science, 1987, pp. 280–285.
[13] T. BUI, S. CHAUDHURI, T. LEIGHTON, AND M. SIPSER, *Graph bisection algorithms with good average-case behavior*, in 25th Annual Symposium on Foundations of Computer Science, 1984, pp. 181–192.
[14] P. BUSER, *On the bipartition of graphs*, Discrete Appl. Math., 9 (1984), pp. 105–109.
[15] D. M. CVETKOVIC, M. DOOB, AND H. SACHS, *Spectra of Graphs*, Academic Press, New York, 1980.
[16] W. E. DONATH AND A. J. HOFFMAN, *Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices*, IBM Technical Disclosure Bulletin, 15 (1972), pp. 938–944.
[17] ———, *Lower bounds for the partitioning of graphs*, IBM J. Res. Develop., 17 (1973), pp. 420–425.
[18] I. S. DUFF, *On algorithms for obtaining a maximum transversal*, ACM Trans. Math. Software, 7 (1981), pp. 315–330.
[19] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[20] I. S. DUFF AND T. WIBERG, *Implementations of $O(n^{1/2}\tau)$ assignment algorithms*, ACM Trans. Math. Software, 14 (1988), pp. 267–287.
[21] T. ERICSSON AND A. RUHE, *The spectral transformation Lanczos method*, Math. Comp., 34 (1980), pp. 1251–1268.
[22] C. FIDUCCIA AND R. MATTHEYSES, *A linear time heuristic for improving network partitions*, in ACM–IEEE 19th Design Automation Conference, Las Vegas, NV, IEEE Press, 1982, pp. 175–181.
[23] M. FIEDLER, *Algebraic connectivity of graphs*, Czechoslovak Math. J., 23 (1973), pp. 298–305.
[24] ———, *A property of eigenvectors of non-negative symmetric matrices and its application to graph theory*, Czechoslovak Math. J., 25 (1975), pp. 619–633.
[25] ———, *Special Matrices and their Applications in Numerical Mathematics*, Martinus Nijhoff Publishers, Dodrecht, The Netherlands, 1986.
[26] J. A. GEORGE, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 15 (1978), pp. 1053–1069.
[27] J. A. GEORGE AND J. W-H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, Englewood Cliffs, NJ, 1981.
[28] J. R. GILBERT, *Separator theorems and sparse Gaussian elimination*, Ph.D. thesis, Stanford University, Stanford, CA, 1981.
[29] J. R. GILBERT AND E. ZMIJEWSKI, *A parallel graph partitioning algorithm for a message passing multiprocessor*, Internat. J. Parallel Programming, 16 (1987), pp. 427–449.
[30] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, Second edition, 1989.
[31] R. G. GRIMES, J. G. LEWIS, AND H. D. SIMON, *Eigenvalue problems and algorithms in structural engineering*, in Large Scale Eigenvalue Problems, J. Cullum and R. Willoughby, eds., North-Holland, Amsterdam, 1986, pp. 81–93.
[32] ———, *The implementation of the block Lanczos algorithm with reorthogonalization methods*, Tech. Report ETA-TR-91, Boeing Computer Services, Seattle, WA, 1988.
[33] R. G. GRIMES, D. J. PIERCE, AND H. D. SIMON, *A new algorithm for finding a pseudoperipheral node in a graph*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 323–335.
[34] A. J. HOFFMAN AND H. W. WIELANDT, *The variation of the spectrum of a normal matrix*, Duke Math. J., 20 (1953), pp. 37–39.
[35] B. W. KERNIGHAN AND S. LIN, *An efficient heuristic procedure for partitioning graphs*, The Bell System Tech. J., 49 (1970), pp. 291–307.
[36] E. L. LAWLER, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, 1976.
[37] T. LEIGHTON AND S. RAO, *An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms*, in 29th Annual Symposium on Foundations of Computer Science, 1988, pp. 422–431.
[38] C. E. LEISERSON AND J. G. LEWIS, *Orderings for parallel sparse symmetric factorization*, Third SIAM Conference on Parallel Processing for Scientific Computing, 1987.
[39] J. G. LEWIS, B. W. PEYTON, AND A. POTHEN, *A fast algorithm for reordering sparse matrices for parallel factorization*, SIAM J. Sci. Statist. Comput., 6 (1989), pp. 1146–1173.
[40] J. W-H. LIU, *Reordering sparse matrices for parallel elimination*, Tech. Report 87-01, Computer Science, York University, North York, Ontario, Canada, 1987; Parallel Computing, to appear.
[41] ———, *The minimum degree ordering with constraints*, Tech. Report CS-88-02, Computer Science, York University, North York, Ontario, Canada, 1988.
[42] ———, *A graph partitioning algorithm by node separators*, ACM Trans. Math. Software, 1989, to appear.

[43] B. MOHAR, *Eigenvalues, diameter, and mean distance in graphs*, Preprint Series Dept. Math. No. 259, University E. K. of Ljubljana, Jadranska 19, 61111 Ljubljana, Yugoslavia, April 1988.
[44] ———, *Isoperimetric numbers of graphs*, J. Combin. Theory, Ser. B, 1988.
[45] ———, *The Laplacian spectrum of graphs*, in Sixth International Conference on Theory and Applications of Graphs, Kalamazoo, MI, 1988.
[46] R. B. MORGAN AND D. S. SCOTT, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 817–825.
[47] C. H. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, NJ, 1982.
[48] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980.
[49] B. N. 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 Math., University of California, Berkeley, CA, 1988.
[50] B. N. PARLETT AND D. S. SCOTT, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.
[51] B. N. PARLETT, H. D. SIMON, AND L. STRINGER, *Estimating the largest eigenvalue with the Lanczos algorithm*, Math. Comp., 38 (1982), pp. 153–165.
[52] A. POTHEN AND C-J. FAN, *Computing the block triangular form of a sparse matrix*. Tech. Report CS-88-51, Computer Science, Penn State, December 1988; ACM Trans. Math. Software, 1990, to appear.
[53] A. POTHEN AND H. D. SIMON, *A parallel iterative algorithm for envelope reduction in sparse matrices*, in preparation.
[54] D. L. POWERS, *Structure of a matrix according to its second eigenvector*, in Current Trends in Matrix Theory, F. Uhlig and R. Grone, eds., Elsevier, New York, 1987, pp. 261–266.
[55] ———, *Graph partitioning by eigenvectors*, Linear Algebra Appl., 101 (1988), pp. 121–133.
[56] D. S. SCOTT, *Block Lanczos software for symmetric eigenvalue problems*, Tech. Report ORNL/CSD 48, Oak Ridge National Lab, Oak Ridge, TN, 1979.
[57] H. D. SIMON, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–136.
[58] J. H. WILKINSON, *Modern error analysis*, SIAM Rev., 13 (1971), pp. 548–568.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 453–465, July 1990

008

SPARSE ORTHOGONAL DECOMPOSITION ON A HYPERCUBE MULTIPROCESSOR*

ELEANOR CHU† AND ALAN GEORGE†

Abstract. In this article the orthogonal decomposition of large sparse matrices on a hypercube multiprocessor is considered. The proposed algorithm offers a parallel implementation of the general row merging scheme for sparse Givens transformations recently developed by Joseph Liu. The proposed parallel algorithm is novel in several aspects. First, a new mapping strategy whose goal is to reduce the communication cost and balance the work load during the entire computing process is proposed. Second, a new sequential algorithm for merging two upper trapezoidal matrices (possibly of different dimensions) is described, wherein the order of computation is different from the standard Givens scheme, and is more suitable for parallel implementation. Third, it is shown that the hypercube network can be employed as a *multi-loop* multiprocessor. The performance of the parallel algorithm applied to a model problem is analyzed and computation/communication complexity results are presented. Finally it is shown that the parallel submatrix merging algorithm can be viewed as a special case of a more general scheme and it is indicated how the generalized scheme may further reduce the communication cost.

Key words. sparse matrix, orthogonal decomposition, parallel computation, Givens rotation, row/submatrix merging, hypercube multiprocessors

AMS(MOS) subject classifications. 65F05, 65F50, 68R10

1. Introduction.

Let A be a large sparse $m \times n$ $(m > n)$ matrix with full column rank. We consider the problem of reducing A to upper triangular form using orthogonal Givens transformations on a hypercube multiprocessor. The decomposition process is commonly expressed as

$$QA = \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where Q is an $m \times m$ orthogonal matrix defined by the sequence of Givens rotations, and R denotes the derived $n \times n$ upper triangular matrix. When A is sparse, some zero entries in A may become nonzero during the computing process and therefore appear in the final structure of R [2], [4], [12], [18]. Because these nonzero elements do not exist in A, they are commonly referred to as *fill*.

It is also known that two kinds of fill may be distinguished when Givens rotations are applied to a sparse matrix A. They are the *fill* in the final structure of R, and the *intermediate fill* which occurs in an initially zero position during the computation but is subsequently annihilated at a later step. Although intermediate fill does not occupy storage in R, it causes higher arithmetic cost. Therefore, serial sparse Givens algorithms usually aim at reducing both kinds of fill. In order to obtain a sparse R, George and Heath [5] make use of the following connection between the factor R and the Cholesky factor of A^TA. First, they note that the factor R is mathematically equal to the Cholesky factor of the symmetric positive definite matrix A^TA. Second,

* Received by the editors August 16, 1989; accepted for publication (in revised form) December 29, 1989. This work was supported in part by Canadian Natural Sciences and Engineering Research Council grant OGP0008111, by Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy contract DE-AC05-84OR21400 with Martin Marietta Energy Systems Inc., by NASA grant NAGW-1457, and by a research grant from the University of Waterloo.

† Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (echu@watfun.waterloo.edu; jageorge@provost.uwaterloo.ca).

they observe that if P_r and P_c are permutation matrices, then

$$(1) \quad (P_r A P_c)^T (P_r A P_c) = P_c^T A^T (P_r^T P_r)(A P_c) = (A P_c)^T (A P_c) = P_c^T (A^T A) P_c .$$

With these observations, they suggest that a symmetric ordering which produces a sparse Cholesky factor for $A^T A$ also yields an equally sparse R if the permuted matrix AP_c is reduced by orthogonal transformations. Although finding an optimal ordering for a symmetric and positive definite matrix is NP-complete [22], there exist a number of good heuristic ordering algorithms which perform well in practice and have efficient implementations [7]. Therefore, it is common practice in sparse matrix computation to subject a given matrix to such an ordering algorithm before determining the structure of R.

Although it is apparent from (1) that the row ordering of A does not have any effect on the sparsity structure of R, it is important in reducing the amount of intermediate fill. In [15] Liu generalizes the row rotations in the George–Heath method [5] to submatrix merging in his general row merging scheme for sparse Givens transformations. The row ordering implicitly imposed by the submatrix merging sequence appears to introduce significantly less intermediate fill in the process of computing R compared to other known row-ordering schemes [9], [10], [11], [19]. It was also shown in [8], [15] that the trade-off for the lower computational cost is only a very modest increase in working storage. Working storage is an important consideration for parallel implementation on local-memory machines, because there is relatively much less memory available on each node processor compared to a sequential machine.

The parallel general row merging scheme we propose is designed for efficient implementation on hypercube multiprocessor architectures, assuming that the columns of the matrix A have been appropriately ordered for a sparse R. An outline of this paper follows. In §2 we briefly review the concept of row merge tree and its use in the general row merging scheme. Interested readers may refer to [5] and [15] for more details about the sequential algorithms. In §3 we describe a new parallel row merging scheme featured by employing the hypercube as a *multi-loop* multiprocessor. In §4 complexity analysis results are presented for a regular grid model problem. Although we analyze the performance of the proposed algorithm only for the model problem, experience in other contexts suggests that the results are representative of the behaviour expected from more general sparse matrices arising from two-dimensional finite element analysis of structural and fluid flow problems. In §5 we show that the parallel submatrix merging algorithm can be generalized to further reduce the communication cost.

2. Row merge tree and the general row merging scheme. For an easy explanation of the concept of row merge tree and its role in guiding the computation in the general row merging scheme, we make use of a k-by-k grid model problem in presenting the definitions of several very closely related tree structures. They are referred to as *elimination tree*, *row merge tree*, *binary row merge tree*, and *reduced row merge tree*. Our definitions follow Liu [15], except that Liu's definition of row merge tree is equivalent to the binary row merge tree defined here.

2.1. Elimination tree and the k-by-k grid problem. The coefficient matrix A of the overdetermined system corresponding to a k-by-k grid has $m = s(k-1)^2$ rows and $n = k^2$ columns, resulting from associating a variable x_i with each of the k^2 grid vertices, and s equations (involving the four variables at the corners of the square) with each of the $(k-1)^2$ small squares. There is an intimate relationship

between the ordering of the grid vertices and the structure of the elimination tree associated with A^TA, because the former amounts to permuting the columns of the coefficient matrix A and thus determines the sparsity structure of R, which in turn determines the structure of the elimination tree as defined below.

DEFINITION 2.1 (ELIMINATION TREE OF A^TA). Given an $m \times n$ matrix A and assuming that A^TA is irreducible, the elimination tree of A^TA is a tree consisting of n vertices each being uniquely labelled by an integer in $\{1, 2, \cdots, n\}$. Let R denote the upper triangular factor from the orthogonal decomposition of A or the Cholesky decomposition of A^TA. If $r_{i,j}$ $(i < j)$ is the leading off-diagonal nonzero in the ith row of R, then vertex j is the parent of vertex i in the elimination tree.

The elimination tree [21] of a square sparse matrix M has been used to set up efficient data structures and to guide serial and parallel computation in factoring M via Gaussian elimination [17]. For general sparse systems, the ordering schemes which generate an elimination tree with minimum or near-minimum height were examined in [13], [14], [16]. For regular grid problems, it is well known that George's nested dissection ordering minimizes the fill in R and yields a balanced elimination tree. We show in Fig. 1 how to generate a nested dissection ordering by recursively defining separators on a 7-by-7 grid. The elimination tree corresponding to the grid in Fig. 1 is displayed in Fig. 2. In Fig. 1, vertices 43 to 49 form the separator S_1^0, which partitions the 7-by-7 grid into two 7-by-3 subgrids. Note that the superscript j in our separator notation S_i^j indicates that there are 2^j separators at this level of recursive partitioning and the subscript i in $\{1, 2, \cdots, 2^j\}$ enumerates them. The vertices 37 to 39 form the separator S_1^1, and the vertices 40 to 42 form the separator S_2^1. S_1^1 and S_2^1 partition the two 7-by-3 subgrids into four 3-by-3 subgrids. Observe that each separator S_j^i corresponds to a chain of length $|S_j^i|$ in the elimination tree.

To obtain the row merge tree of an $m \times n$ matrix A, we add m leaves to the elimination tree of A^TA in the following manner. If A has m_i rows with leading nonzeros in the ith column, m_i leaves are attached to vertex i in the corresponding elimination tree.

21	25	22	43	30	34	31
23	26	24	44	32	35	33
19	27	20	45	28	36	29
37	38	39	46	42	41	40
3	7	4	47	12	16	13
5	8	6	48	14	17	15
1	9	2	49	10	18	11

21	S_2^2	22	S_1^0	30	S_4^2	31
S_5^3	S_2^2	S_6^3	S_1^0	S_7^3	S_4^2	S_8^3
19	S_2^2	20	S_1^0	28	S_4^2	29
S_1^1	S_1^1	S_1^1	S_1^0	S_2^1	S_2^1	S_2^1
3	S_1^2	4	S_1^0	12	S_3^2	13
S_1^3	S_1^2	S_2^3	S_1^0	S_3^3	S_3^2	S_4^3
1	S_1^2	2	S_1^0	10	S_3^2	11

FIG. 1. *Nested dissection ordering of a 7-by-7 grid and separators.*

If the resulting row merge tree is not binary, it can be transformed into a binary tree by removing those parent vertices that have only one child, and by introducing additional interior vertices (binary splitting) if a vertex has more than two children. The transformed binary tree is A's binary row merge tree. For a given $m \times n$ matrix, its binary row merge tree is thus a strictly binary tree with m leaves, each corresponding to a row in the matrix. There are different ways to perform binary splitting of the row merge tree; to find the best possible splitting in the context of sparse QR factorization is a research problem in its own right. More on the splitting criterion and strategies can be found in [2], [15], [23].

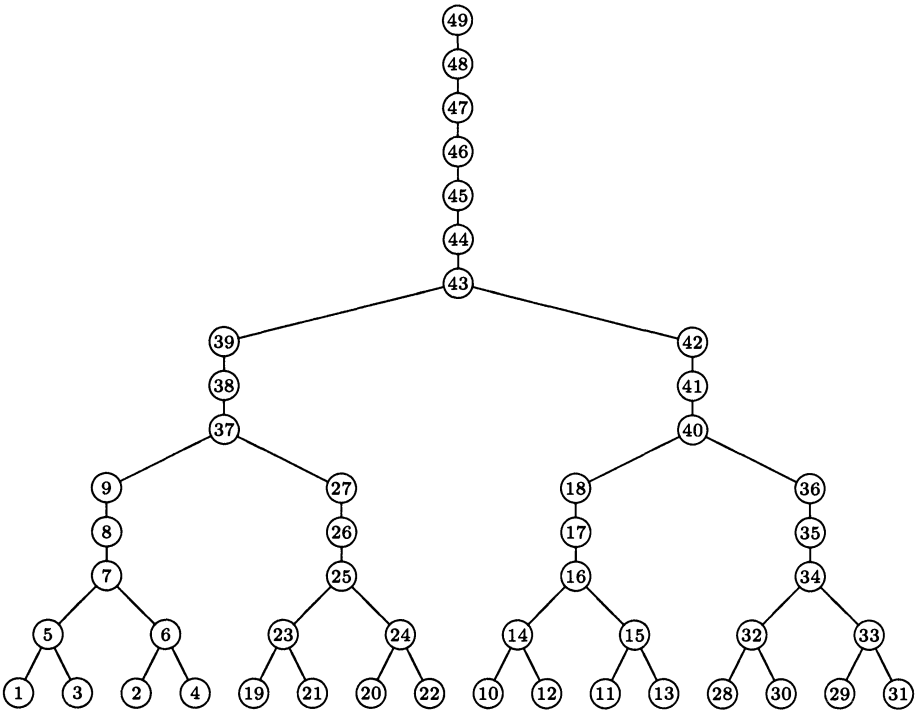


FIG. 2. *The elimination tree associated with a nested dissection ordering on a 7-by-7 grid.*

Lastly, the reduced row merge tree is the tree induced by the interior vertices of the binary row merge tree. The "reduced row merge tree" associated with a nested dissection ordering on a 7-by-7 grid is shown in Fig. 3. Note that each separator S_j^i is represented by its lowest numbered vertex in the reduced row merge tree. Since the leaves of the binary row merge tree are simply data vertices and the interior vertices are readily interpreted as task vertices, the reduced row merge tree is particularly suitable for investigating various task scheduling strategies.

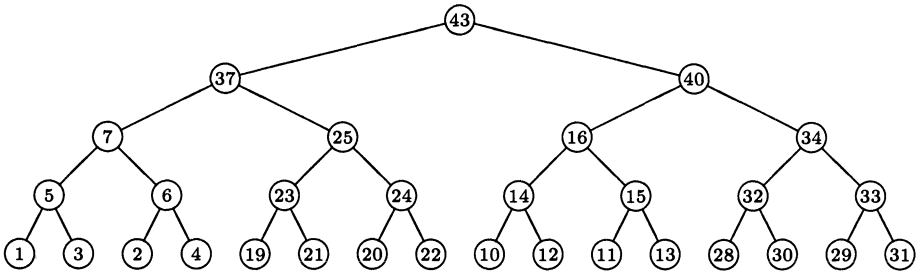


FIG. 3. *The reduced row merge tree associated with a nested dissection ordering on a 7-by-7 grid.*

The name "row merge tree" is based on the following observation. By the definition of the binary row merge tree, the leaves of each subtree rooted at an interior vertex represent a subset of rows from the coefficient matrix A. Since every interior vertex defines a subtree rooted at itself, one can associate with each interior vertex an

upper triangular matrix obtained by the orthogonal reduction of the corresponding rows in its subtree. Clearly the matrix associated with the root of the binary row merge tree is the triangular factor R.

The general row merging scheme assigns to each interior vertex the task of merging two submatrices associated with its two children *after* they are formed. It is desirable to find an ordering to perform these tasks so that the submatrices are always formed before they are needed and they are conveniently accessible whenever they are needed. The postorder traversal of the reduced row merge tree generates such a sequence, because it ensures that the children are always visited before their parent is visited in the traversal.

To generate the row merge tree we note that the nonzero structure of the factor R is available after symbolic factorization of A^TA. Alternatively, the structure of R can be generated directly from A using a symbolic submatrix merging algorithm described in [15], where more details about the properties of the row merge tree and other related work can be found. For our purpose, it is essential to understand how the row merge tree induces a computational sequence for performing Givens rotations in the serial row merging scheme, and what role the row merge tree can play to identify and exploit parallelism in the parallel row merging scheme we propose in this article.

3. Parallel row merging scheme. We now consider implementing the numeric factorization step of the general row merging scheme on a hypercube multiprocessor. The proposed parallel algorithm is designed to perform the numeric factorization on a hypercube machine with $p = 2^d$ processors, where d is the dimension of the hypercube network. We assume that A^TA is irreducible and that the permuted matrix P_rAP_c, and the structure of the triangular factor R are all available in the host.

3.1. Basic mapping considerations. Since there is no globally shared memory among the p processing nodes, or between the host and a node processor, the data must be distributed among the processors in some way, and the mapping strategy should be devised to maintain high parallelism throughout the computation. We exploit the following observations in mapping data and computing tasks to processors.

Observation 1. If all of the rows associated with the leaves of a subtree are assigned to a single processor, no communication is needed for the designated processor to execute the sequential row merging scheme on its local data. The computation is guided by the postorder traversal of the subtree.

Observation 2. The computation associated with each of the p disjoint subtrees can be completed by p processors *independently* and *simultaneously*. The p disjoint subtrees which represent approximately equal amounts of work are easily identified if the row merge tree is a balanced binary tree, and there exist heuristic algorithms to balance [13], [14], [16] and partition [3] an unbalanced binary tree.

Observation 3. Since the tasks corresponding to the subtrees are performed by different processors on local data independently, as far as the parallel algorithm is concerned, the p subtrees (except for the root vertices) may be simply pruned from the row merge tree after they are appropriately identified. By doing so, we obtain a much shorter and smaller tree with p leaves. Associated with the p leaves are the p submatrices resulting from the independent merging operations by the p processors. *The design of our parallel algorithm will focus on how the p processors cooperate to complete the factorization process from this stage on.*

Since the computation is now guided by a row merge tree of p leaves, and the task associated with each interior vertex merges the two submatrices associated with its two children vertices, clearly the number of tasks becomes steadily less than the

number of processors as we traverse the tree to the root. With the dimension and the fill of the submatrices increasing, the merging tasks increase in cost, eventually reaching the same complexity as that of the sequential algorithm. Thus, multiple processors must now be employed to complete each submatrix merging task in order to achieve acceptable parallelism.

3.2. A parallel submatrix merging algorithm. There are two crucial decisions to be made with respect to the implementation of the "one task divided among q processors" strategy, namely, how to map data among the q processors and how to embed an efficient communication topology in the hypercube connection network provided for this subset of q processors. The fundamental step in the general row merging scheme is the merging of QR factorizations of subtrees. Corresponding to each interior vertex of the binary row merge tree is a task that "merges" the two submatrices associated with the two children of that vertex. More specifically, the task computes the QR factorization of the matrix consisting of those nonzero rows from either child's factor R that contain nonzeros only in columns corresponding to the parent vertex and its ancestors in the elimination tree. Barring accidental cancellation, these rows as drawn from either child, with identically zero columns removed, form a dense upper trapezoidal submatrix in the corresponding child's QR factor. In Liu's terminology these submatrices are "essentially full" and dense matrix operations can be used to obtain the required QR factorization. Thus, the general row merging task for vertex c is to compute the dense QR factorization of two dense upper trapezoidal submatrices. The data to be divided among the q processors comprise the two upper trapezoidal submatrices. Each task computes *either* one row of the final upper triangular matrix R *or* $|S_j^i|$ rows of R in the case that the task vertex represents a chain of $|S_j^i|$ vertices in the elimination tree.

We shall first propose a variant of the sequential Givens algorithm, which is more suitable for parallel implementation.

Pairwise Givens rotations. Without loss of generality, we shall present the method by applying it to the merging of two full $n \times n$ upper triangular matrices R and $\tilde{R}$. In this method, we pair the ith row of R and the ith row of $\tilde{R}$ for $1 \leq i \leq n$. We then apply the algorithm below to each pair of rows, resulting in annihilating the first nonzero value in row $\tilde{r}_{i,*}$.

if $|\tilde{r}_{i,i}| \geq |r_{i,i}|$ **then**
  $t \leftarrow |r_{i,i}|/|\tilde{r}_{i,i}|$
  $s \leftarrow 1/\sqrt{1+t^2}$
  $c \leftarrow st$
else
  $t \leftarrow |\tilde{r}_{i,i}|/|r_{i,i}|$
  $c \leftarrow 1/\sqrt{1+t^2}$
  $s \leftarrow ct$
for $j = i, i+1, \cdots, n$ **do**
  $v \leftarrow r_{i,j}$
  $w \leftarrow \tilde{r}_{i,j}$
  $r_{i,j} \leftarrow cv + sw$
  $\tilde{r}_{i,j} \leftarrow -sv + cw$

We now have n updated rows in R and $(n-1)$ updated rows in $\tilde{R}$. (The nth row of $\tilde{R}$

has been eliminated.) The rows with their first nonzero elements in identical positions are again paired together, and we again apply the algorithm above to each pair of rows, where $2 \leq i \leq n$. This is repeated $(n-2)$ more times to eliminate the remaining $(n-2)$ rows from $\tilde{R}$. The arithmetic cost (in terms of multiplicative operations) is the same as the standard Givens method, namely, $(2/3)n^3 + 2n^2 + (4/3)n$.

In the proposed method, the q row merging operations corresponding to the q pairs of rows can be done simultaneously by q available processors. By distributing the rows of the two submatrices over a loop of q processors in a wraparound fashion, and requiring each processor to send the reduced row to its right neighbour after each row merging operation, we obtain a parallel algorithm to perform the submatrix merging operation on a loop of q processors. In [1], it is shown that the basic scheme can be easily adapted to accommodate the case when the two submatrices are of different dimensions.

3.3. Hypercube partitioning. In the previous section we proposed a parallel algorithm to divide a submatrix merging task among a number of processors which form a loop. When the precedence relationship induced by the row merge tree allows us to process several tasks in parallel, it is desirable to partition the available processors into several loops — one for each task. Furthermore, if these tasks have different computational demand, it is also desirable to assign more processors to a bigger task and fewer processors to a smaller task so that the work can be divided evenly among all processors. Therefore, before we can lay out the overall strategy, an important question is "Given an arbitrary number q, does there always exist a subset of q processors in the hypercube machine so that a loop can be embedded?" Although the answer to this question is negative, we have an affirmative answer if q is an even number. With this very mild restriction, we can actually obtain much stronger results which turn out to be very important for the performance of the proposed parallel algorithm. We establish these results in the following theorem.

THEOREM 3.1. *Suppose we are given a hypercube connection network with $p = 2^d$ processors. If it is desirable to partition the set of p processors into k disjoint subsets $S_1, S_2, \cdots$, and S_k such that*

$$p = \sum_{i=1}^{k} |S_i|$$

and

$$|S_i| = 2\ell_i ,$$

where ℓ_i is a positive integer, then there exists a (possibly different) partition which maintains the cardinality of each subset, and permits the embedding of k disjoint loops, one for each subset. For each $\ell_i = 1$ we have a degenerate loop consisting of two processors.

Proof. There is a unique mapping from the d-bit *reflected binary Gray code* [20] to the 2^d processor id's of the given hypercube machine, and it is known that the former coding embeds a loop on the hypercube network. Furthermore, any two processors whose id's are different in one bit, regardless of the bit position, are connected by a direct link in a hypercube network. By the definition of the reflected binary Gray code, if we represent the 2^{d-1} $(d-1)$-bit Gray code by the array

$$G(d-1) = \{G_0, G_1, G_2, \cdots, G_{2^{d-1}-1}\} ,$$

then the 2^d d-bit Gray code can be defined recursively by the following equation.

$$G(d) = \{0G_0, 0G_1, 0G_2, \cdots, 0G_{2^{d-1}-1}, 1G_{2^{d-1}-1}, \cdots, 1G_2, 1G_1, 1G_0\}.$$

Thus any two Gray codes in symmetric positions from the left end and the right end of the array $G(d)$ also differ in one bit only. Therefore, the ℓ_1 processors from the left end of the array and the ℓ_1 processors from the right end of the array form a loop of $2\ell_1$ processors. The $2\ell_2$ processors for S_2 can be chosen from the remaining processors in exactly the same manner, and so on for the $2\ell_i$ processors for the subsets S_i, $3 \leq i \leq k$. This proves the theorem. □

The implication of Theorem 3.1 is in essence that it is not only possible to assign processor loops of different sizes to handle independent tasks which demand different amounts of computation, but it is also feasible to have all of the processor loops operating simultaneously. Using a hypercube of dimension 4, we illustrate three such partitionings in Fig. 4.

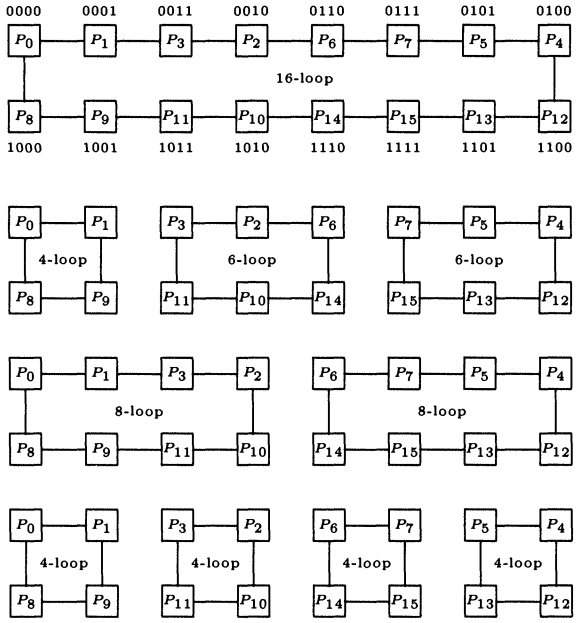


FIG. 4. *Embedding loop(s) in a hypercube of dimension* 4.

4. A mapping example and complexity analysis. Using the reduced row merge tree in Fig. 3 as an example, we show how the ideas presented in the previous sections can be used to divide the computing tasks among the processors in a hypercube network. Suppose we are given a hypercube of dimension 3, i.e., there are eight processors available. Our discussion so far suggests the following mapping. Each of the eight independent subtrees rooted at vertices 5, 6, 23, 24, 14, 15, 32, and 33 will be assigned to one processor. The four tasks associated with vertices 7, 25, 16, and 34 will each be handled by two processors. In the next level, the two tasks associated with vertices 37 and 40 will each be handled by a loop of four processors. Finally the merging of two 7-by-7 full upper triangular matrices, which is the task associated

with the root vertex 43, will be handled by a loop of eight processors. Since the loops assigned to the task vertices at the same level of the row merge tree form a partition of the hypercube network, there is no competition for communication channels among the loops.

We next present analytical complexity results for a k-by-k grid model problem on a hypercube multiprocessor of dimension d. The analysis of the model problem can be greatly simplified by examining the work associated with a branch of the reduced row merge tree, which corresponds to the critical path of the parallel algorithm. For convenience, we shall assume $k = 2^\ell - 1$, where $\ell > 0$. Letting p denote the total number of node processors on the machine, we have $p = 2^d$. Since all of the p processors cooperate to perform the last task of merging two full k-by-k upper triangular matrices, we shall assume $k \gg p$, that is, that k is sufficiently large that we can effectively use p processors in the final step(s).

Recall that each separator is represented by its lowest numbered vertex in the reduced row merge tree. Now, with $p = 2^d$ processors available, the subtrees rooted at separators S_i^d, $1 \le i \le 2^d$, will each be assigned to one processor, and the separators one level above will each be assigned to a loop of two processors, and so on. The last separator, S_1^0, which is the root of the reduced row merge tree, will be assigned a loop of p processors. Therefore, if we let $T(S_i^j, p/2^j)$ denote the time (computation and communication) required by the *parallel* submatrix merging operation associated with one task vertex, and let $T_s(S_i^j)$ denote the time for processing the subtree rooted at separator S_i^j by the serial row merging scheme, the total time required by the parallel row merging scheme to factor the coefficient matrix associated with the k-by-k grid can be expressed as

$$T_{k\times k}(k,p) = T_s\left(S_i^d\right) + \sum_{j=0}^{d-1} T\left(S_i^j, \frac{p}{2^j}\right), \tag{2}$$

where each S_i^j refers to one particular jth level separator which is located on a *highest-cost* path of the *reduced* row merge tree. (There may be several.) Thus the values of i's may not be the same for all S_i^j. To determine a highest-cost path, note that the subgrids produced by the separators may have one more grid line along one or more sides. Such η-by-η subgrids are termed *bordered* subgrids in [7]. Now if we let $\rho(\eta, i, q)$ be the cost (computation and communication) of factoring on q processors the coefficient matrix associated with an η-by-η subgrid which is bordered along i sides, then we have

$$\rho(\eta, j, q) > \rho(\eta, i, q), \quad \text{for every } j > i\,. \tag{3}$$

Therefore, a highest-cost path (which is not unique here) can be defined by the separators in Fig. 5. Applying this to the 7-by-7 grid in Fig. 1, the corresponding branch on its row merge tree is identified by the path labelled in Fig. 6. We can then derive the total cost of the parallel algorithm by setting up the recurrence equations for the subproblems along the highest-cost path.

Since the derivation of the individual cost functions is straightforward but quite tedious, and the complete details of the cost functions as well as the recurrence equations are available in the report [1], we shall present only the solution $\rho(k, 0, p)$ here

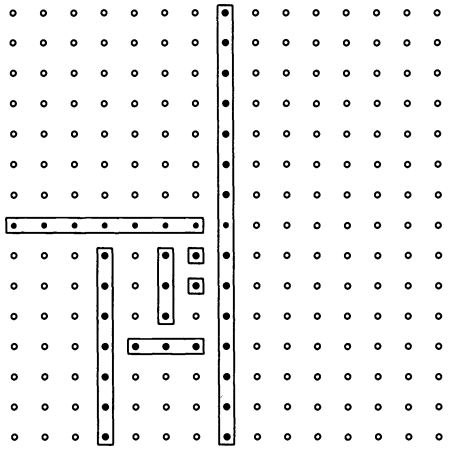


FIG. 5. *Separators along a highest-cost path of a* 15*-by-*15 *grid ordered by nested dissection method.*

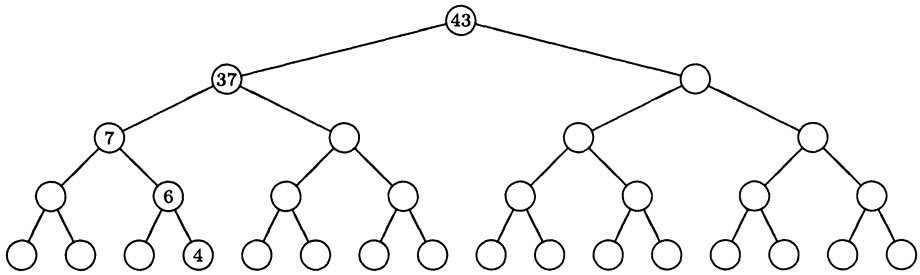


FIG. 6. *A highest-cost branch on a reduced row merge tree associated with a nested dissection ordering of a* 7*-by-*7 *grid.*

in Theorem 4.1, which gives the upper bound of the sum of total computational cost and communication cost. Included in the communication cost are the start-up time, the time for passing data during each submatrix merging operation, the time for data relocation when combining loops, as well as the time for data distribution within each loop before each submatrix operation. In deriving the upper bound, we have made the following assumptions. First, we assume that all processors in a loop send a reduced row to their neighbours simultaneously during the pairwise Givens reduction process, and the communication time is determined by the row with most nonzero elements. Second, we assume the maximum communication path, $\log_2 p$, where p is the total number of processors in the hypercube network, in computing the time for relocating one submatrix when combining two loops of processors for the next merging operation. Third, we observe that each loop of processors form a subcube and that messages can be pipelined in the data distribution phase. The proof of Theorem 4.1 can be found in [1] and is omitted.

THEOREM 4.1. *An upper bound of the total cost for applying the parallel row merging scheme to a k-by-k grid model problem on a hypercube having p processors is*

given by

$$\begin{aligned}\rho(k,0,p) \;<\; & \left(\frac{146}{3}+\frac{146}{12}\alpha-\frac{371}{12\sqrt{p}}\alpha\right)\frac{k^3}{p}+31\frac{k^2\log_2 k}{p}+\frac{521}{48}k^2(\log_2 p)\alpha \\ & +\left(\frac{31}{8}\beta-\frac{1099}{24}\alpha\right)\frac{k^2\log_2 p}{p}-155\frac{k^2}{p}-\frac{17}{2}\frac{k^2}{p}(\alpha+\beta) \\ & -\frac{20}{21}kp+O(k\log_2 p)\,,\end{aligned} \tag{4}$$

where β is the start-up time for sending a message and α is the ratio of the time for transmitting one floating-point number across one link to the time for one floating-point multiplicative operation.

Comparing with the serial cost given by

$$\theta(k,0) \;=\; \frac{829}{21}k^3+\frac{155}{3}k^2\log_2 k-\frac{569}{3}k^2+\frac{488}{3}k-\frac{176}{7}\,, \tag{5}$$

we see that the coefficient of the $O(k^3/p)$ term for the parallel arithmetic cost in $\rho(k,0,p)$ is (146/3), which is slightly larger than the coefficient of (829/21) of the $O(k^3)$ term of the serial arithmetic cost $\theta(k,0)$. They are not exactly the same because the tasks associated with the *critical path* are bigger than the tasks associated with other branches of the row merging tree. The upper bound we obtained in Theorem 4.1 for $\rho(k,0,p)$ indicates that the $O(k^3/p)$ communication cost of the proposed algorithm is of the same order of magnitude as the arithmetic cost. This is undesirable on a machine where the communication cost is not negligible compared to the arithmetic cost. In the next section we examine a generalized version of the parallel submatrix merging algorithm and indicate how it may reduce the communication cost.

5. Generalizing the algorithm. The analysis detailed in [1] indicates that the $O(k^3/p)$ term in the total communication cost has sole contribution from the pairwise Givens reduction process. We first recall that the parallel implementation we proposed in §3.2 requires that the consecutive rows of both matrices be assigned to consecutive processors of the loop, with assignment "wrapping around" to processor 1 after a pair of rows is assigned to processor q. This mapping strategy can be viewed as a special case of a more general *block* wrap-mapping scheme, where each submatrix is divided into blocks of b consecutive rows, and the consecutive blocks are wrap-mapped to the processors in the loop. When the block size $b = 1$ we obtain the parallel pairwise Givens reduction scheme. When the block size $b > 1$, more nonzeros may be annihilated in merging the two blocks before data transmission is needed. The results presented in Theorem 5.1 below show that by choosing a particular block size the communication cost for merging two $n \times n$ full upper triangular matrices using a loop of q processors can be reduced from $O(n^3/q)$ to $O(qn^2)$, and that the leading term of the arithmetic cost remains unchanged. The proof of Theorem 5.1 may be found in [1].

THEOREM 5.1. *Consider merging two $n \times n$ full upper triangular matrices on a loop of q processors using the parallel pairwise block Givens scheme. By choosing the block size $b = n/q^2$, the arithmetic cost $C_b(n,b,q)$ and the communication cost $\phi_b(n,b,q)$ are given by*

$$\begin{aligned}C_b\left(n,\frac{n}{q^2},q\right) \;=\; & \frac{2}{3}\frac{n^3}{q}+2\frac{n^2}{q}-\frac{2}{3}\frac{n^3}{q^3}+6\frac{n^3}{q^4}-10\frac{n^3}{q^5} \\ & +4\frac{n^3}{q^6}-2\frac{n^2}{q^2}+8\frac{n^2}{q^3}-4\frac{n^2}{q^4}\end{aligned} \tag{6}$$

and

$$(7) \qquad \phi_b\left(n, \frac{n}{q^2}, q\right) = \frac{\alpha}{12}\left(2qn^2 - 6n^2 + 3qn + 7\frac{n^2}{q} + 3\frac{n^2}{q^2} - 30\frac{n^2}{q^3} + 24\frac{n^2}{q^4} - 3n + 12\frac{n}{q} - 12\frac{n}{q^2}\right).$$

Comparing $C_b(n, n/q^2, q)$ with $\phi_b(n, n/q^2, q)$, we see that the arithmetic cost dominates the communication cost when $n \gg q$. More specifically, the $O\left(n^3/q\right)$ arithmetic cost and $O\left(qn^2\right)$ communication cost imply that q should be chosen to be less than $\sqrt{n}$ in order to have the arithmetic cost dominate the communication cost. This implication is important because for the k-by-k grid model problem we have analyzed in this section, the last task on the critical path involves merging two $k \times k$ full upper triangular matrices using a loop of p processors.

In [1] it is also shown that the communication cost $\phi_b(n, b, q)$ is b times smaller than that of the parallel pairwise Givens scheme. Since the results in this section are obtained without pipelining the b rows in data transmission, more savings can be expected when data are also pipelined. All of these suggest that the communication cost of the parallel row merging scheme can be reduced by applying the generalized submatrix merging algorithm with appropriate block size to each task. The development of an enhanced parallel row merging scheme by incorporating this idea merits further research.

REFERENCES

[1] E. C. H. CHU, *Orthogonal decomposition of dense and sparse matrices on multiprocessors*, Tech. Rep. CS-88-08 (Ph.D. thesis), Waterloo, Ontario, Canada N2L 3G1, March 1988.

[2] I. S. DUFF, *Pivot selection and row ordering in Givens reduction on sparse matrices*, Computing, 13(1974), pp. 239–248.

[3] G. A. GEIST AND E. NG, *A partitioning strategy for parallel sparse Cholesky factorization*, Tech. Rep. ORNL/TM-10937, Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, TN, 1988.

[4] W. M. GENTLEMAN, *Row elimination for solving sparse linear systems and least squares problems*, Lecture Notes in Mathematics (506), G. A. Watson, ed., pp. 122–133, Springer-Verlag, New York–Berlin–Heidelberg, 1975. (Proc. 1975 Dundee Conference on Numerical Analysis.)

[5] J. A. GEORGE AND M. T. HEATH, *Solution of sparse linear least squares problems using Givens rotations*, Linear Algebra Appl., 34(1980), pp. 69–83.

[6] J. A. GEORGE, M. T. HEATH, AND J. W-H. LIU, *Parallel Cholesky factorization on a shared-memory multiprocessor*, Linear Algebra Appl., 77(1986), pp. 165–187.

[7] J. A. GEORGE AND J. W-H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[8] ———, *Householder reflections versus Givens rotations in sparse orthogonal decomposition*, Linear Algebra Appl., 88–89(1987), pp. 223–238.

[9] J. A. GEORGE, J. W-H. LIU, AND E. G-Y. NG, *Row ordering schemes for sparse Givens transformations: I. Bipartite graph model*, Linear Algebra Appl., 61(1984), pp. 55–81.

[10] ———, *Row ordering schemes for sparse Givens transformations: II. Implicit graph model*, Linear Algebra Appl., 75(1986), pp. 203–223.

[11] ———, *Row ordering schemes for sparse Givens transformations: III. Analysis for a model problem*, Linear Algebra Appl., 75(1986), pp. 225–240.

[12] M. T. HEATH, *Numerical methods for large sparse linear least squares problems*, SIAM J. Sci. Statist. Comput., 5(1984), pp. 497–513.

[13] J. A. G. JESS AND H. G. M. KEES, *A data structure for parallel* L/U *decomposition*, IEEE Trans. Comput., C-31(1982), pp. 231–239.

[14] 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. 1146–1173.

[15] J. W-H. LIU, *On general row merging schemes for sparse Givens transformations*, SIAM J. Sci. Statist. Comput., 7(1986), pp. 1190–1211.

[16] ———, *Reordering sparse matrices for parallel elimination*, Tech. Rep. CS-87-01, Dept. of Computer Science, York University, Ontario, Canada M3J 1P3, 1987.

[17] ———, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11(1990), pp. 134–172.

[18] O. OSTERBY AND Z. ZLATEV, *Direct methods for sparse matrices*, Lecture Notes in Computer Science 157, Springer-Verlag, Berlin, 1983.

[19] G. OSTROUCHOV, *Symbolic Givens reduction and row-ordering in large sparse least squares problems*, SIAM J. Sci. Statist. Comput., 8(1987), pp. 248–264.

[20] E. M. REINGOLD, J. NIEVERGELT, AND N. DEO, *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[21] R. SCHREIBER, *A new implementation of sparse Gaussian elimination*, ACM Trans. Math. Software, 8(1982), pp. 256–276.

[22] M. YANNAKAKIS, *Computing the minimum fill-in is* NP*-complete*, SIAM J. Algebraic Discrete Methods, 2(1981), pp. 77–79.

[23] Z. ZLATEV, *Comparison of two pivotal strategies in sparse plane rotations*, Comput. Math. Appl., 8(1982), pp. 119–135.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 466-480, July 1990

009

ROBUST REGRESSION COMPUTATION USING ITERATIVELY REWEIGHTED LEAST SQUARES*

DIANNE P. O'LEARY†

Abstract. Several variants of Newton's method are used to obtain estimates of solution vectors and residual vectors for the linear model $Ax = b + e = b_{true}$ using an iteratively reweighted least squares criterion, which tends to diminish the influence of outliers compared with the standard least squares criterion. Algorithms appropriate for dense and sparse matrices are presented. Solving Newton's linear system using updated matrix factorizations or the (unpreconditioned) conjugate gradient iteration gives the most effective algorithms. Four weighting functions are compared, and results are given for sparse well-conditioned and ill-conditioned problems.

Key words. iteratively reweighted least squares, robust regression

AMS(MOS) subject classifications. 62J05, 65F20

1. Introduction.

Consider the linear model

$$Ax = b + e = b_{true},$$

where A, the model matrix, has dimension $m \times n$; b is the vector of observations; b_{true} is the unknown vector of true values; e is the unknown vector of observation errors; and x is the unknown vector of parameters. For a given vector x, we define the residual vector $r(x) = b - Ax$.

We discuss in this paper various algorithms for obtaining estimates of the solution vector $\hat{x}$, the residual vector $r(\hat{x})$, and the norm of the residual vector using the iteratively reweighted least squares criterion: i.e., we wish to solve the problem

$$\min_x \sum_{i=1}^{m} \rho(r_i(x)), \tag{1}$$

where ρ is a given function. For a discussion of the statistical properties of this type of regression, see, for example [19]. Taking $\rho(z) = z^2/2$ gives the ordinary linear least squares problem. In order to reduce the influence of outliers, other functions have been proposed, and we consider in this paper four such functions, each twice continuously differentiable almost everywhere, with nonnegative second derivative wherever it is defined. Huber [18] used

$$\rho(z) = \begin{cases} z^2/2, & |z| \le \beta, \\ \beta|z| - \beta^2/2, & |z| > \beta, \end{cases}$$

where β is a problem-dependent parameter. Dutter [11] gives a safeguarded algorithm that overcomes degenerate cases. Minimizing Huber's function leads to a quadratic programming problem, and it is possible to develop finitely terminating algorithms as in the work of Clark and Osborne [3]. The logistic function [4] is

$$\rho(z) = \beta^2 \log(\cosh(z/\beta)).$$

* Received by the editors August 9, 1989; accepted for publication (in revised form) December 8, 1989. This work was supported by the Air Force Office of Scientific Research under grant 87-0158.

† Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland 20742 (oleary@cs.umd.edu).

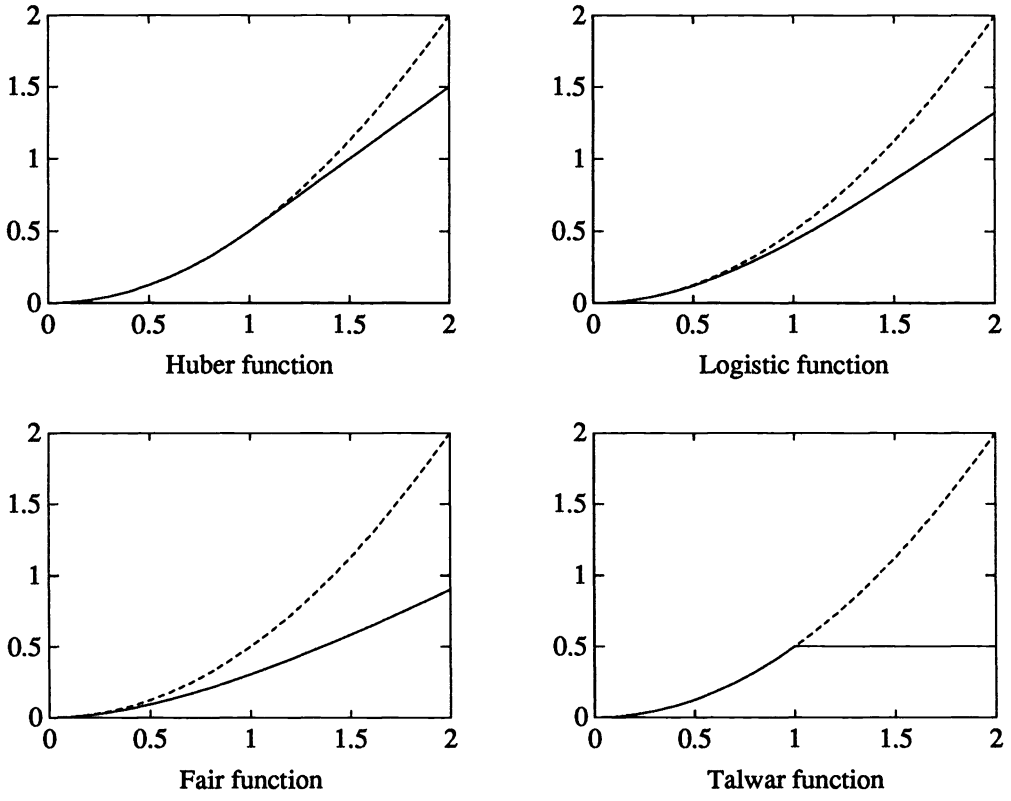


FIG. 1. *The weighting functions (solid lines) with the standard least squares function (dotted lines) as reference. The constant β is set to* 1.

Fair [14] proposed the function

$$\rho(z) = \beta^2(|z|/\beta - \log(1 + |z|/\beta)).$$

Huber [18] also proposed the function

$$\rho(z) = \begin{cases} z^2/2, & |z| \le \beta, \\ \beta^2/2, & |z| > \beta, \end{cases}$$

which is given the name Talwar [16] in [4]. Graphs of these functions are given in Fig. 1. There have been other proposals to use other convex and nonconvex weighting functions, but the methods we discuss may have stability problems for nonpositive second derivatives.

The subject of this paper is the comparison of some algorithms for solving the iteratively reweighted least squares problem, a comparison of the performance of various weighting functions, and a discussion of the interaction between the weighting and the conditioning in the matrix A.

Robust regression through the use of functions related to least squares has been the subject of intense research, and we note only selected references here. Dempster, Laird, and Rubin [6] discuss statistical properties of the estimates under the assumption that the observation errors are independent normal. Coleman et al. [4] developed a high quality set of routines to compute robust estimators for eight weight functions, including the four discussed in this paper.

We take the viewpoint that the variance of the observation errors is known, at least approximately, and thus the scale is fixed. This is appropriate in some but not all applications, and it is possible to obtain simultaneous estimates of the scale factors and the solution; see, for example, Shanno and Rocke [25] and Ekblom [13].

The algorithms are presented in §2, and results are discussed in §3 and summarized in §4.

A similar computational problem (see (2)) arises in the core step in algorithms like that of Karmarkar for solving linear programming problems [20], and the algorithms in this paper have application there as well.

2. The algorithms. We will use Newton-like methods to solve our problem. Our first observation is that although we are minimizing over x-space, it is easier to work in the appropriate subspace of r-space. To establish some notation, we express (1) as

$$\min_x \hat{f}(x) \equiv \min_x f(r(x)) \equiv \min_x \sum_{i=1}^{m} \rho(r_i(x)).$$

Let y be the gradient vector for $f(r)$:

$$y_i = \rho'(r_i),$$

and let $D(r)$ be a diagonal matrix with entries

$$d_{ii} = \rho''(r_i).$$

Now, the function $\hat{f}(x)$ has a gradient $\hat{g}$ and Hessian matrix of second derivatives $\hat{H}$ defined by

$$\hat{g} = -A^T y, \qquad \hat{H} = A^T D A,$$

and $\hat{H}$ is positive semidefinite if ρ'' is nonnegative.

The step direction for Newton's method for minimizing $\hat{f}(x)$ is $\hat{s} = -\hat{H}^{-1}\hat{g}$, and a change of $\hat{s}$ in the x variables will create a change in the residual of $s = -A\hat{s}$, or

$$s = -A(A^T D A)^{-1} A^T y. \tag{2}$$

Since we need to assess the progress of Newton's method by evaluating the function ρ at each element of the residual, the computation is more conveniently done without the x variables. Further, determining the search direction for the x variables involves a computation whose conditioning is related to that of A, but, as we demonstrate below (see Algorithm 2), the conditioning for the problem of determining the search direction in r depends on $Q^T D Q$, where the columns of Q form an orthonormal basis for the range of A.

The general method is as follows:

Given an initial x, compute an initial $r = b - Ax$.

Repeat until convergence:

1. Compute the search direction $s = -A(A^T D(r) A)^{-1} A^T y$.
2. Perform a linesearch to determine a value α for which $f(r + \alpha s)$ is sufficiently less than $f(r)$.

Upon convergence to a residual vector r_{opt}, compute the corresponding x_{opt} by solving the consistent linear system $Ax = b - r_{opt}$.

Due to round-off errors, the system $Ax = b - r_{opt}$ may fail to be consistent, and the norm of the residual from this system is a good diagnostic.

If fewer than n residuals are below the cut-off value β for the function ρ, then the Hessian matrix may be rank deficient. To prevent this from occurring at initial stages of the iteration, where we may be far from the optimal solution, we gradually decrease the cut-off value from a very large number to the desired value over the first four steps of the iteration. This has the effect of starting the iteration from the least squares solution.

Developing efficient and reliable linesearch algorithms is not an easy task, but one such algorithm, due to Jorge J. Moré and David J. Thuente, is CVSRCH in the MINPACK collection of routines. It uses function and gradient values. Since the value $\alpha = 1$ is almost always a good choice, we use that for the initial guess and use coarse tolerances (.1) for convergence in x, the function value, and the gradient value.

We now focus attention on the strategies for computing the search direction. Our basic tool is the QR factorization of an $m \times n$ matrix into the product of an $m \times n$ matrix Q with orthogonal columns, and an $n \times n$ upper triangular matrix R (see, for example, [7]).

Algorithm 1. If ρ'' is nonnegative, then the matrix D has nonnegative elements, and we may factor the matrix $D^{1/2}A$ as $\hat{Q}\hat{R}$. The definition (2) of s then becomes

$$s = -D^{-1/2}\hat{Q}\hat{R}(\hat{R}^T\hat{Q}^T\hat{Q}\hat{R})^{-1}\hat{R}^T\hat{Q}^TD^{-1/2}y = -D^{-1/2}\hat{Q}\hat{Q}^TD^{-1/2}y.$$

Dutter [10] uses this formulation in his "HV algorithm" for the Huber function.

Algorithm 2 (QR Newton). The first algorithm requires a QR factorization of an $m \times n$ matrix at each iteration. To avoid this, we could factor $A = QR$, which yields

$$s = -QR(R^TQ^TDQR)^{-1}R^TQ^Ty = -Q(Q^TDQ)^{-1}Q^Ty.$$

Each iteration is accomplished using the Cholesky factors of the symmetric $n \times n$ matrix $B = Q^TDQ$.

Algorithm 3 ($\bar{B}$ Newton). We can express the matrix B as

$$B = Q^TDQ = \sum_{i=1}^{m} d_{ii}\bar{q}_i\bar{q}_i^T,$$

where $\bar{q}_i^T$ is the ith row of Q. Only the elements d_{ii} change from iteration to iteration, and as the algorithm converges, we can expect many terms in the summation to remain relatively constant. Thus a reasonable way to reduce the computational work is to monitor B and perform rank-one updates to the Cholesky factors only when the change in some component $d_{ii}\bar{q}_i^T\bar{q}_i$ is large compared to the size of B. One way to measure this is to test whether $\bar{q}_i^T\bar{q}_i$ times the change in d_{ii} is greater than some tolerance times the norm of the matrix that we have factored. If so, a rank-one update (or downdate) to the factorization can be performed using standard algorithms implemented, for example, in LINPACK [7]. Ekblom [13] also used the update idea, but worked with A^TDA rather than with Q^TDQ.

Since B is not necessarily fully updated, the computed search direction is not necessarily the true Newton direction but is some approximation to it.

TABLE 1

Costs per iteration of the various algorithms. Not included in the table are costs common to all algorithms: the function evaluations in the line search (m ρ evaluations each) and the Hessian evaluation (m ρ'' evaluations). "Qmult." means multiplication of Q (or Q^T) times a vector. "Solve" means solution of a linear system using Cholesky factors.

Algorithm	Work per iteration	Operations counts (full matrix)
1. First Newton	QR fact. and 2 Qmults.	$mn^2 - 1/3n^3 + 2mn + O(n^2)$
2. QR Newton	Form and factor $Q^T DQ$, 1 solve, and 2 Qmults.	$m(n^2+n)/2 + n^3/6 + 2mn + n^2$
3. $\bar{B}$ Newton	k updates to B factors, 1 solve, and 2 Qmults.	$(1.75k+1)n^2 + 2mn$
4. PCG Newton	k updates to B factors, l pcg itns., and 2 Qmults.	$1.75kn^2 + 2(l+1)mn + ln^2 + 5nl$
5. CG Newton	2 Qmults. and l cg itns.	$2(l+1)mn + 5nl$

Algorithm 4 (PCG Newton). In Algorithm 3, we established a distinction between a matrix $\bar{B}$ for which we have Cholesky factors and the current true matrix B, and we settled for an approximation to the Newton search direction rather than fully updating $\bar{B}$. We can, however, compute the Newton direction quite efficiently by using the preconditioned conjugate gradient algorithm to solve the linear system $Bw = Q^T y$ with $\bar{B}$ as the preconditioner. Decreasing the number of matrix updates increases the number of conjugate gradient iterations.

This algorithm is related to the truncated Newton method [5], [22].

Algorithm 5 (CG Newton). For the particular weighting functions we are using, the matrix D has a very special form. For the Huber and the Talwar functions, each diagonal entry is 1 for residuals with magnitude less than β, and 0 for the outlying residuals. The logistic and Fair functions have diagonal entries that fall quickly from 1 to 0 as the residual increases from 0. We notice that B is a multiple of the identity matrix whenever D is, and thus for practical problems B may differ from the identity by a matrix of small rank, where the rank is equal to the number of outliers, plus a matrix of small norm (for the logistic and Fair functions). Thus we also consider computing the Newton direction using the conjugate gradient algorithm with no preconditioning. This algorithm is particularly well suited for large sparse problems, since only the matrix Q and the diagonal matrix D are required. Conjugate gradients have been used by Scales, Gersztenkorn, and Treitel [24] with $\rho(z) = |z|^p$ (p less than one), but they solved linear systems involving $A^T DA$ rather than $Q^T DQ$.

Table 1 presents the costs associated with an iteration of each of the five algo-

rithms. The number of floating point additions and multiplications are tabulated, assuming that the matrix is dense and that updates to Cholesky factors result from increases in diagonal elements (at a cost of $1.5n^2$ operations) as often as decreases ($2n^2$ operations). From these numbers we see that function and Hessian evaluations have a negligible cost compared to the linear algebra overhead of an iteration.

For sparse matrices, working with the original matrix A rather than the factor Q would better preserve sparsity but, as we will see later, the linear system expressed in terms of Q requires no preconditioning in the conjugate gradient algorithm, and this is a substantial savings. There has been some work in reorderings of A that produce a sparse representation of Q (see, for example, Tewarson [27], Chen and Tewarson [2], and Duff [8]), but most of this work has been directed toward maintaining sparsity in Q and R simultaneously. The sparsity of R is not essential to the algorithms considered here.

A compromise between sparsity and ease of solution of systems $Q^T DQ$ can be achieved by performing an LU factorization of A rather than a QR. Peters and Wilkinson suggested the use of this factorization for standard least squares problems, and Björck and Duff [1] studied its implementation for sparse matrices. All of the algorithms above can be rewritten for this factorization, substituting L for Q, and U for R. Since it has been observed that L is usually well conditioned, even for ill-conditioned A, there is hope that solving systems involving $L^T DL$ will be substantially easier than solving those involving the original matrix $A^T DA$. Computational experience is reported in §3.3.

3. Results.

3.1. A note on perturbations. The solution x^* of an iteratively reweighted least squares problem is characterized by the gradient of $\hat{f}(x^*)$ being zero. The weight functions ρ are designed to diminish the effects on x^* of outliers in the observations b, but how is x^* affected by small perturbations in b? A simple first-order perturbation analysis will yield insight.

The gradient of $\hat{f}(x^*)$ is $-A^T\rho'(b - Ax^*) = 0$. If b is changed to $b + \Delta b$, then the solution will be changed to $x^* + \Delta x$, where

$$-A^T\rho'\left(b + \Delta b - A(x^* + \Delta x)\right) = 0.$$

We expand this to first-order terms as

$$A^T\rho'(b+\Delta b-A(x^*+\Delta x)) \approx A^T(\rho'(b-Ax^*)+\rho''(b-Ax^*)(\Delta b-A\Delta x)) = A^T D(\Delta b-A\Delta x),$$

and Δx is a vector that makes this equal to zero. Thus,

$$A^T DA\Delta x \approx A^T D\Delta b,$$

or, Δx is defined by $\Delta x \approx A_D^\dagger \Delta b$, where $A_D^\dagger = (A^T DA)^{-1}A^T D$ is a weighted pseudo-inverse of A. We now have the conclusion that

$$||\Delta x||_2 \lesssim ||A_D^\dagger||_2 ||\Delta b||_2. \tag{3}$$

Unfortunately, the D in this expression is evaluated at the unknown solution r^*, but results in [26] and [21] guarantee that if D is positive semidefinite and if Q is a matrix whose columns form an orthonormal basis for the range of A, and if $\zeta \leq 1$ is the

smallest of the nonzero singular values of all matrices formed from nonempty subsets of rows of Q, then

$$||A_D^\dagger||_2 \leq \zeta^{-1}||A_I^\dagger||_2.$$

Thus the change (3) in x can be bounded by the change in b magnified by a factor dependent only on the matrix A.

These expressions suggest that for ill-conditioned matrices A the weighting functions will not overcome the sensitivity of the solution to small perturbations in the observations, and this will be illustrated by the numerical results.

3.2. The test problems. There is a large number of small least squares test problems in the literature (see, for example, the previously cited references) but a very small number of large ones in the Harwell-Boeing test set [9]. This makes parametric studies difficult. Shanno and Rocke [25] and others use randomly generated problems, but such problems tend to be very well conditioned [12].

The following procedure was used to generate test problems with varying conditioning and varying number of outliers.

The $m \times n$ matrix A was constructed as the product of three matrices C, E, and F. The matrix C had the same dimensions as A and had $\mu m/2$ nonzeros in each column, each sampled from a normal probability distribution $N(0,1)$. The positions for the nonzeros were chosen randomly from a uniform distribution. F was a square matrix with diagonal entries chosen to be two times $N(0,1)$ samples and with one off-diagonal $N(0,1)$ entry (except in row n) in a random position. E was a diagonal matrix with entries between 1 and $1/\kappa$ (equally spaced on log scale). The product $A = CEF$ has approximately μm nonzeros per column (i.e., "density" μ) and its singular values usually have separations proportional to those of E.

The true solution vector was taken to be z, the vector of all ones, and the right-hand side was chosen to be $b = Az + \sigma N(0,1)$, except that outliers were generated by adding $100\sigma N(0,1)$ to n_{out} randomly chosen elements of b. In all cases, σ was taken to be .01.

Thus, the test problems have five parameters: m, n, μ, κ, and n_{out}.

Computations used double precision arithmetic on a Sun-3 machine. Convergence was declared when the change in the function value was less than 10^{-5}. This test is not suitable in general, but because of the uniform scaling of our problems it is sufficient for our purposes. See [4] for a better termination criterion.

The termination test for the conjugate gradient iterations was that the residual norm be less than 10^{-8} times the norm of the right-hand side, forcing a rather accurate solution to the linear systems.

We investigated several questions, some related to the algorithms and some related to the performance of the various weighting functions. Since Algorithm 1 is not as stable as Algorithm 2 and failed to find a full-rank Hessian matrix quite often in the experiments, we do not present data on its performance.

3.3. How well do the algorithms perform? How does the convergence rate depend on the test problem parameters? As shown in Tables 2 and 3, there seems to be no trend to increased work as the condition number of the problem increases or as the number of outliers increases. As the number of outliers increases, however, there is an increased tendency for the algorithms to fail to find a full rank Hessian matrix. Updating $\bar{B}$ less frequently usually increased the number of function and Hessian evaluations. But a factor of 10 fewer updates, costing $O(n^2)$ each, at worst

TABLE 2

Results of varying condition number. 500×100 *matrix, density* $\mu = .1$, $n_{out} = 10$ *outliers, constant* $\beta = 2.5\sigma$. *Table entries: number of function evaluations, number of Hessian evaluations, number of* cg *iterations, number of Cholesky updates for algorithms with few or frequent updates to the factors.*

	QR	$\bar{B}$	PCG	CG	CG–LU
$\kappa = 6$					
Fair, few updt.	16, 9	39,20,0, 100	16, 9, 76, 100	16, 9, 76	16, 9,362
Fair, freq. updt.		20,10,0,1184	16, 9, 32,1071		
Talwar, few updt.	19, 9	39,20,0, 100	24,11,186, 100	24,11,186	25,10,520
Talwar, freq. updt.		26,11,0, 788	18, 9, 50, 707		
$\kappa = 175$					
Fair, few updt.	16, 9	39,20,0, 100	16, 9, 76, 100	16, 9, 76	16, 9,414
Fair, freq. updt.		20,10,0,1184	16, 9, 32,1071		
Talwar, few updt.	67,10	75,20,0, 100	23,11,186, 100	23,11,186	19,10,553
Talwar, freq. updt.		62,11,0, 788	18,10, 53, 709		
$\kappa = 14576$					
Fair, few updt.	17, 9	40,20,0, 100	16, 9, 76, 100	16, 9, 76	16, 9,432
Fair, freq. updt.		21,10,0,1184	16, 9, 32,1071		
Talwar, few updt.	29,11	16, 5,0, 100	60,10,179, 100	60,10,179	23,10,568
Talwar, freq. updt.		31,11,0, 786	56,10, 55, 711		

TABLE 3

Results of varying number of outliers. 100×20 *matrices, density* $\mu = .1$, *well-conditioned problems* (E =*identity matrix*), *constant* $\beta = 2.5\sigma$. *Table gives number of function evaluations, number of Hessian evaluations, number of* cg *iterations, and number of Cholesky updates.*

Outliers True	Outliers Est.		QR	$\bar{B}$	PCG	CG
0	0	Huber	9,5	9,5,0, 20	13, 5, 0, 20	13, 5, 0
0	0	Logistic	9,5	9,5,0, 58	9, 5, 2, 58	9, 5, 3
0	0	Fair	8,5	8,5,0,106	13, 5, 3,106	13, 5, 5
0	0	Talwar	9,5	9,5,0, 20	13, 5, 0, 20	13, 5, 0
10	–	Huber	fail	fail	fail	fail
10	10	Logistic	19,9	17,9,0,295	19, 9, 14,291	19, 9, 54
10	13	Fair	16,8	16,8,0,314	16, 8, 12,309	16, 8, 39
10	10	Talwar	21,9	22,9,0,170	42, 9, 5,170	46,13,111
20	–	Huber	fail	fail	fail	fail
20	–	Logistic	fail	fail	fail	fail
20	28	Fair	15,8	16,8,0,330	15, 8, 11,335	15, 8, 45
20	10	Talwar	fail	40,9,0,120	47,20,110,190	25,11,104
30	–	Huber	fail	fail	fail	fail
30	–	Logistic	fail	fail	fail	fail
30	48	Fair	18,9	18,9,0,347	18, 9, 15,348	18, 9, 62
30	88-90	Talwar	fail	13,6,0,114	28, 7, 9,118	22, 7, 27

TABLE 4

Results of varying update parameter for factors. 100×20 *well-conditioned matrix, density* $\mu = .5$, $n_{out} = 10$ *outliers. Table entries: number of function evaluations, number of Hessian evaluations, number of* cg *iterations, and number of Cholesky updates.*

Update tolerance		$\bar{B}$	PCG
0.001	Huber	10, 5,0, 63	fail
	Logistic	23,10,0,222	22,10,59,215
	Fair	16, 8,0,227	15, 8,16,231
	Talwar	10, 5,0, 63	35, 8, 5, 92
0.010	Huber	10, 5,0, 41	fail
	Logistic	22, 9,0, 64	22,10,43, 66
	Fair	18, 9,0, 81	15, 8,37, 72
	Talwar	10, 5,0, 41	34, 8,23, 56
0.100	Huber	10, 5,0, 20	fail
	Logistic	26,13,0, 20	22,10,59, 20
	Fair	33,17,0, 20	15, 8,48, 20
	Talwar	10, 5,0, 20	33, 8,38, 20

TABLE 5

Variability of results over a set of 10 *well-conditioned problems.* 100×20, *density* $\mu = .25$, $n_{out} = 10$ *outliers.* $\beta = 10\sigma$, *update param* $= .001$. *Table entries: range and average number of function evaluations, number of Hessian evaluations, and number of Cholesky updates for Algorithm* 3, *the* $\bar{B}$ *Newton method. (Results for Huber exclude one problem that produced failure.)*

	Function evaluations		Hessian evaluations		Updates	
Huber	7-39	ave. 22	5-8	ave. 7	21- 45	ave. 35
Logistic	12-16	ave. 14	7-9	ave. 8	54-170	ave. 92
Fair	12-16	ave. 13	7-8	ave. 8	128-222	ave. 171
Talwar	4-39	ave. 20	5-8	ave. 7	21- 83	ave. 40

doubled the number of function and Hessian evaluations, that cost $O(m)$, resulting in a faster algorithm.

The last two columns of Table 2 show the results of using the conjugate gradient algorithms with no preconditioning. The "CG" data results from use of the QR factors, while the "CG–LU" data involved the LU factors. The use of the LU factors required between 2.8 and 5.6 times as many conjugate gradient iterations, but there was no trend to increased work as the condition number of A was increased. Use of the original matrix A would have shown an increase in the number of iterations as the condition number grew. Each use of conjugate gradients for the LU factors took on average 40-50 iterations, while the theoretical maximum is $n = 100$. Using the LU factors with conjugate gradients, with or without a preconditioning matrix, seems to be a good approach for sparse matrices if the resulting Q would be too dense.

Table 4 shows further results of performing fewer updates to the approximate Hessian. On this problem of size 100×20, the matrix was never updated if the update tolerance was set greater than or equal to 0.100, and there was very little penalty in the number of function or gradient evaluations for either the $\bar{B}$ Newton (Algorithm 3) or the preconditioned conjugate gradient (Algorithm 4) methods.

Table 5 shows the variability of the computational work for a set of 10 random problems with the same test parameters.

3.4. Which functions perform better? How does ill-conditioning affect the performance of the functions? Figure 2 shows graphs of the solutions and residuals produced for well-conditioned problems by ordinary least squares and by the different ρ functions considered in this paper. A well-conditioned matrix of dimension 100×20 was generated, and 10 sample right-hand sides were generated by adding random noise to Az, using 10 different sets of outliers. The Huber and the Talwar functions each produced a solution vector bigger than 10^4 on one of the right-hand sides, and those runs were disregarded. Each of the weighting functions produces a solution vector closer to the unperturbed vector of ones than ordinary least squares, but the corresponding residual vectors are slightly larger.

Figure 3 shows the errors in the solution vector for a sequence of increasingly more ill-conditioned problems with 10 outliers. The residual norm for least squares was 5.00, whereas that for the Fair function was 5.74; neglecting the 10 largest components of the residual, the norm for least squares was 2.48 whereas that for the Fair function was 0.21. The norm of the error in the x vector was also at least ten times smaller in all cases using the Fair function. For a problem with condition number 175, least squares gave an error of 11.5, compared with the true solution of norm 10.0, so the computed solution vector had little resemblance to the true solution. Both least squares and the Fair function were unable to recover the x vector for the most ill-conditioned problem. This is predicted by the perturbation results in §3.1.

3.5. How do the algorithms perform on "real" problems? Experiments were also run using the housing price equation and the 506 observations of Boston census tracts discussed in [15]. This model expresses the median value of homes in each tract as a combination of 14 factors (crime rate, zoning statistics, average number of rooms in homes, accessibility to radial highways, etc.). The model was used without the scaling discussed in [15], and the integer parts of the singular values were 10,128, 672, 632, 272, 197, 156, 84, 74, 10, 8, 6, 5, 2, and 1. The right-hand side elements were around 10, and σ was estimated as 0.1. The constant β was taken to be 2.5σ. The four functions found between 59 and 61 outliers, using a solution vector of size approximately 10. The x vectors from the Huber, Fair, and Logistic

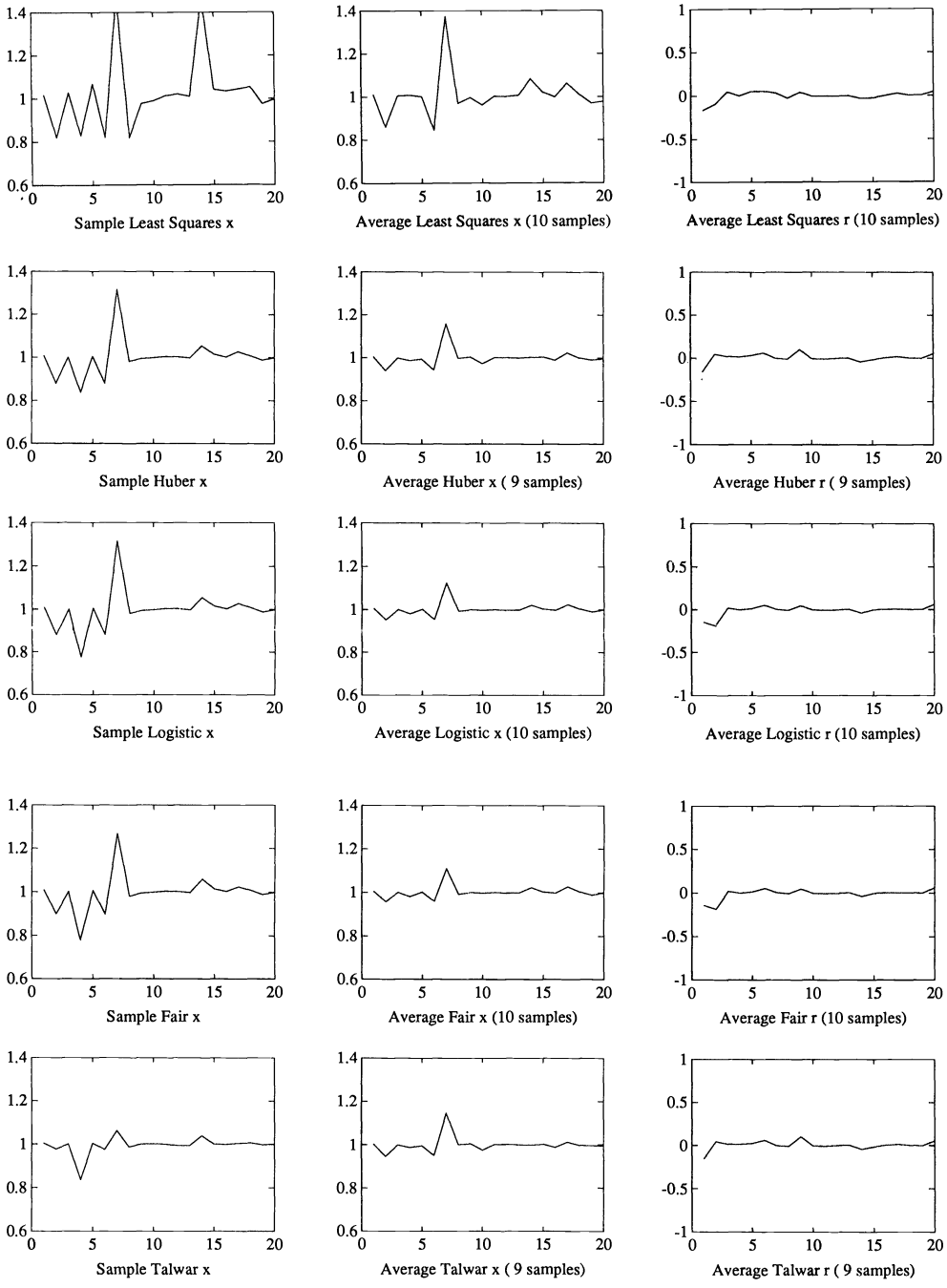


FIG. 2. *Solution vectors for one problem and average solution and residual vectors for* 10 *problems,* 100×20, *density* $\mu = .1$, *well conditioned,* 10 *outliers.*

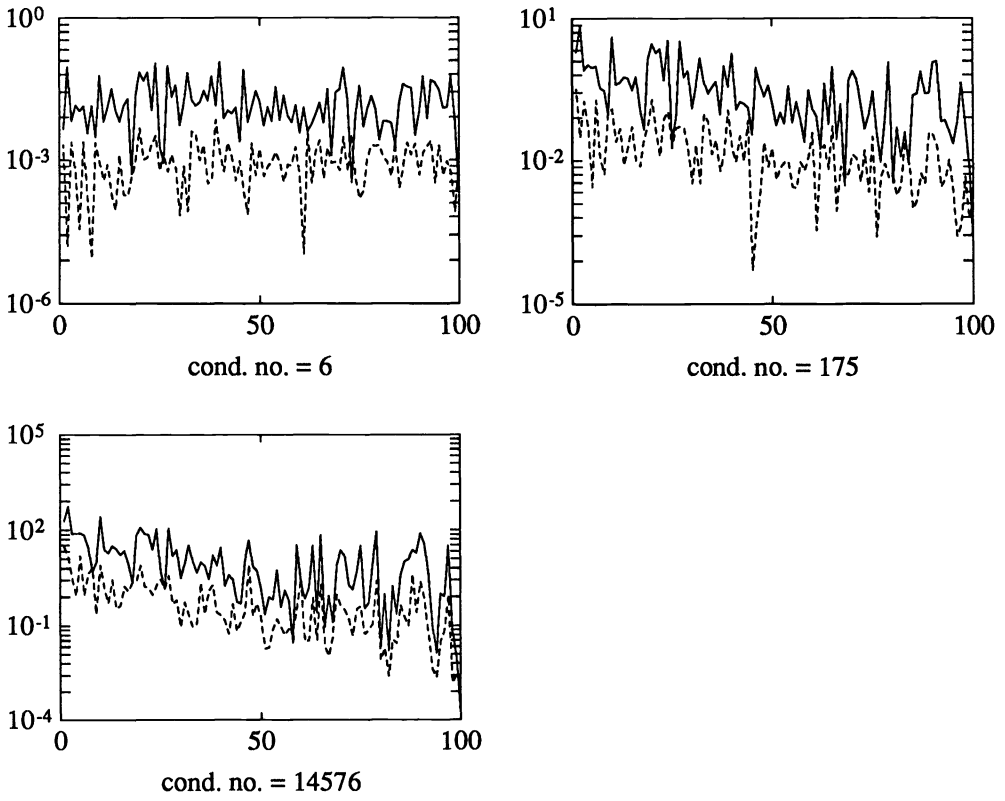


FIG. 3. *The absolute error in each component of the x vector for ordinary least squares (solid) vs. the Fair function (dashed) ($\beta = 2.5\sigma$) for three problems of dimension* 500×100 *with density* $\mu = .1$ *and* 10 *outliers. The residual norm for each problem was* 5.00 *for least squares and* 5.74 *for the Fair function. Neglecting the* 10 *largest components of the residual, the norm was* 2.48 *for least squares and* 0.21 *for the Fair function.*

functions had infinity norm differences of at most .04; the Talwar vector differed from the Fair function by .25. The residual norms were 4.096, 4.086, and 4.088 for the first three functions and 4.650 for Talwar. The CG Newton algorithm took 27 function evaluations, 7 Hessian evaluations, and 17 cg iterations for the Huber function, and 10 function evaluations, 7 Hessian evaluations, and 16-18 cg iterations for the Logistic and the Fair functions.

4. Conclusions. (1) Quadratic programming algorithms should be used for functions such as those of Huber and Talwar, but the best algorithms for the other functions are the $\bar{B}$ Newton algorithm if the problem is not too large and the CG Newton algorithm (with QR or LU factorization) for larger problems.

(2) The functions considered here give better solution vectors than ordinary least squares, but even so, the elements of the solution vector are often heavily contaminated with error if the product of the matrix condition number and the standard deviation of the errors in the data is greater than one.

(3) The number of iterations for the Newton-type algorithms seems insensitive to the conditioning of the matrix and to the number of outliers in the data.

(4) The algorithm for generating sparse test problems with varying conditioning may be useful elsewhere.

(5) The development of parallel algorithms for this class of problems is the subject

of current research. For these Newton-like algorithms, we need a parallel algorithm for determining the search direction and a parallel linesearch algorithm. Parallel versions of the conjugate gradient algorithm [23] are promising candidates for computing the direction.

Acknowledgments. Virginia Klema and Beth Ducot kindly provided the data for the housing model, and Gene Golub provided several of the references. Bob Plemmons made helpful comments on the manuscript.

REFERENCES

[1] Å. 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.
[2] Y. T. CHEN AND R. P. TEWARSON, *On the fill-in when sparse vectors are orthonormalized*, Computing, 9 (1972), pp. 53–56.
[3] D. I. CLARK AND M. R. OSBORNE, *Finite algorithms for Huber's M-estimator*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 72–85.
[4] D. COLEMAN, P. HOLLAND, N. KADEN, AND V. KLEMA, *A system of subroutines for iteratively reweighted least squares computations*, ACM Trans. Math. Software, 6 (1980), pp. 327–336.
[5] R. S. DEMBO AND T. STEIHAUG, *Truncated-Newton algorithms for large-scale unconstrained optimization*, Math. Programming, 26 (1983), pp. 190–212.
[6] A. P. DEMPSTER, N. M. LAIRD, AND D. B. RUBIN, *Iteratively reweighted least squares for linear regression when errors are normal/independent distributed*, in Multivariate Analysis V, P. R. Krishnaiah, ed., North-Holland, New York, 1980, pp. 35–57.
[7] J. J. DONGARRA, C. B. MOLER, J. R. BUNCH, AND G. W. STEWART, LINPACK *Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.
[8] I. S. DUFF, *Pivot selection and row ordering in Givens reduction on sparse matrices*, Computing, 13 (1974), pp. 239–248.
[9] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[10] R. DUTTER, *Robust regression: Different approaches to numerical solutions and algorithms*, Tech. Report Research Report No. 6, Fachgruppe fuer Statistik, ETH, Zurich, 1975.
[11] ———, *Algorithms for the Huber estimator in multiple regression*, Computing, 18 (1977), pp. 167–176.
[12] A. EDELMAN, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 543–560.
[13] H. EKBLOM, *A new algorithm for the Huber estimator in linear models*, BIT, 28 (1988), pp. 123–132.
[14] R. C. FAIR, *On the robust estimation of econometric models* (ref. by [17]), Ann. Econ. Social Measurement, 3 (1974), pp. 667–678.
[15] G. GOLUB, V. KLEMA, AND S. C. PETERS, *Rules and software for detecting rank degeneracy*, J. Econometrics, 12 (1980), pp. 41–48.
[16] M. J. HINICH AND P. P. TALWAR, *A simple method for robust regression*, J. Amer. Statist. Assoc., 70 (1975), pp. 113–119.
[17] P. W. HOLLAND AND R. E. WELSCH, *Robust regression using iteratively reweighted least-squares*, Commun. Statist. - Theor. Meth., A6 (1977), pp. 813–827.
[18] P. J. HUBER, *Robust estimation of a location parameter*, Annals Math. Statist., 35 (1964), pp. 73–101.
[19] ———, *Robust Statistics*, John Wiley, New York, 1981.
[20] N. KARMARKAR, *A new polynomial algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[21] D. P. O'LEARY, *On bounds for scaled projections and pseudo-inverses*, Linear Algebra Appl., 1990, to appear.
[22] ———, *A discrete Newton algorithm for minimizing a function of many variables*, Math. Programming, 23 (1982), pp. 20–33.
[23] ———, *Parallel implementation of the block conjugate gradient algorithm*, Parallel Comput., 5 (1987), pp. 127–139.

[24] J. A. SCALES, A. GERSZTENKORN, AND S. TREITEL, *Fast l_p solution of large, sparse, linear systems: application to seismic travel time tomography*, J. Comput. Phys., 75 (1988), pp. 314–333.

[25] D. F. SHANNO AND D. M. ROCKE, *Numerical methods for robust regression: Linear models*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 86–97.

[26] G. W. STEWART, *On scaled projections and pseudo-inverses*, Linear Algebra Appl., 112 (1989), pp. 189–194.

[27] R. P. TEWARSON, *On the orthonormalization of sparse vectors*, Computing, 3 (1968), pp. 268–279.

DEDICATION TO VERA N. KUBLANOVSKAYA ON HER 70TH BIRTHDAY

Vera Nikolajevna Kublanovskaya was born on September 21, 1920. She is a distinguished representative of the world-famous Soviet research tradition in linear algebra stemming from Gantmacher and Kantorovich, who were her teachers. Her Ph.D. thesis (Kandidat in Russian) appeared in 1955 on the topic of the application of analytic continuation to numerical methods. In 1972 she submitted her thesis for the senior Russian Doctorate and now the theme was the use of orthogonal transformations to solve algebraic problems. During most of her professional career she has worked at the Steklov Institute of Mathematics in Leningrad (a branch of the USSR Academy of Sciences), where she collaborated with Faddeev and Faddeeva, among others. In October 1985 Kublanovskaya was awarded an honorary doctorate at the University of Umeå, Sweden.

Vera Kublanovskaya belongs to the generation who laid the foundations of modern computational techniques in matrix analysis and its applications, such as Householder, Forsythe, and Wilkinson. Although she had been invited to several Gatlinburg symposia (nowadays called the Householder symposia), the Oxford meeting in 1981, organized by Fox and Wilkinson, was the first of these symposia that she was able to attend. Since then she has been invited to several more international conferences, of which she could unfortunately attend only a few. Those of us who had the privilege of hearing her speak in June of this year at the Householder Symposium XI in Tylösand, Sweden, are certainly impressed by the scientific activity she displays at her age. She is a source of great inspiration to the international scientific community.

Kublanovskaya's 1961 paper "On some algorithms for the solution of the complete eigenvalue problem," together with Francis's paper published in the same year, forms the basis of the *QR* algorithm for computing the eigenvalues of an unsymmetric matrix. In this paper Kublanovskaya also presents a convergence proof of the *QR* algorithm based on sophisticated determinantal theory. Her 1966 paper "On a method for solving the complete eigenvalue problem for a degenerate matrix" (translated from Russian in 1968) was another milestone in this area. There she presents a method for computing the Jordan structure of a multiple eigenvalue by unitary similarity transformations. This paper stimulated several subsequent papers on the numerical computation of the Jordan and Kronecker canonical forms.

In the recent series of papers "Spectral problems for matrix pencils: Methods and algorithms. I, II and III," Kublanovskaya illustrates well her impact on eigenvalue problems and their generalizations to matrix pencils and polynomial matrices. These papers give a fine survey of computational methods for these generalized eigenvalue problems and their applications in systems theory, and nicely demonstrate her originality in developing new algorithms. They almost add up to a book on generalized eigenvalue problems, and they give a thorough description of algorithms for computing the complete eigenstructure of matrix pencils and polynomial matrices in their most general form. Several of these algorithms are due to her and her collaborators.

Those interested in reading some of Kublanovskaya's important contributions in detail will find a selected list of her English and Russian publications below. It is our pleasure to celebrate Vera Kublanovskaya's 70th birthday by dedicating this issue of the *SIAM Journal on Matrix Analysis and Applications* to her. Her papers have been an inspiration to us, and we look forward to seeing a continuation of her excellent contri butions. We wish her good health and many more creative years.

Gene Golub
Bo Kågström
Axel Ruhe
Paul Van Dooren

SELECTED PUBLICATIONS OF VERA KUBLANOVSKAYA

IN ENGLISH

On some algorithms for the solution of the complete eigenvalue problem, U.S.S.R. Comput. Math. and Math. Physics, 3 (1961), pp. 637–657.

with V. N. Faddeeva, *Computational methods for the solution of a generalized eigenvalue problem*, Amer. Math. Soc. Transl., 2 (1964), pp. 271–290.

An approach to construct the canonical basis of a matrix, in Proceedings of the International Congress of Mathematicians, 1966.

On a method for solving the complete eigenvalue problem for a degenerate matrix, U.S.S.R. Comput. Math. and Math. Physics, 6 (1968), pp. 1–14.

On an approach to the solution of the generalized latent value problem for λ-matrices, SIAM J. Numer. Anal., 7 (1970), pp. 532–537.

Construction of a canonical basis for matrices and pencils of matrices, Soviet Mathematics, 20 (1982), pp. 1929–1942.

Eigenvalue problem for a regular linear pencil of matrices close to singular ones, J. Soviet Mathematics, 20 (1982), pp. 1943–1958.

An approach to solve the spectral problem for A-λB, in Matrix Pencils, B. Kågström and A. Ruhe, eds., Lecture Notes in Math., 973 (1983), pp. 17–29.

The AB*-algorithm and its modifications for the spectral problem of linear pencils*, Numer. Math., 43 (1984), pp. 319–342.

Solution of spectral problems for matrix pencils, in Numerical Methods, Colloquie Societas János Bolyai, North-Holland, 1987, pp. 65–107.

with V. B. Khazanov, *Deflation in spectral problems for matrix pencils*, Soviet J. Numer. Anal. Math. Modelling, 2 (1987), pp. 15–36.

with V. B. Khazanov, Spectral problems for matrix pencils. Methods and algorithms. I, Soviet J. Numer. Anal. Math. Modelling, 3 (1988), pp. 337–371.

with V. B. Khazanov, Spectral problems for matrix pencils. Methods and algorithms. II, Soviet J. Numer. Anal. Math. Modelling, 3 (1988), pp. 467–484.

with V. A. Belyi and V. B. Khazanov, Spectral problems for matrix pencils. Methods and algorithms. III, Soviet J. Numer. Anal. Math. Modelling, 4 (1989), pp. 19–51.

IN RUSSIAN

Application of analytic continuation to numerical analysis, Kandidat, University of Leningrad, Leningrad, USSR, 1955.

An approach to solve the eigenvalue problem for a singular matrix, Zh. Vychisl. Mat. i Mat. Fiz., 6 (1966), pp. 611–620.

with D. K. Faddeev and V. N. Faddeeva, *Linear algebraic systems with rectangular matrices*, in Modern Numerical Methods, Vol. 1, Moscow, 1969, pp. 16–75. (Also in French in Colloq. Internat. de CNRS No. 165 Besancon 1966, Paris, 1968, pp. 161–170.)

On the application of the Newton method to determine the eigenvalues of λ-matrices, Dokl. Akad. Nauk SSSR, 7 (1969), pp. 1004–1005.

Newton's method to compute matrix eigenvalues and eigenvectors, Zh. Vychisl. Mat. i Mat. Fiz., 12 (1972), pp. 1371–1380.

with T. N. Smirnova and V. B. Khazanov, *On the solùtion of the eigenvalue problem for a polynomial matrix*, Proc. Leningrad Shipbuilding Inst., 97 (1975), pp. 94–111.

with T. Y. Konkova and L. T. Savinova, *On the solution of a nonlinear matrix spectral problem*, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 58 (1976), pp. 54–66.

with V. B. Mikhailov and V. B. Khazanov, *On the solution of the eigenvalue problem of a nonregular λ-matrix*, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 58 (1976), pp. 80–92.

On the analysis of singular matrix pencils, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 70 (1977), pp. 89–102.

On the spectral problem for polynomial matrices, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 80 (1978), pp 83–97.

On the spectral problem for polynomial matrices. 2, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 80 (1978), pp. 97–116.

On the solution of spectral problems for singular matrix pencils, Zh. Vychisl. Mat. i Mat. Fiz., 18 (1978), pp. 1056–1060.

The AB-algorithm and its properties, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 102 (1980), pp. 42–60.

On the spectral problem for polynomial matrices, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 111 (1981), pp. 109–116.

with V. N. Simonova, *On some modifications to the AB-algorithm*, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 111 (1981), pp. 117–136.

On an algorithm for solving spectral problems of linear matrix pencils, LOMI, Preprint E-1-82, Leningrad, 1982.

An approach to construct the fundamental polynomial solution row and Jordan chains for singular linear matrix pencils, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 124 (1983), pp. 101–113.

The construction of the fundamental solution row for matrix pencils, Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (*LOMI*), 139 (1984), pp. 74–93.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 481–498, October 1990

AN ANALOG OF THE CAUCHY–SCHWARZ INEQUALITY FOR HADAMARD PRODUCTS AND UNITARILY INVARIANT NORMS*

ROGER A. HORN† AND ROY MATHIAS†‡

Abstract. The authors show that for any unitarily invariant norm $\|\cdot\|$ on M_n (the space of n-by-n complex matrices)

$$\|A^*B\|^2 \leq \|A^*A\|\,\|B^*B\| \quad \text{for all } A, B \in M_{m,n} \tag{1}$$

and

$$\|A \circ B\|^2 \leq \|A^*A\|\,\|B^*B\| \quad \text{for all } A, B \in M_n,$$

where $\circ$ denotes the Hadamard (entrywise) product. These results are a consequence of an inequality for absolute norms on C^n

$$\|x \circ y\|^2 \leq \|x \circ \bar{x}\|\,\|y \circ \bar{y}\| \quad \text{for all } x, y \in C^n. \tag{2}$$

The authors also characterize the norms on C^n that satisfy (2), characterize the unitary similarity invariant norms on M_n that satisfy (1), and obtain related results on norms on C^n and unitary similarity invariant norms on M_n that are of independent interest.

Key words. Cauchy–Schwarz inequality, unitarily invariant norms, absolute norms, Hadamard products, unitary similarity invariant norms

AMS(MOS) subject classifications. 15A60, 15A18, 15A45

1. Introduction and notation.

Let $M_{m,n}$ denote the space of m-by-n complex matrices and write $M_n \equiv M_{n,n}$; let $A^* \equiv \bar{A}^*$ denote the conjugate transpose of a matrix in $M_{m,n}$. Recently, Wimmer [20, p. 315] conjectured that an analog of the Cauchy–Schwarz inequality holds for any unitarily invariant norm $\|\cdot\|$ on $M_{m,n}$:

$$\|A^*B\|^2 \leq \|A^*A\|\,\|B^*B\| \quad \text{for all } A, B \in M_{m,n}. \tag{1.1}$$

For three special choices of norm $\|\cdot\|$ (the trace norm, the Frobenius norm, and the spectral norm), Wimmer proved (1.1) and identified the cases of equality.

In §3 we give a proof of (1.1) and a similar inequality for Hadamard products; both results follow from a simple norm inequality (Theorem 2.3) for the Hadamard product of vectors. We identify the cases of equality for the latter inequality as well as for (1.1). In §§3 and 4 we prove some results of independent interest on unitary similarity invariant norms. In §4 we provide a variety of examples and show that the set of norms satisfying (1.1) is a convex set that strictly contains the unitarily invariant norms.

We use $A \succeq 0$ to mean that A is positive semidefinite. If $A \succeq 0$ then $A^{1/2}$ denotes the unique positive semidefinite square root of A. Given $A \in M_{m,n}$ we define $|A| \equiv (A^*A)^{1/2}$. The real vector space of n-by-n Hermitian matrices is denoted by H_n. If $A, B \in H_n$, we write $A \succeq B$ if $A - B \succeq 0$. Recall that the *Hadamard product* of two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ of the same size is $A \circ B \equiv [a_{ij}b_{ij}]$. We denote the ordered singular values of any $A \in M_{m,n}$ by $\sigma_1(A) \geq \cdots \geq \sigma_q(A) \geq$

* Received by the editors July 5, 1989; accepted for publication (in revised form) November 11, 1989.

† Mathematical Sciences Department, The Johns Hopkins University, Baltimore, Maryland 21218.
‡ Present address, Mathematics Department, The College of William & Mary, Williamsburg, Virginia 23185.

0 (where $q = \min\{m,n\}$) and define $\sigma(A) \equiv [\sigma_1(A),\cdots,\sigma_q(A)]^T \in R_+^q$; for $A \in H_n$ we denote the ordered eigenvalues of A by $\lambda_1(A) \geq \cdots \geq \lambda_n(A)$ and define $\lambda(A) \equiv [\lambda_1(A),\cdots,\lambda_n(A)]^T \in R^n$. The eigenvalues and singular values of a positive semidefinite matrix are identical. A complex matrix is a *partial isometry* if each of its singular values is 0 or 1. The *trace* of a square matrix A (the sum of its main diagonal entries, or, equivalently, the sum of its eigenvalues) is denoted by tr A.

Given $x \in C^n$ and an index set $\mathcal{I} \subset \{1,\cdots,n\}$ we define $x(\mathcal{I}) \in C^n$ by

$$x(\mathcal{I})_i = \begin{cases} x_i & \text{if } i \in \mathcal{I}, \\ 0 & \text{if } i \notin \mathcal{I}, \end{cases}$$

and we define $|x| \equiv [|x_i|]_{i=1}^n$. Given a vector $x \in C^n$ we define $\operatorname{diag}(x) \in M_n$ to be the diagonal matrix with i,i entry x_i. Given vectors $x,y \in R^n$ we use $x \leq y$ to mean that $x_i \leq y_i$ for $i = 1,\cdots,n$. A norm $\|\cdot\|$ on C^n is *absolute* if $\|x\| = \| \, |x| \, \|$ for all $x \in C^n$, and is *monotone* if $|x| \geq |y|$ implies $\|x\| \geq \|y\|$. These two notions were introduced by Bauer, Stoer, and Witzgall in [2], where they arose naturally in the study of induced norms on M_n. It is a fact that a norm is absolute if and only if it is monotone [2, Thm. 2] or [8, Thm. 5.5.10].

If vectors $x,y \in R^n$ are given, and if τ and π are permutations of $\{1,\ 2,\ \cdots,\ n\}$ such that $x_{\tau(1)} \geq x_{\tau(2)} \geq \cdots \geq x_{\tau(n)}$ and $y_{\pi(1)} \geq y_{\pi(2)} \geq \cdots \geq y_{\pi(n)}$, we say that x is *weakly majorized* by y if

$$\sum_{i=1}^k x_{\tau(i)} \leq \sum_{i=1}^k y_{\pi(i)} \quad \text{for all } k = 1,\cdots,n.$$

If, in addition, equality holds when $k = n$, then we say that x is *majorized* by y. A function $g(\cdot) : C^n \to R_+$ is called a *symmetric gauge function* if it is a permutation invariant absolute norm on C^n. We will make frequent use of the fact that for $x,y \in C^n$ and any symmetric gauge function $g(\cdot)$

$$|x| \text{ is weakly majorized by } |y| \quad \text{implies} \quad g(x) \leq g(y). \tag{1.2}$$

Given a norm $\|\cdot\|$ on C^m we define its *dual* (with respect to the Euclidean inner product) by

$$\|x\|^D \equiv \max\{|y^*x| : y \in C^m, \|y\| \leq 1\}. \tag{1.3}$$

Given a norm $\|\cdot\|$ on $M_{m,n}$ we define its *dual* (with respect to the Frobenius inner product $< A,B > \equiv \operatorname{tr} B^*A$) by

$$\|A\|^D \equiv \max\{|\operatorname{tr}(B^*A)| : B \in M_{m,n}, \|B\| \leq 1\}.$$

If we take $n = 1$, then this definition specializes to (1.3). The duality theorem for norms [8, Thm. 5.5.14] states that $\|\cdot\| = (\|\cdot\|^D)^D$ for any norm $\|\cdot\|$. A norm $\|\cdot\|$ on $M_{m,n}$ is *unitarily invariant* if $\|A\| = \|UAV\|$ for all $A \in M_{m,n}$ and all unitary $U \in M_m$ and $V \in M_n$. A theorem of von Neumann [19] (or [8, Thm. 7.4.24], or [16, Thm. V.5]) states that a norm $\|\cdot\|$ on $M_{m,n}$ is unitarily invariant if and only if there is a symmetric gauge function g such that $\|X\| = g(\sigma(X))$ for all $X \in M_{m,n}$. A norm $\|\cdot\|$ on M_n is *unitary similarity invariant* if $\|A\| = \|UAU^*\|$ for all $A,U \in M_n$ with U unitary.

See [8] for further information on Hadamard products, norms, dual norms, unitarily invariant norms, symmetric gauge functions, singular values, and other concepts discussed in this paper. See [13, p. 263] for a general discussion of the connection between majorization and unitarily invariant norms.

2. An inequality for absolute norms. In this section we are interested in an inequality for Hadamard products of vectors that leads directly to a proof of the matrix inequality (1.1). To obtain Theorem 2.3, the main result in this section, it is helpful to know two lemmata, whose proofs we omit. The first result is Theorem 1 in [2]; the second can be proved by an argument very similar to the proof of Lemma 3.7.

LEMMA 2.1. *A norm on C^n is absolute if and only if its dual norm is absolute.*

LEMMA 2.2. *Let $\|\cdot\|$ be an absolute norm on C^n and let $x \in C^n$ be given. Then*

$$\begin{aligned} \|x\| &= \max\{|y^*x| : y \in C^n \text{ and } \|y\|^D \le 1\} \\ &= \max\{z^T|x| : z \in R^n_+ \text{ and } \|z\|^D \le 1\}. \end{aligned}$$

We use the following notation in Theorem 2.3. Given $x \in C^n$ and an index set $\mathcal{I} \subset \{1, \cdots, n\}$, define $x(\mathcal{I}) \in C^n$ by

$$x(\mathcal{I})_i = \begin{cases} x_i & \text{if } i \in \mathcal{I}, \\ 0 & \text{if } i \notin \mathcal{I}. \end{cases}$$

THEOREM 2.3. *Let $\|\cdot\|$ be an absolute norm on C^n. Then*

$$\|x \circ y\|^2 \le \|x \circ \bar{x}\| \, \|y \circ \bar{y}\| \quad \text{for all } x, y \in C^n. \tag{2.1}$$

If $x, y \neq 0$ then equality holds in (2.1) if and only if there is a positive constant c and an index set $\mathcal{I} \subset \{1, \cdots, n\}$ such that

$$|x(\mathcal{I})| = c|y(\mathcal{I})|, \ \|x \circ \bar{x}(\mathcal{I})\| = \|x\|, \ \text{and} \ \|y \circ \bar{y}(\mathcal{I})\| = \|y\|.$$

Proof. Use Lemma 2.2 to compute

$$\begin{aligned} \|x \circ y\|^2 &= [\max\{z^T(|x| \circ |y|) : z \in R^n_+ \text{ and } \|z\|^D \le 1\}]^2 \\ &= \max\{[(z^{1/2} \circ |x|)^T (z^{1/2} \circ |y|)]^2 : z \in R^n_+ \text{ and } \|z\|^D \le 1\} \\ &\le \max\{[(z^{1/2} \circ |x|)^T (z^{1/2} \circ |x|)][(z^{1/2} \circ |y|)^T (z^{1/2} \circ |y|)] : \\ &\qquad\qquad z \in R^n_+ \text{ and } \|z\|^D \le 1\} \qquad (2.3) \\ &\le \max\{(z^T|x \circ x|) : z \in R^n_+ \text{and } \|z\|^D \le 1\} \\ &\qquad \cdot \max\{(z^T|y \circ y|) : z \in R^n_+ \text{and } \|z\|^D \le 1\} \qquad (2.4) \\ &= \| \, |x \circ x| \, \| \, \| \, |y \circ y| \, \| \\ &= \|x \circ \bar{x}\| \, \|y \circ \bar{y}\|. \end{aligned}$$

For $z = [z_i] \in R^n_+$, we have written $z^{1/2} \equiv [z_i^{1/2}] \in R^n_+$ for the Hadamard (entrywise) nonnegative square root of z.

That the stated conditions are sufficient for equality is clear from the monotonicity of $\|\cdot\|$:

$$\begin{aligned} \|x \circ y\|^2 &\ge \|x \circ y(\mathcal{I})\|^2 \\ &= \|x \circ (1/c)x(\mathcal{I})\| \, \|cy \circ y(\mathcal{I})\| \\ &= \|x \circ \bar{x}(\mathcal{I})\| \, \|y \circ \bar{y}(\mathcal{I})\| \\ &= \|x \circ \bar{x}\| \, \|y \circ \bar{y}\| \end{aligned}$$

and hence equality holds in (2.1).

Conversely, suppose that equality holds in (2.1) for given nonzero vectors x, y. Then any $z \in R_+^n$ that attains the maximum in (2.2) must also attain the maximum in (2.3) and both maxima in (2.4). Let $\tilde{z}$ be such a vector and define the index set $\mathcal{I} \equiv \{i : \tilde{z}_i > 0\}$. Equality in (2.3) implies that there is a positive scalar c such that

$$\tilde{z}_i^{1/2}|x|_i = c\tilde{z}_i^{1/2}|y|_i \quad \text{for } i = 1, \cdots, n$$

and hence, by the definition of $\mathcal{I}$, it follows that $|x(\mathcal{I})| = c|y(\mathcal{I})|$. Because $\tilde{z}$ attains the first maximum in (2.4), we have $\|x \circ \bar{x}\| = \tilde{z}^T|x \circ \bar{x}|$, and we can use Lemma 2.2 again to compute

$$\begin{aligned} \|x \circ \bar{x}(\mathcal{I})\| &= \max\{z^T|x \circ \bar{x}(\mathcal{I})| : z \in R_+^n \text{ and } \|z\|^D \leq 1\} \\ &\geq \tilde{z}^T(x \circ \bar{x}) \\ &= \|x \circ \bar{x}\|. \end{aligned}$$

But $\|x \circ \bar{x}\| \geq \|x \circ \bar{x}(\mathcal{I})\|$ by the monotonicity of $\|\cdot\|$, so $\|x \circ \bar{x}\| = \|x \circ \bar{x}(\mathcal{I})\|$. The same argument shows that $\|y \circ \bar{y}\| = \|y \circ \bar{y}(\mathcal{I})\|$. □

Note that the inequality (2.1) with the l_1 norm is the heart of the classical Cauchy–Schwarz inequality:

$$\left|\sum_{i=1}^n x_i y_i\right|^2 \leq \left\{\sum_{i=1}^n |x_i y_i|\right\}^2 = \|x \circ y\|_1^2 \leq \|x \circ \bar{x}\|_1 \, \|y \circ \bar{y}\|_1 = \sum_{i=1}^n |x_i|^2 \sum_{i=1}^n |y_i|^2.$$

It is of interest to characterize the norms on C^n that satisfy the conclusion of Theorem 2.3. We discuss the converse of the following preliminary lemma in Theorem 4.8.

LEMMA 2.4. *Let $\|\cdot\|$ be a norm on C^n such that $\|x\| \leq \| |x| \|$ for all $x = [x_i] \in C^n$, where $|x| \equiv [|x_i|]$. Then the function $\nu(x) \equiv \| |x| \|$ is a norm on C^n.*

Proof. Since the function $\nu(x) \equiv \| |x| \|$ is positive definite and homogeneous on C^n, we need only show that it obeys the triangle inequality. We claim that it suffices to prove that

$$\|u\| \leq \|v\| \text{ whenever } u, v \in R_+^n \text{ and } u \leq v. \tag{2.5}$$

Since $|x + y| \leq |x| + |y|$ for all $x, y \in C^n$, (2.5) and the triangle inequality for $\|\cdot\|$ give the desired result:

$$\nu(x + y) = \| |x + y| \| \leq \| |x| + |y| \| \leq \| |x| \| + \| |y| \| = \nu(x) + \nu(y).$$

To prove (2.5), let $u, v \in R_+^n$ be given with $u \leq v$. If $u = v$, there is nothing to prove, so assume that $u \neq v$. Some corresponding entries of u and v may be equal but at least one entry of u must be strictly less than the corresponding entry of v. We shall construct a vector $w \in R_+^n$ such that $u \leq w \leq v$, $\|w\| \leq \|v\|$, and w has one more entry than v that is equal to the corresponding entry of u. A finite induction then leads to the conclusion that $\|u\| \leq \|v\|$. Define $k = \min\{i : u_i < v_i, i = 1, \cdots, n\}$ and define $v', v'' \in R^n$ by

$$v_i' = \begin{cases} v_i & \text{for } i \neq k, \\ -v_i & \text{for } i = k, \end{cases} \qquad v_i'' = \begin{cases} v_i & \text{for } i \neq k, \\ 0 & \text{for } i = k. \end{cases}$$

Note that $v'' = \frac{1}{2}(v' + v)$ and $|v'| = v$. Using the hypothesis on $\|\cdot\|$, we have

$$\|v'\| \leq \| |v'| \| \leq \|v\|,$$

and hence

$$\|v''\| = \tfrac{1}{2}\|v + v'\| \leq \tfrac{1}{2}(\|v\| + \|v'\|) \leq \tfrac{1}{2}(\|v\| + \|v\|) = \|v\|. \tag{2.6}$$

Now define $\alpha \equiv u_k/v_k$, so $0 \leq \alpha < 1$. Define $w \equiv \alpha v + (1-\alpha)v''$ and note that $w_i = v_i$ if $i \neq k$ and that $w_k = u_k$. Thus, w has one more entry than v that is equal to the corresponding entry of u. Using (2.6) we obtain

$$\|w\| = \|\alpha v + (1-\alpha)v''\| \leq \alpha\|v\| + (1-\alpha)\|v''\| \leq \alpha\|v\| + (1-\alpha)\|v\| = \|v\|,$$

as desired. □

We can now characterize the norms that satisfy the inequality (2.1).

THEOREM 2.5. *Let $\|\cdot\|$ be a norm on C^n. Then*

$$\|x \circ y\|^2 \leq \|x \circ \bar{x}\|\,\|y \circ \bar{y}\| \quad \textit{for all} \quad x, y \in C^n \tag{2.7}$$

if and only if

$$\|z\| \leq \|\,|z|\,\| \quad \textit{for all} \quad z = [z_i] \in C^n, \tag{2.8}$$

where $|z| \equiv [|z_i|]$.

Proof. Suppose $\|z\| \leq \|\,|z|\,\|$ for all $z \in C^n$. Lemma 2.4 guarantees that $\nu(x) \equiv \|\,|x|\,\|$ is an absolute norm on C^n, so we may apply Theorem 2.3 to $\nu(\cdot)$ and obtain

$$\|x \circ y\|^2 \leq \|\,|x \circ y|\,\|^2 = \nu^2(x \circ y) \leq \nu(x \circ \bar{x})\nu(y \circ \bar{y}) = \|x \circ \bar{x}\|\,\|y \circ \bar{y}\|.$$

Conversely, suppose (2.7) holds and let $z \in C^n$ be given. Define $x, y \in C^n$ by

$$x_i \equiv \begin{cases} z_i/|z_i|^{1/2} & \text{if } z_i \neq 0, \\ 0 & \text{if } z_i = 0, \end{cases} \qquad y_i \equiv |z_i|^{1/2}, \quad i = 1, \cdots, n.$$

Then

$$\|z\|^2 = \|x \circ y\|^2 \leq \|x \circ \bar{x}\|\,\|y \circ \bar{y}\| = \|\,|z|\,\|\,\|\,|z|\,\| = \|\,|z|\,\|^2,$$

which is (2.7). □

An example of a norm on C^2 that is not absolute but nevertheless satisfies the condition (2.8), and hence (2.7) as well, is $\|x\| \equiv \max\{|x_1 + x_2|, |x_1|, |x_2|\}$.

Although we have characterized the norms for which the inequality (2.7) holds in terms of the natural condition (2.8), it is not always easy to determine whether a particular norm has this property. For example, it is not known which unitarily invariant norms on $M_{m,n}$ satisfy (2.8).

3. Inequalities for matrices. We are now ready to prove (1.1), as well as an analogous inequality for the Hadamard product of matrices, and to discuss the cases of equality.

THEOREM 3.1. *Let $\|\cdot\|$ be a unitarily invariant norm on M_n. Then*

$$\|A^*B\|^2 \leq \|A^*A\|\,\|B^*B\| \quad \textit{for all } A, B \in M_{m,n} \tag{3.1}$$

and

$$\|A \circ B\|^2 \leq \|A^*A\|\,\|B^*B\| \quad \textit{for all } A, B \in M_{m,n}. \tag{3.2}$$

Inequality (3.2) has also been obtained by Okubo [14, Thm. 4.3], while (3.1) can be derived by using an argument similar to that used by Bhatia to prove Proposition 5 (another Cauchy–Schwarz type inequality) in [4]. Both of these inequalities can also be derived as corollaries of Theorem 2.3 in [10].

Proof. Let g be the symmetric gauge function associated with the unitarily invariant norm $\|\cdot\|$. Theorem 3 of A. Horn [6] gives the weak majorization relation

$$\sum_{i=1}^{k} \sigma_i(A^*B) \leq \sum_{i=1}^{k} \sigma_i(A)\sigma_i(B) \qquad k = 1, \cdots, n \tag{3.3}$$

between the singular values of the product A^*B and those of A and B. Compute

$$\begin{aligned} \|A^*B\|^2 &= g^2(\sigma_1(A^*B), \cdots, \sigma_n(A^*B)) \\ &\leq g^2(\sigma_1(A)\sigma_1(B), \cdots, \sigma_n(A)\sigma_n(B)) \\ &\leq g(\sigma_1^2(A), \cdots, \sigma_n^2(A))\, g(\sigma_1^2(B), \cdots, \sigma_n^2(B)) \\ &= g(\sigma_1(A^*A), \cdots, \sigma_n(A^*A))\, g(\sigma_1(B^*B), \cdots, \sigma_n(B^*B)) \\ &= \|A^*A\|\, \|B^*B\|. \end{aligned}$$

The first inequality comes from combining (3.3) and (1.2), the second is an application of Theorem 2.3 to the monotone norm $g(\cdot)$, and the penultimate equality is because $\sigma_i^2(A) = \sigma_i(A^*A)$.

To prove (3.2) we use the weak majorization relation [7, Lem. 1]

$$\sum_{i=1}^{k} \sigma_i(A \circ B) \leq \sum_{i=1}^{k} \sigma_i(A)\sigma_i(B), \qquad k = 1, \cdots, n \tag{3.4}$$

for the Hadamard product and apply exactly the same argument. □

Inequality (3.2) is a generalization to all unitarily invariant norms of a classical inequality of Schur for the spectral norm [17, Satz III, p. 8]: If $\|\cdot\|$ is chosen to be the spectral norm $\|X\|_2 \equiv \sigma_1(X)$, then (3.2) is Schur's inequality $\sigma_1(A \circ B) \leq \sigma_1(A)\sigma_1(B)$.

Theorem 3.1 allows us to make the following generalization of Theorem 2.3 in [20].

COROLLARY 3.2. *Let $\|\cdot\|$ be a given unitarily invariant norm on M_n. Then for all $A \in M_n$,*

$$\|A\| = \min\{\|B^*B\|^{1/2}\|C^*C\|^{1/2} : B, C \in M_n \text{ and } B^*C = A\}. \tag{3.5}$$

Proof. For any $A \in M_n$, Theorem 3.1 gives

$$\|A\| \leq \inf\{\|B^*B\|^{1/2}\|C^*C\|^{1/2} : B, C \in M_n \text{ and } B^*C = A\}. \tag{3.6}$$

That the infimum is attained and is equal to $\|A\|$ follows by setting $B = P^{1/2}$ and $C = P^{1/2}U$, where $A = PU$ is a polar decomposition of A, i.e., $P, U \in M_n$, $P \succeq 0$, and U is unitary. □

In Example 4.4 we give a non-unitarily invariant norm that satisfies (3.5). It is possible to characterize the unitary similarity invariant norms that satisfy (1.1); to do so, we require the following analog of Lemma 2.4.

LEMMA 3.3. *Let $\|\cdot\|$ be a unitary similarity invariant norm on M_n such that*

$$\|A\| \leq \| \, |A| \, \| \quad \text{for all } A \in M_n, \tag{3.7}$$

where $|A| \equiv (A^*A)^{1/2}$. *Then* $N(A) \equiv \| \, |A| \, \|$ *is a unitarily invariant norm on* M_n.

Proof. The unitary invariance, positivity, and homogeneity of the function $N(\cdot)$ are clear, so it suffices to prove that $N(\cdot)$ obeys the triangle inequality. We have the weak majorization (see [5], or [13, p. 243], or [9, Cor. 3.4.3]):

$$\sum_{i=1}^{k} \sigma_i(A+B) \leq \sum_{i=1}^{k} \sigma_i(A) + \sum_{i=1}^{k} \sigma_i(B), \quad k = 1, \cdots, n,$$

which expresses the subadditivity of the Ky Fan k-norms, i.e., the vector $\sigma(A+B)$ is weakly majorized by the vector $\sigma(A) + \sigma(B)$. Define a norm $\| \cdot \|'$ on C^n by $\|x\|' \equiv \|\mathrm{diag}(x)\|$. Condition (3.7) guarantees that $\|x\|' \leq \| \, |x| \, \|'$ for all $x \in C^n$, so Lemma 2.4 ensures that the function $\|x\|'' \equiv \| \, |x| \, \|'$ is a monotone norm on C^n. Since the given norm $\| \cdot \|$ is unitary similarity invariant, the norm $\| \cdot \|''$ is permutation invariant. Thus, the norm $\| \cdot \|''$ is actually a symmetric gauge function.

If $C \in M_n$ is a given positive semidefinite matrix, then there is a unitary $U \in M_n$ such that $C = U \Lambda U^*$, where $\Lambda = \mathrm{diag}(\lambda(C))$. Because the norm $\| \cdot \|$ is unitary similarity invariant,

$$\|C\| = \|U \Lambda U^*\| = \|\mathrm{diag}(\lambda(C))\| = \|\lambda(C)\|' = \|\sigma(C)\|''.$$

Now use this identity with $C \equiv |A+B|$, noting that the eigenvalues and singular values of a positive semidefinite matrix are identical, to write

$$\begin{aligned} N(A+B) &= \| \, |A+B| \, \| \\ &= \|\sigma(A+B)\|'' \\ &\leq \|\sigma(A) + \sigma(B)\|'' \\ &\leq \|\sigma(A)\|'' + \|\sigma(B)\|'' \\ &= \| \, |A| \, \| + \| \, |B| \, \| \\ &= N(A) + N(B). \end{aligned}$$

The first inequality uses weak majorization and the fact that $\| \cdot \|''$ is a symmetric gauge function, while the second is just the triangle inequality for $\| \cdot \|''$. □

The condition (3.7) is sufficient for a unitary similarity invariant norm on M_n to satisfy the conclusion of Lemma 3.3, but it is not necessary. See Theorem 4.9 for a stronger version of Lemma 3.3, which provides a necessary and sufficient condition.

Another way to prove the triangle inequality for $N(A) \equiv \| \, |A| \, \|''$ is to use the matrix-valued triangle inequality (see [18, Thm. 2] or [9, 3.1.15]). We demonstrate this technique in the proof of Theorem 4.9.

The following characterization is a matrix analog of Theorem 2.5 for the usual matrix product.

THEOREM 3.4. *Let* $\| \cdot \|$ *be a unitary similarity invariant norm on* M_n. *Then*

$$\|A^*B\|^2 \leq \|A^*A\| \, \|B^*B\| \quad \textit{for all } A, B \in M_n \tag{3.8}$$

if and only if

$$\|A\| \leq \| \, |A| \, \| \quad \textit{for all } A \in M_n, \tag{3.9}$$

where $|A| \equiv (A^*A)^{1/2}$. *If either* (3.8) *or* (3.9) *holds, then*

$$\|A^*B\|^2 \leq \| \, |A^*B| \, \|^2 \leq \|A^*A\| \, \|B^*B\| \quad \textit{for all } A, B \in M_n. \tag{3.10}$$

Proof. Let $\|\cdot\|$ be a given unitary similarity invariant norm on M_n. If condition (3.9) holds, then the function $\|\,|\cdot|\,\|$ is a unitarily invariant norm on M_n by Lemma 3.3. It now follows from (3.1) that for any $A, B \in M_n$

$$\|A^*B\|^2 \leq \|\,|A^*B|\,\|^2 \leq \|\,|A^*A|\,\|\,\|\,|B^*B|\,\| = \|A^*A\|\,\|B^*B\|.$$

Thus (3.9) implies both (3.8) and (3.10).

Conversely, suppose that (3.8) holds and let $A \in M_n$ be given. Let $A = UP$ be a polar decomposition of A. Using the condition (3.8) and the hypothesis of unitary similarity invariance, we obtain the desired inequality:

$$\begin{aligned}\|A\|^2 &= \|(P^{1/2}U^*)^*P^{1/2}\|^2 \\ &\leq \|(P^{1/2}U^*)^*(P^{1/2}U^*)\|\,\|(P^{1/2})^*(P^{1/2})\| \\ &= \|UPU^*\|\,\|P\| \\ &= \|P\|\,\|P\| = \|\,|A|\,\|^2.\end{aligned}$$ □

The hypothesis that the norm $\|\cdot\|$ be unitary similarity invariant is essential, as we show in Example 4.2. In Example 4.13 we exhibit a unitary similarity invariant norm that does not satisfy the condition (3.9).

We now determine the case of equality in (3.1), and to do so we require two preliminary results that are analogs of Lemmata 2.1 and 2.2. There is an analogy between absolute norms on C^n and unitarily invariant norms on M_n. A unitarily invariant norm $\|\cdot\|$ on M_n is a function only of the singular values, and hence $\|A\| = \|\,|A|\,\|$ for all $A \in M_n$ because A and $|A|$ have the same singular values.

LEMMA 3.5. *A norm on $M_{m,n}$ is unitarily invariant if and only if its dual norm is unitarily invariant.*

Proof. Let $\|\cdot\|$ be a given unitarily invariant norm and let $A \in M_{m,n}, U \in M_m$, and $V \in M_n$ be given with U and V unitary. Then

$$\begin{aligned}\|UAV\|^D &= \max\{|\text{tr } B^*UAV| : B \in M_{m,n},\ \|B\| \leq 1\} \\ &= \max\{|\text{tr } (U^*BV^*)^*A| : B \in M_{m,n},\ \|B\| \leq 1\} \\ &= \max\{|\text{tr } C^*A| : C \in M_{m,n},\ \|C\| \leq 1\} \\ &= \|A\|^D,\end{aligned}$$

which shows that $\|\cdot\|^D$ is unitarily invariant. The hypothesis that $\|\cdot\|$ is unitarily invariant is used to obtain the penultimate equality in this series of identities. The converse now follows from the duality theorem for norms. □

Before proceeding, it is convenient to isolate some simple but useful facts about the Frobenius inner product on M_n.

LEMMA 3.6. *Let $A, B \in M_n$ be given.*

(a) *If A and B are positive semidefinite then* tr $AB \geq 0$.

(b) *If A and B are Hermitian then* tr AB *is real.*

(c) *Let A be positive semidefinite and let $B = H + iK$, where $H, K \in H_n$. Then* Re tr $B^*A =$ tr $HA \leq$ tr $|H|A$.

Proof. If A and B are positive semidefinite then so is $A^{1/2}BA^{1/2}$, and hence tr $AB =$ tr $A^{1/2}BA^{1/2} \geq 0$, which verifies (a). The assertion in (b) follows from applying (a) to the positive semidefinite matrices $A + \|A\|_2 I$ and $B + \|B\|_2 I$ and noting that the trace of a Hermitian matrix is real. The assertion in (c) that Re tr $B^*A =$ Re (tr $HA - i$tr KA) = tr HA follows from (b), while the inequality tr $HA \leq$ tr $|H|A$ follows from the observation that $(|H| - H)$ is positive semidefinite and therefore tr $(|H| - H)A \geq 0$. □

The following is a matrix analog of Lemma 2.2.

LEMMA 3.7. *Let $\|\cdot\|$ be a given unitarily invariant norm on M_n and let $A \in M_n$ be given. Then*

$$\begin{aligned}\|A\| &= \max\{|\mathrm{tr}\ (C^*A)| : C \in M_n \text{ and } \|C\|^D \le 1\} \\ &= \max\{\mathrm{tr}\ C|A| : C \in H_n, C \succeq 0, \text{ and } \|C\|^D \le 1\}.\end{aligned}$$

Proof. The first identity is the duality theorem for norms. To show the second, compute

$$\begin{aligned}\|A\| &= \| \, |A| \, \| \\ &= \max\{|\mathrm{tr}\ C^*|A| \, | : C \in M_n \text{ and } \|C\|^D \le 1\} \\ &= \max\{\mathrm{Re\ tr}\ C^*|A| : C \in M_n \text{ and } \|C\|^D \le 1\} \\ &= \mathrm{tr}\ C_0^*|A|\end{aligned}$$

for some $C_0 \in M_n$ with $\|C_0\|^D \le 1$. Then $\mathrm{tr}\ C_0^*|A| \le \mathrm{tr}\ |C_0^*||A|$ by Lemma 3.6(c), and hence, using Lemma 3.5 for the second inequality,

$$\begin{aligned}\|A\| &\le \mathrm{tr}\ |C_0^*||A| \\ &\le \max\{\mathrm{tr}\ C|A| : C \in H_n, C \succeq 0, \text{ and } \|C\|^D \le 1\} \\ &\le \max\{\mathrm{Re\ tr}\ C^*|A| : C \in M_n \text{ and } \|C\|^D \le 1\} \\ &= \| \, |A| \, \| = \|A\|.\end{aligned}$$

Thus, all the inequalities must be equalities and the asserted identity follows. □

We also need a well-known result expressing the monotonicity of a unitarily invariant norm with respect to multiplication by a partial isometry [3, Prop. 7.7.3]. We provide a proof that uses only the existence of the singular value decomposition.

LEMMA 3.8. *Let $\|\cdot\|$ be a unitarily invariant norm on $M_{m,n}$, and let $A \in M_{m,n}$, $P_1 \in M_m$, and $P_2 \in M_n$ be given. Then*

$$\|P_1AP_2\| \le \sigma_1(P_1)\sigma_1(P_2)\|A\|.$$

In particular, if P_1 and P_2 are partial isometries then

$$\|P_1AP_2\| \le \|A\|.$$

Proof. Let $A \in M_{m,n}$, $P_1 \in M_m$, and $P_2 \in M_n$ and let $\|\cdot\|$ be a unitarily invariant norm on $M_{m,n}$. Assume without loss of generality that $\sigma_1(P_1) = \sigma_1(P_2) = 1$. We will show that

$$\|P_1A\| \le \|A\|.$$

The inequality

$$\|AP_2\| \le \|A\|$$

can be proved by a very similar argument. Combining these two results gives the desired conclusion.

Let $P_1 = U\Sigma V$ be a singular value decomposition of P_1, i.e., U, V are unitary and $\Sigma = \mathrm{diag}(\sigma(P_1))$. Define $s \in C^m$ by

$$s_j = \sigma_j(P_1) + i\sqrt{1 - \sigma_j^2(P_1)}, \qquad j = 1, \cdots, m$$

and define $S = \operatorname{diag}(s)$. Then S is unitary, since $0 \leq \sigma_j(P_1) \leq 1$ for all $j = 1, \cdots, m$, and $\Sigma = \frac{1}{2}(S + S^*)$. Using the unitary invariance of $\|\cdot\|$, the triangle inequality, and the fact that S and S^* are unitary we compute:

$$\begin{aligned} \|P_1 A\| &= \|U\Sigma V A\| = \|\Sigma V A\| = \|\tfrac{1}{2}(S + S^*)VA\| \\ &\leq \tfrac{1}{2}\|SVA\| + \tfrac{1}{2}\|S^*VA\| \\ &= \tfrac{1}{2}\|A\| + \tfrac{1}{2}\|A\| = \|A\|. \end{aligned}$$ □

We are now ready to identify the cases of equality in (3.1), with a result that is an analog of the last part of Theorem 2.3.

THEOREM 3.9. *Let $\|\cdot\|$ be a given unitarily invariant norm on M_n, and let $A, B \in M_{m,n}$ be given nonzero matrices. Then*

$$\|A^*B\|^2 = \|A^*A\|\|B^*B\|$$

if and only if there is a positive constant c and there are partial isometries P_1 and P_2 such that

$$AP_1 = cBP_2, \quad \|P_1^*A^*AP_1\| = \|A^*A\|, \text{ and } \|P_2^*B^*BP_2\| = \|B^*B\|. \tag{3.11}$$

Proof. Let $\|\cdot\|$ be a unitarily invariant norm on M_n, and let $A, B \in M_{m,n}$. The polar decomposition [8, Thm. 7.3.2] guarantees that there is a unitary $U \in M_n$ such that A^*BU is positive semidefinite. Use Lemma 3.7 and the Cauchy–Schwarz inequality for the Frobenius inner product to compute

$$\begin{aligned} \|A^*B\|^2 &= \|A^*BU\|^2 \\ &= \max\{[\operatorname{tr} CA^*BU]^2 : C \in H_n,\ C \succeq 0, \text{ and } \|C\|^D \leq 1\} \\ &= \max\{[\operatorname{tr} C^{1/2}A^*BUC^{1/2}]^2 : C \in H_n,\ C \succeq 0,\ \|C\|^D \leq 1\} \\ &\leq \max\{(\operatorname{tr} C^{1/2}A^*AC^{1/2})(\operatorname{tr}\ C^{1/2}U^*B^*BUC^{1/2}) : \\ &\qquad\qquad C \in H_n,\ C \succeq 0,\ \|C\|^D \leq 1\} \qquad (3.12) \\ &\leq \max\{\operatorname{tr} C^{1/2}A^*AC^{1/2} : C \in H_n,\ C \succeq 0,\ \|C\|^D \leq 1\} \\ &\quad \cdot \max\{\operatorname{tr} C^{1/2}U^*B^*BUC^{1/2} : C \in H_n, C \succeq 0, \|C\|^D \leq 1\} \qquad (3.13) \\ &= \max\{\operatorname{tr} CA^*A : C \in H_n,\ C \succeq 0,\ \|C\|^D \leq 1\} \\ &\quad \cdot \max\{\operatorname{tr} CU^*B^*BU : C \in H_n,\ C \succeq 0,\ \|C\|^D \leq 1\} \\ &= \|A^*A\|\ \|U^*B^*BU\| \\ &= \|A^*A\|\ \|B^*B\|. \end{aligned}$$

Now suppose $\|A^*B\|^2 = \|A^*A\|\|B^*B\|$, so that the preceding inequalities must all be equalities. If inequality (3.12) is an equality, then there must be a positive semidefinite $\tilde{C} \in H_n$ such that $\|\tilde{C}\|^D \leq 1$ and

$$[\operatorname{tr}\ \tilde{C}^{1/2}A^*BU\tilde{C}^{1/2}]^2 = \operatorname{tr}\,(\tilde{C}^{1/2}A^*A\tilde{C}^{1/2})\ \operatorname{tr}\,(\tilde{C}^{1/2}U^*B^*BU\tilde{C}^{1/2}).$$

This last condition states that equality holds in the Cauchy–Schwarz inequality, which can occur only if $A\tilde{C}^{1/2}$ and $BU\tilde{C}^{1/2}$ are dependent, i.e., there is a nonzero scalar α such that

$$A\tilde{C}^{1/2} = \alpha BU\tilde{C}^{1/2}. \tag{3.14}$$

If inequality (3.13) is also an equality it is necessary that

$$\|A^*A\| = \text{tr } \tilde{C}^{1/2}A^*A\tilde{C}^{1/2} \tag{3.15}$$

and

$$\|B^*B\| = \text{tr } \tilde{C}^{1/2}U^*B^*BU\tilde{C}^{1/2}. \tag{3.16}$$

Let $E \in M_n$ be the Hermitian projection onto the range of $\tilde{C}^{1/2}$, so E is a partial isometry, $E = E^*$, and $E\tilde{C}^{1/2} = \tilde{C}^{1/2}E = \tilde{C}^{1/2}$. We now show that $c \equiv |\alpha|$ is the required positive constant and $P_1 \equiv E$, $P_2 \equiv (\alpha/|\alpha|)UE$ are the required partial isometries. By the definition of E and (3.14) we have

$$AE\tilde{C}^{1/2} = A\tilde{C}^{1/2} = \alpha BU\tilde{C}^{1/2} = \alpha BUE\tilde{C}^{1/2}$$

and hence $AE = \alpha BUE$, which is the same as $AP_1 = cBP_2$. Now use Lemma 3.8, Lemma 3.7, and (3.15) to compute

$$\begin{aligned}
\|A^*A\| &\geq \|P_1^*A^*AP_1\| \\
&= \|EA^*AE\| \\
&= \max\{\text{tr } C(EA^*AE) : C \in H_n, C \succeq 0, \text{ and } \|C\|^D \leq 1\} \\
&= \max\{\text{tr } C^{1/2}EA^*AEC^{1/2} : C \in H_n, C \succeq 0, \text{ and } \|C\|^D \leq 1\} \\
&\geq \text{tr } \tilde{C}^{1/2}EA^*AE\tilde{C}^{1/2} \\
&= \text{tr } \tilde{C}^{1/2}A^*A\tilde{C}^{1/2} \\
&= \|A^*A\|.
\end{aligned}$$

Thus, $\|P_1^*A^*AP_1\| = \|A^*A\|$, as asserted. The same argument shows that $\|P_2^*B^*BP_2\| = \|B^*B\|$.

Conversely, if there is a positive constant c and there are partial isometries P_1 and P_2 such that

$$AP_1 = cBP_2, \quad \|P_1^*A^*AP_1\| = \|A^*A\|, \text{ and } \|P_2^*B^*BP_2\| = \|B^*B\|,$$

then by Lemma 3.8 we have

$$\begin{aligned}
\|A^*B\|^2 &\leq \|A^*A\|\|B^*B\| \\
&= \|P_1^*A^*AP_1\|\|P_2^*B^*BP_2\| \\
&= \|P_1^*A^*cBP_2\|\|(1/c)P_1^*A^*BP_2\| \\
&= \|P_1^*A^*BP_2\|^2 \\
&\leq \|A^*B\|^2.
\end{aligned}$$

Thus, both inequalities must be equalities. □

4. Examples, counterexamples, and corollaries. Although Wimmer's conjecture (1.1) is now settled, there are several interesting points to be noted. The first is that *not every* norm on M_n satisfies (1.1) and there are norms satisfying (1.1) that are *not* unitarily invariant. Moreover, the set of norms satisfying (1.1) is a convex set.

Example 4.1. Consider the l_p norms defined on $M_{m,n}$ by

$$\begin{aligned}
\|A\|_p &\equiv \left(\sum_{i,j} |a_{ij}|^p\right)^{1/p}, \quad 1 \leq p < \infty, \\
\|A\|_\infty &\equiv \max\{|a_{ij}| : 1 \leq i \leq m, 1 \leq j \leq n\}.
\end{aligned}$$

Let $m = n = 2$ and

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

Then (1.1) does not hold for the l_p norms when $1 \leq p < 2$ since

$$\|A^*B\|_p^2 = 2^{4/p} \not\leq (2 \cdot 2^{1/p})(2^{1/p}) = \|A^*A\|_p \|B^*B\|_p.$$

Example 4.2. Let $A = [a_{ij}], B = [b_{ij}] \in M_n$. The l_∞ norm $\|\cdot\|_\infty$ on $M_{m,n}$ is not unitarily invariant, but does satisfy (1.1). Let $A = [a_1 \cdots a_n]$ and $B = [b_1 \cdots b_n]$ be partitioned according to their columns. Then

$$\|A^*B\|_\infty^2 = \max |a_i^* b_j|^2 \leq (\max a_i^* a_i)(\max b_j^* b_j) \leq \|A^*A\|_\infty \|B^*B\|_\infty.$$

Note that $\|\cdot\|_\infty$ does not satisfy condition (3.7); for example, consider

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \quad |A| = \sqrt{\frac{1}{2}} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

For this choice of A we have $\|A\|_\infty = 1 \not\leq 1/\sqrt{2} = \| \, |A| \, \|_\infty$. However this does not contradict Theorem 3.4 because $\|\cdot\|_\infty$ is not unitary similarity invariant.

Example 4.3. Let $C \in M_n$ be given. The *C-numerical radius* is defined on M_n by

$$r_C(A) \equiv \max\{|\text{tr } CU^*AU| : U \in M_n \text{ is unitary}\}.$$

If C is not a scalar matrix and tr $C \neq 0$ then it is known [12] that $r_C(\cdot)$ is a norm on M_n. The function $r_C(\cdot)$ is unitary similarity invariant but is never unitarily invariant when $n > 1$ and $C \neq 0$ because, under these conditions, we can always construct $A \in M_n$ such that $r_C(A) \neq r_C(|A|)$. The classical numerical radius,

$$r(A) \equiv \max\{|x^*Ax| : x \in C^n \text{ and } \|x\|_2 = 1\},$$

is an example of a norm of the form $r_C(\cdot)$; it corresponds to the positive semidefinite matrix $C = [1] \oplus 0_{n-1}$. Note that

$$\begin{aligned} r_C(A) &= \max\{|\text{tr } (CU^*AU)| : U \in M_n \text{ is unitary}\} \\ &\leq \max\left\{\sum_{i=1}^n \sigma_i(CU^*AU) : U \in M_n \text{ is unitary}\right\} \\ &\leq \max\left\{\sum_{i=1}^n \sigma_i(C)\sigma_i(U^*AU) : U \in M_n \text{ is unitary}\right\} \qquad (4.1) \\ &= \sum_{i=1}^n \sigma_i(C)\sigma_i(A). \end{aligned}$$

The first inequality is an application of the inequality (see [13, Thm. 20.b.1] or [9, Thm. 3.3.13])

$$|\text{tr } X| \leq \sigma_1(X) + \cdots + \sigma_n(X) \text{ for any } X \in M_n,$$

while the second is a special case of (3.3).

Suppose A and C are both positive semidefinite. Then there are unitary matrices $U_1, U_2 \in M_n$ such that

$$A = U_1 \operatorname{diag}(\sigma(A))\, U_1^* \quad \text{and} \quad C = U_2 \operatorname{diag}(\sigma(C))\, U_2^*.$$

The choice $U \equiv U_1 U_2^*$ in (4.1) then shows that the preceding inequalities are equalities in this case. Hence for positive semidefinite $C \in M_n$ and any $A \in M_n$ we have

$$r_C(|A|) = \sum_{i=1}^{n} \sigma_i(C)\sigma_i(A) \geq r_C(A).$$

Thus, Theorem 3.4 guarantees that whenever $C \in M_n$ is a nonscalar positive semidefinite matrix the unitary similarity invariant (but not unitarily invariant when $n > 1$) norm $r_C(\cdot)$ on M_n satisfies (1.1). The numerical radius is an example of such a norm.

Example 4.4. The *Hadamard operator norm* $\|\cdot\|_H$ on $M_{m,n}$ is

$$\|A\|_H \equiv \max\{\sigma_1(A \circ B) : B \in M_{m,n} \text{ and } \sigma_1(B) = 1\}.$$

Although $\|\cdot\|_H$ is not unitarily invariant, it satisfies not only (1.1) [1, §5] but also a rectangular version of (3.5) [15, §7.7] : for any $A \in M_{m,n}$

$$\|A\|_H = \min\{(\|B^*B\|_H \|C^*C\|_H)^{1/2} : B, C \in M_{m,n},\ B^*C = A\}. \tag{4.2}$$

If $A \succeq 0$ then it is known that $\|A\|_H = \max\{a_{ii} : i = 1, \cdots, n\}$. For general $A \in M_{m,n}$, however there is no known explicit formula to calculate $\|A\|_H$, so (4.2) may provide a useful bound, or may provide the basis for a practical algorithm.

THEOREM 4.5. *Let $N_1(\cdot)$ and $N_2(\cdot)$ be given norms on M_n that satisfy the inequality* (1.1) *and let $\alpha \in [0,1]$ be given. Then $N(\cdot) \equiv \alpha N_1(\cdot) + (1-\alpha)N_2(\cdot)$ also satisfies* (1.1), *so the set of norms satisfying* (1.1) *is a convex set that does not include all norms and is strictly larger than the set of unitarily invariant norms.*

Proof. Since any convex combination of norms is a norm, we need only show that $N(\cdot)$ satisfies (1.1). Compute

$$\begin{aligned}
N(A^*B)^2 &= [\alpha N_1(A^*B) + (1-\alpha)N_2(A^*B)]^2 \\
&= \alpha^2 N_1^2(A^*B) + 2\alpha(1-\alpha)N_1(A^*B)N_2(A^*B) + (1-\alpha)^2 N_2^2(A^*B) \\
&\leq \alpha^2 N_1(A^*A)N_1(B^*B) \\
&\quad +2\alpha(1-\alpha)([N_1(A^*A)N_1(B^*B)][N_2(A^*A)N_2(B^*B)])^{1/2} \\
&\quad +(1-\alpha)^2 N_2(A^*A)N_2(B^*B) \\
&= [\alpha N_1(A^*A) + (1-\alpha)N_2(A^*A)]\,[\alpha N_1(B^*B) + (1-\alpha)N_2(B^*B)] \\
&= N(A^*A)N(B^*B).
\end{aligned}$$

□

These ideas suggest a way to generate new norms on M_n. A *prenorm* is a continuous, homogeneous, positive function on a real or complex vector space; it does not necessarily satisfy the triangle inequality.

THEOREM 4.6. *Let $\|\cdot\|$ be a given norm on M_n, and define $N(A) \equiv \|A^*A\|^{1/2}$. Then $N(\cdot)$ is always a prenorm on M_n. If the norm $\|\cdot\|$ also satisfies the inequality*

$$\|A^*B\|^2 \leq \|A^*A\|\,\|B^*B\| \quad \textit{for all } A, B \in M_{m,n}, \tag{4.3}$$

then $N(\cdot)$ is a norm on M_n. In particular, $N(A) \equiv \|A^*A\|^{1/2}$ is a unitarily invariant norm on M_n if $\|\cdot\|$ is a unitarily invariant norm, or if $\|\cdot\|$ is a unitary similarity invariant norm such that $\|A\| \leq \| |A| \|$ for all $A \in M_n$, where $|A| \equiv (A^*A)^{1/2}$.

Proof. The function $N(A) \equiv \|A^*A\|^{1/2}$ is clearly positive, homogeneous, and continuous for any norm $\|\cdot\|$, so it is always a prenorm on M_n. It is a straightforward computation to show that $N(\cdot)$ satisfies the triangle inequality if it satisfies the inequality (4.3). □

If we choose for $\|\cdot\|$ the spectral norm, the trace norm ($\|A\| \equiv \operatorname{tr} |A|$), the numerical radius, the l_∞ norm, or the Hadamard operator norm, then the respective norms $N(A) \equiv (\|A^*A\|)^{1/2}$ are the spectral norm, the Frobenius norm ($N(A) \equiv [\operatorname{tr} A^*A]^{1/2}$), the spectral norm, and $N(A) \equiv$ the largest Euclidean column length in the last two cases.

See Example 4.13 for a unitary similarity invariant norm that does not satisfy the monotonicity condition at the end of Theorem 4.6.

We have the following analog of Theorem 4.6 for vector norms on C^n. Its proof is very similar to that of Theorem 4.6.

THEOREM 4.7. *Let $\|\cdot\|$ be a norm on C^n such that $\|z\| \leq \| |z| \|$ for all $z = [z_i] \in C^n$, where $|z| \equiv [|z_i|]$. Then $\nu(x) \equiv (\|x \circ \bar{x}\|)^{1/2}$ is an absolute norm on C^n.*

We will now consider the converses of some of the results proved so far. First we characterize the norms $\|\cdot\|$ on C^n such that $\| |\cdot| \|$ is also a norm, and the unitary similarity invariant norms $\|\cdot\|$ on M_n such that $\| |\cdot| \|$ is a norm on M_n.

THEOREM 4.8. *Let $\|\cdot\|$ be a given norm on C^n, and let $|x| \equiv [|x_i|]$ for all $x = [x_i] \in C^n$. Then $\nu(\cdot) \equiv \| |\cdot| \|$ is a norm if and only if*

$$\|u\| \leq \|v\| \quad \text{whenever} \quad u, v \in R_+^n \quad \text{and} \quad u \leq v. \tag{4.4}$$

This result is Theorem 5 in [2] (see also [8, Thm. 5.5.10]), where norms that satisfy the condition (4.4) are referred to as *monotone on the positive orthant.*

Note that (2.8) provides a *sufficient* condition for $\| |\cdot| \|$ to be a norm on C^n (Lemma 2.4), while the condition (4.4) is both necessary and sufficient. In Example 4.10, we show that the condition (4.4) is strictly weaker than (2.8).

THEOREM 4.9. *Let $\|\cdot\|$ be a given unitary similarity invariant norm on M_n, and let $|A| \equiv (A^*A)^{1/2}$ for $A \in M_n$. Then $N(A) \equiv \| |A| \|$ is always a unitarily invariant function on M_n, and it is a norm on M_n if and only if*

$$\|X\| \leq \|Y\| \quad \text{whenever} \quad X, Y \in H_n \quad \text{and} \quad 0 \preceq X \preceq Y. \tag{4.5}$$

Proof. The unitary invariance of $N(\cdot)$ is clear. If $N(\cdot)$ is a norm then it is unitarily invariant and agrees with $\|\cdot\|$ on the positive semidefinite matrices. Inequality (4.5) is true for any unitarily invariant norm because if $0 \preceq X \preceq Y$, then $\sigma_i(X) \leq \sigma_i(Y)$ for $i = 1, \cdots, n$; in particular, the singular values of X are weakly majorized by those of Y. Conversely, let $A, B \in M_n$ be given and assume (4.5). By the matrix-valued triangle inequality, there are unitary $U, V \in M_n$ such that

$$|A + B| \preceq U|A|U^* + V|B|V^*$$

and hence (4.5), the ordinary triangle inequality, and the unitary similarity invariance of $\|\cdot\|$ give

$$\begin{aligned} N(A+B) &= \| |A+B| \| \\ &\leq \|U|A|U^* + V|B|V^*\| \end{aligned}$$

$$\begin{aligned} &\leq \|U|A|U^*\| + \|V|B|V^*\| \\ &= \| \, |A| \, \| + \| \, |B| \, \| \\ &= N(A) + N(B). \end{aligned}$$

Since positivity and homogeneity are clear, it follows that $N(\cdot)$ is a norm. □

In Example 4.13 we show that there are unitary similarity invariant norms that do not satisfy the monotonicity condition (4.5).

As we might suspect from Theorem 4.8, the converses of Lemma 2.4 and Theorem 4.7 are not true. There are norms on C^n that satisfy the condition (4.4) but not (2.8).

Example 4.10. Consider the function $\|\cdot\|$ defined on C^2 by

$$\|x\| \equiv \max\{|x_1|, |x_2|, |x_1 - x_2|\}.$$

Then $\|\cdot\|$ is easily shown to be a norm, but it does not satisfy the monotonicity condition $\|x\| \leq \| \, |x| \, \|$. Consider, for example, $x = [1, -1]^T$. However, $\nu_1(x) \equiv \|x \circ \bar{x}\|^{1/2}$ and $\nu_2(x) \equiv \| \, |x| \, \|$ are both norms since

$$\nu_1(x) = \nu_2(x) = \|x\|_\infty = \max\{|x_1|, |x_2|\}.$$

Thus, $\|\cdot\|$ is a norm on C^2 that satisfies the condition (4.4) but not (2.8).

Similarly, we might suspect from Theorem 4.9 that the converses of Lemma 3.3 and Theorem 4.6 are also false. There are unitary similarity invariant norms on M_n that satisfy (4.5), but not (3.7). To construct one we first prove Lemma 4.11.

LEMMA 4.11. *Let $\|\cdot\|$ be a given norm on the real vector space H_n. Then the function $\|\cdot\|' : M_n \to R_+$ defined by*

$$\|A\|' \equiv \max\{\|\tfrac{1}{2}(\alpha A + \bar{\alpha} A^*)\| : \alpha \in C \text{ and } |\alpha| = 1\} \tag{4.6}$$

is a self-adjoint norm on M_n that agrees with $\|\cdot\|$ on H_n, i.e., $\|A\|' = \|A^*\|'$ for all $A \in M_n$ and $\|A\|' = \|A\|$ for all $A \in H_n$. If the given norm $\|\cdot\|$ is unitary similarity invariant on H_n, then the norm $\|\cdot\|'$ is also unitary similarity invariant on M_n.

Furthermore, the norm $\|\cdot\|'$ is minimal in the following sense: if $N(\cdot)$ is any self-adjoint norm on M_n that agrees with $\|\cdot\|$ on H_n, then $N(A) \geq \|A\|'$ for all $A \in M_n$.

Proof. The positivity and homogeneity of $\|\cdot\|'$ follow from the positivity and homogeneity of $\|\cdot\|$. For the triangle inequality, take $A, B \in M_n$ and compute

$$\begin{aligned} \|A+B\|' &= \max\{\|\tfrac{1}{2}[\alpha(A+B) + \bar{\alpha}(A+B)^*]\| : \alpha \in C \text{ and } |\alpha| = 1\} \\ &\leq \max\{\|\tfrac{1}{2}(\alpha A + \bar{\alpha} A^*)\| + \|\tfrac{1}{2}(\alpha B + \bar{\alpha} B^*)\| : \alpha \in C \text{ and } |\alpha| = 1\} \\ &\leq \max\{\|\tfrac{1}{2}(\alpha A + \bar{\alpha} A^*)\| : \alpha \in C \text{ and } |\alpha| = 1\} \\ &\quad + \max\{\|\tfrac{1}{2}(\alpha B + \bar{\alpha} B^*)\| : \alpha \in C \text{ and } |\alpha| = 1\} \\ &= \|A\|' + \|B\|'. \end{aligned}$$

We now know that $\|\cdot\|'$ is a norm on M_n, and the fact that $\|A\|' = \|A^*\|'$ is immediate from the definition, as is the assertion about unitary similarity invariance. Suppose that $A \in H_n$ and $|\alpha| = 1$. Then

$$\|\tfrac{1}{2}(\alpha A + \bar{\alpha} A^*)\| = \|\tfrac{1}{2}(\alpha A + \bar{\alpha} A)\| = \|(\operatorname{Re} \alpha) A\| = |\operatorname{Re} \alpha| \, \|A\| \leq \|A\|$$

with equality for $\alpha = \pm 1$. This shows that $\|A\|' = \|A\|$ whenever $A \in H_n$.

Finally, consider the assertion about the minimality of $\|\cdot\|'$. Let $N(\cdot)$ be a given norm on M_n such that $N(A) = N(A^*)$ for all $A \in M_n$ and $N(A) = \|A\|$ for all $A \in H_n$. Then for any $\alpha \in C$ with $|\alpha| = 1$ we have

$$N(A) = \tfrac{1}{2}[N(\alpha A) + N((\alpha A)^*)] \geq N(\tfrac{1}{2}[\alpha A + \bar{\alpha} A^*]) = \|\tfrac{1}{2}[\alpha A + \bar{\alpha} A^*]\|$$

and hence

$$N(A) \geq \max\{\|\tfrac{1}{2}[\alpha A + \bar{\alpha} A^*]\| : \alpha \in C \text{ and } |\alpha| = 1\} = \|A\|'. \qquad \square$$

Example 4.12. We shall exhibit a norm that shows that the implications in Theorem 4.6 and Lemma 3.3 cannot be reversed. Let $\lambda_1(X) \geq \lambda_2(X)$ denote the algebraically ordered eigenvalues of $X \in H_2$. Define the function $\|\cdot\| : H_2 \to R_+$ by

$$\|X\| = \max\{|\lambda_1(X)|, |\lambda_2(X)|, |\lambda_1(X) - \lambda_2(X)|\}.$$

Note the similarity between this function and the norm on C^2 defined in Example 4.10. We can easily verify that the function $\|\cdot\|$ is a norm on the real vector space H_2: Either use the Weyl inequalities [8, Thm. 4.3.1]

$$\begin{aligned} \lambda_1(A+B) &\leq \lambda_1(A) + \lambda_1(B) \\ \lambda_2(A+B) &\geq \lambda_2(A) + \lambda_2(B) \end{aligned}$$

for the eigenvalues of any $A, B \in H_2$, or note that the function $\|X\|$ is a Schur-convex function of the eigenvalues of X and apply Theorem 3 of [11]. Note that $\|A\| = \lambda_1(A) = \|\!|A|\!\|_2$ if $A \in H_2$ is positive semidefinite.

Let the norm $\|\cdot\|'$ be derived from $\|\cdot\|$ as defined in (4.6), and use Lemma 4.11 to observe that $\|\cdot\|'$ is a unitary similarity invariant norm on M_2. For $X \in M_2$ set

$$N(X) \equiv (\|X^*X\|')^{1/2} \text{ and } \nu(X) \equiv \| \, |X| \, \|', \text{ where } |X| \equiv (X^*X)^{1/2}.$$

Then

$$N(X) = (\|X^*X\|')^{1/2} = (\|X^*X\|)^{1/2} = (\|\!|X^*X|\!\|_2)^{1/2} = \|\!|X|\!\|_2$$

and

$$\nu(X) = \| \, |X| \, \|' = \| \, |X| \, \| = \|\!| \, |X| \, |\!\|_2 = \|\!|X|\!\|_2.$$

Thus, both $N(X)$ and $\nu(X)$ are unitarily invariant norms on M_2. However, the choice

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

shows that the norm $\|\cdot\|'$ satisfies neither (4.5) nor (3.7) since $\|A^*A\|' = \|B^*B\|' = 1$, $\|A^*B\|' = 2$, $\|B\|' = 2$, and $\| \, |B| \, \|' = 1$.

Example 4.13. There is a unitary similarity invariant norm on M_2 that satisfies neither (3.7) nor (4.5). Let

$$C = \begin{pmatrix} 2 & 0 \\ 0 & -1 \end{pmatrix}$$

and consider the unitary similarity invariant norm $r_C(\cdot)$ on M_2. There is no unitary U for which $U|C|U^*$ is a scalar multiple of C. By the Cauchy–Schwarz inequality for the Frobenius inner product and the definition of $r_C(\cdot)$ we have

$$|\text{tr } CU|C|U^*|^2 < (\text{tr } C^2)(\text{tr } U|C|^2U^*) = (\text{tr } C^2)^2 \leq r_C^2(C)$$

for any unitary U, and hence $r_C(|C|) < r_C(C)$. Thus, the unitary similarity invariant norm $r_C(\cdot)$ does not satisfy (3.7). To see that $r_C(\cdot)$ does not satisfy (4.5) either, set

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and note that $0 \preceq X \preceq Y$, but $r_C(X) = 2 > 1 = r_C(Y)$.

5. Open questions. In Theorem 3.4 we characterized the unitary similarity invariant norms on M_n for which (1.1) holds, and in Example 4.2 we showed that the norm $\|\cdot\|_\infty$, which is not unitary similarity invariant, satisfies (1.1). Is there a useful characterization of the norms on $M_{m,n}$ that satisfy (1.1) ?

In Corollary 3.2 we have shown that for any unitarily invariant norm $\|\cdot\|$ on M_n and all $A \in M_n$,

$$\|A\| = \min\{\|B^*B\|^{1/2}\|C^*C\|^{1/2} : B, C \in M_n \text{ and } B^*C = A\}. \tag{5.1}$$

If $\|\cdot\|$ is a unitary similarity invariant norm, then the right-hand side of (5.1) is a unitarily invariant function of $A \in M_n$. Thus, a unitary similarity invariant norm satisfies (5.1) if and only if it is unitarily invariant. We showed in Example 4.4 that the nonunitarily invariant norm $\|\cdot\|_H$ also obeys (5.1). How can we characterize the norms that satisfy (5.1)?

Consider the l_p norms $\|\cdot\|_p$ on $M_{m,n}$, defined in Example 4.1. We have shown that the inequality

$$\|A^*B\|_p^2 \leq \|A^*A\|_p \, \|B^*B\|_p \quad \text{for all } A, B \in M_{m,n} \tag{5.2}$$

is false for $p \in [1, 2)$ (Example 4.1), and true for $p = 2$ (this is the Cauchy–Schwarz inequality for the Frobenius inner product). Does (5.2) hold for $p \in (2, \infty)$? This question is partially answered in [10, Ex. 4.4].

For $p \geq 1$, the Schatten p–norm on M_n is defined by

$$\|A\|_{Sp} \equiv \left(\sum_{i=1}^n \sigma_i^p(A)\right)^{1/p}.$$

If $p = 2k$ for some integer k, then the Schatten p–norm can also be defined by

$$\|A\|_{Sp} \equiv (\text{tr } (A^*A)^k)^{1/2k}.$$

From this representation it is clear that

$$\|[a_{ij}]\|_{Sp} \leq \|[|a_{ij}|]\|_{Sp} \tag{5.3}$$

whenever p is an even integer. Thus, Theorem 2.5 ensures that

$$\|A \circ \bar{B}\|_{Sp}^2 \leq \|A \circ \bar{A}\|_{Sp}\|B \circ \bar{B}\|_{Sp} \quad \text{for all } A, B \in M_n \tag{5.4}$$

whenever p is an even integer. If $n = 2$, then (5.3) holds for all $p \geq 2$. To see that (5.3) is not true for $1 \leq p < 2$, consider

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

What are the values of p for which (5.3), and hence (5.4), holds? We can show that the answer depends on n. More generally, what are the unitarily invariant norms $\|\cdot\|$ for which

$$\|[a_{ij}]\| \leq \|[|a_{ij}|]\| \text{ for all } [a_{ij}] \in M_{m,n}?$$

Note that the spectral norm and the Frobenius norm satisfy this inequality.

REFERENCES

[1] T. ANDO, R. A. HORN, AND C. R. JOHNSON, *The singular values of a Hadamard product: A basic inequality*, Linear Multilinear Algebra, 21 (1987), pp. 345–365.

[2] F. L. BAUER, J. STOER, AND C. WITZGALL, *Absolute and monotonic norms*, Numer. Math., 3 (1961), pp. 257–264.

[3] R. BHATIA, *Perturbation Bounds for Matrix Eigenvalues*, Pitman Research Notes in Mathematics 162, Longman Scientific and Technical, New York, 1987.

[4] ———, *Perturbation inequalities for the absolute value map in norm ideals of operators*, J. Oper. Theory, 19 (1988), pp. 129–136.

[5] K. FAN, *Maximum properties and inequalities for the eigenvalues of completely continuous operators*, Proc. Nat. Acad. Sci., 37 (1951), pp. 760–766.

[6] A. HORN, *On the singular values of a product of completely continuous operators*, Proc. Nat. Acad. Sci., 36 (1950), pp. 374–375.

[7] R. A. HORN AND C. R. JOHNSON, *Hadamard and conventional submultiplicativity for unitarily invariant norms on matrices*, Linear Multilinear Algebra, 20 (1987), pp. 91–106.

[8] ———, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[9] ———, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1990.

[10] R. A. HORN AND R. MATHIAS, *Cauchy-Schwarz inequalities associated with positive semidefinite matrices*, Linear Algebra Appl., to appear.

[11] C. LI AND N. TSING, *Norms that are invariant under unitary similarities and the C-numerical radii*, Linear Multilinear Algebra, 24 (1989), pp. 209–222.

[12] M. MARCUS AND M. SANDY, *Three elementary proofs of the Goldberg–Strauss theorem on numerical radii*, Linear Multilinear Algebra, 11 (1982), pp. 243–252.

[13] A. W. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization and its Applications*, Academic Press, London, 1979.

[14] K. OKUBO, *Hölder-type inequalities for Schur products of matrices*, Linear Algebra Appl., 91 (1987), pp. 13–28.

[15] V. I. PAULSEN, *Completely Bounded Maps and Dilations*, Pitman Research Notes in Mathematics 146, Longman Scientific and Technical, Harlow, 1986.

[16] R. SCHATTEN, *Norm Ideals of Completely Continuous Operators*, Springer-Verlag, Berlin, 1970.

[17] J. SCHUR, *Bermerkungen zur Theorie der beschränkten Bilinearformen mit unendlich vielen Veränderlichen*, J. Reine Angew. Math., 140 (1911), pp. 1–28.

[18] R. C. THOMPSON, *Convex and concave functions of singular values of matrix sums*, Pacific J. Math., 66 (1976), pp. 285–290.

[19] J. VON NEUMANN, *Some matrix inequalities and metrization of matric space*, Tomsk. Univ. Rev., 1 (1937), pp. 286–300. Also in John von Neumann Collected Works, A. H. Taub, ed., Vol. IV, Pergamon Press, Oxford, 1962, pp. 205-218.

[20] H. WIMMER, *Extremal problems for Hölder norms of matrices and realizations of linear systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 314–322.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 499–520, October 1990

HYPERBOLIC HOUSEHOLDER ALGORITHMS FOR FACTORING STRUCTURED MATRICES*

G. CYBENKO† AND M. BERRY‡

Abstract. Efficient algorithms for computing triangular decompositions of Hermitian matrices with small displacement rank using hyperbolic Householder matrices are derived. These algorithms can be both vectorized and parallelized. Implementations along with performance results on an Alliant FX/80, Cray X-MP/48, and Cray-2 are discussed. The use of Householder-type transformations is shown to improve performance for problems with nontrivial displacement ranks. In special cases, the general algorithm reduces to the well-known Schur algorithm for factoring Toeplitz matrices and Elden's algorithm for solving structured regularization problems. It gives a Householder formulation to the class of algorithms based on hyperbolic rotations studied by Kailath, Lev-Ari, Chun, and their colleagues for Hermitian matrices with small displacement structure. In addition, an extension to the efficient factorization of indefinite systems is described.

Key words. hyperbolic transformations, displacement rank, parallel algorithms

AMS(MOS) subject classifications. 65F05, 65F25, 65F30

1. Introduction. In this paper, we derive efficient algorithms for computing triangular decompositions of Hermitian matrices with small displacement rank using hyperbolic Householder matrices. The general algorithm we introduce reduces to, in special cases, the well-known Schur algorithm for factoring Toeplitz matrices [2], [27] and Elden's algorithm for structured regularization problems [19]. The relationship between the algorithm presented here and the class of algorithms studied by Kailath, Lev-Ari, Chun, and others for Hermitian matrices with small displacement structure [8], [9], [15], [25] is the same as the relationship between algorithms based on classical plane rotations and classical unitary Householder transformations. The operation count is slightly reduced in the complex case, but the more important performance factor is that the majority of the computation involves matrix-vector instead of vector-scalar operations thereby improving data locality, vectorization, and concurrency properties. Another important contribution of this work is the extension of previously known methods to the indefinite case.

Hyperbolic Householder matrices have recently been studied in detail by Rader and Steinhardt [31], [32] in the context of fast updating and downdating of linear least squares problems. Such matrices are natural generalizations of classical Householder matrices [23] and hyperbolic rotations that have been used over the years to implement downdating of various factorizations [13], [1], [22], [21]. The existence of Householder-like hyperbolic matrices was already known to a number of researchers as long ago as a decade [26], [12] in the context of canonical factorizations of general hyperbolic matrices.

* Received by the editors May 22, 1989; accepted for publication (in revised form) September 18, 1989.

† Center for Supercomputing Research and Development and Department of Electrical and Computer Engineering, University of Illinois, Urbana, Illinois 61801 (gc@uicsrd.csrd.uiuc.edu). This work was supported in part by the National Science Foundation under grant NSF CCR-8717942, and DCR-8619103 (Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign), the U.S. Department of Energy under grant DOE-DE-FG02-85ER25001, AT&T Corporation under grant AT&T-AFFL-67-SAMEH, and the Office of Naval Research under grant ONR N000-86-G-0202 (Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign).

‡ Center for Supercomputing Research and Development, University of Illinois, Urbana, Illinois 61801 (berry@uicsrd.csrd.uiuc.edu). This work was supported in part by the National Science Foundation under grants NSF CCR-8717942 and CCR-890003N (NCSA/Cray Research), the U.S. Department of Energy under grant DOE-DE-FG02-85ER25001, and the AT&T Corporation under grant AT&T-AFFL-67-SAMEH.

Combining these two ideas produces a class of vectorizable and parallelizable algorithms requiring $O(\alpha n^2)$ sequential operations for an order n Hermitian matrix with displacement rank α. They are not the most efficient algorithms from the point of view of abstract complexity theory since $O(n \log^2 n)$ sequential methods are known for Toeplitz matrices [2], [28], [11] and $O(\log^2 n)$ parallel methods using $O(n \log^2 n)$ processors [29]. On the other hand, the above-mentioned theoretical results are only asymptotically more efficient than the algorithm presented here.

The algorithm presented here is conceptually easy to understand. Order n matrices with displacement rank α have simple representations in terms of α n-vectors called *generators* [8]. This representation involves inner products with respect to a hyperbolic norm. These facts have been noted and developed by Kailath, Lev-Ari, Chun, and their colleagues over the years, leading to a general class of algorithms using hyperbolic rotations to reduce the generators while preserving the hyperbolic inner product. Our use of hyperbolic Householders merely replaces a sequence of hyperbolic rotations when eliminating elements in a row or column. Thus the algorithm we present is based on precisely the same ideas as used in say [8], but it substitutes a single hyperbolic Householder matrix for a sequence of α hyperbolic rotations.

Although we give examples where α is rather small, namely, two and four, there are applications where α can become arbitrarily large in absolute terms while remaining small relative to the size n of the underlying matrix. Such matrices arise, for example, in atmospheric light scattering studies [18], [20] where Hermitian block Toeplitz matrices are encountered. It is a well-known fact that block Toeplitz systems have small displacement rank [8].

In addition to a derivation of these algorithms, we present performance results on machines such as the Alliant FX/80, Cray X-MP/48, and Cray-2. The performance of the classical Levinson and Schur algorithms for Toeplitz systems has recently been studied on Alliant FX/8 and Cray 1S vector machines [14]. In light of the considerable recent interest in Toeplitz and related algorithms as well as possibilities for dedicated hardware implementations, perhaps most notably systolic arrays, we hope that these results will serve as standards by which to measure the performance of such specialized hardware.

The paper is organized as follows. Sections 2 and 3 briefly review the main ideas behind hyperbolic Householder matrices and matrices with small displacement ranks, respectively. Section 4 contains a derivation of the algorithm. Section 5 has examples showing how the algorithm generalizes Schur's algorithm and partial computation of the QR factorization of a Toeplitz matrix. The performance results presented in § 6 are followed by a summary in § 7.

2. Hyperbolic Householder transformations. Householder transformations have come to play a major role in modern numerical linear algebra. They are used to introduce large numbers of zeros into matrices with the typical goal of using orthogonal transformations to reduce dense matrices into triangular or Hessenberg form [23].

Hyperbolic Householder matrices, which have recently been studied by Rader and Steinhardt [31], [32], play a key role in our algorithm. All of the material in this section is a review of the material in [31] and [32] and is included for the sake of completeness.

To define hyperbolic Householder transformations, we need a few simple definitions. First, let W be a Hermitian idempotent matrix, that is, W satisfies

$$W^2 = I_n, \qquad W^* = W$$

so that W is unitary as well. All of the interesting examples currently known involve a matrix $W = (w_{ij})$ that satisfies $w_{ii} = \pm 1$. In fact, it is a simple exercise to verify that every Hermitian idempotent matrix is of the form $Q^* W Q$ where Q is unitary (that is,

$Q^*Q = I$) and W is of the above form, namely, with $w_{ii} = \pm 1$. We note that throughout this paper it should be assumed that $W = (w_{ij})$ with $w_{ii} = \pm 1$.

A matrix U is said to be W-unitary if it satisfies the equation

$$U^*WU = W.$$

Clearly, the class of I_n-unitary matrices corresponds to the traditional notion of unitary matrices.

It is easy to check that W-unitary matrices form a multiplicative group, that is, I_n is W-unitary, the product of W-unitary matrices is W-unitary, and inverses of W-unitary matrices are W-unitary. In particular, every W-unitary matrix is invertible. We will use the terms W-unitary and hyperbolic interchangeably.

We now define the notion of a hyperbolic Householder matrix. Let x be an n-vector for which $x^*Wx \neq 0$. Define

$$U_x = W - 2\frac{xx^*}{x^*Wx}. \tag{1}$$

All such matrices U_x are W-unitary. To see this, we merely use the definition (1) to get

$$\begin{aligned} U_x^* WU_x &= \left(W - 2\frac{xx^*}{x^*Wx}\right)W\left(W - 2\frac{xx^*}{x^*Wx}\right) \\ &= W^3 - 2\frac{W^2xx^*}{x^*Wx} - 2\frac{xx^*W^2}{x^*Wx} + 4\frac{xx^*Wxx^*}{(x^*Wx)^2} \\ &= W \end{aligned}$$

since $W^2 = I$.

Just as classical Householder matrices can be defined to transform one vector to another of the same Euclidean norm, hyperbolic Householders can be defined to map one vector to another providing that they have the same nonzero hyperbolic norm. Thus, suppose that a and b are two vectors with the same hyperbolic norm, $a^*Wa = b^*Wb$, and define

$$x = Wa + \sigma b$$

where

$$\sigma = \begin{cases} \operatorname{sign}(a^*Wa)b^*a/|a^*b| & \text{if } a^*b \neq 0, \\ \operatorname{sign}(a^*Wa) & \text{otherwise.} \end{cases}$$

Here,

$$\operatorname{sign}(\theta) = \begin{cases} +1 & \text{if } \theta \geqq 0, \\ -1 & \text{if } \theta < 0. \end{cases}$$

Note that

$$\begin{aligned} x^*Wx &= a^*Wa + b^*Wb + \sigma a^*b + \bar{\sigma} b^*a \\ &= 2(a^*Wa + \sigma a^*b) \\ &= 2(a^*Wa + \bar{\sigma} b^*a) \\ &= 2 \operatorname{sign}(a^*Wa)(|a^*Wa| + |a^*b|), \end{aligned}$$

so the choice of σ not only makes x^*Wx real, but maximizes its magnitude as well.

Assuming for the moment that x^*Wx thus defined is nonzero, we get

$$\begin{aligned} U_x a &= \left(W - 2\frac{xx^*}{x^*Wx}\right)a \\ &= Wa - 2\frac{(Wa+\sigma b)(a^*Wa+\bar{\sigma}b^*a)}{2(a^*Wa+\sigma a^*b)} \\ &= -\sigma b, \end{aligned}$$

so that U_x maps a to $-\sigma b$ or, equivalently, $-U_x/\sigma$ is W-unitary and maps a to b. Moreover, this demonstrates that U_x is undefined if and only if both a^*Wa and a^*b are zero. Thus, a vector with nonzero hyperbolic norm can be mapped to any other vector with the same hyperbolic norm by a hyperbolic Householder matrix. However, a vector a with zero hyperbolic norm can only be mapped to another vector b with zero hyperbolic norm for which $a^*b \neq 0$.

One important property that hyperbolic Householder matrices share with classical Householder matrices is the fact that if the jth coordinate of x is zero, $x_j = 0$, then $U_x y$ has the same jth component as y except for a possible sign change. This property is noteworthy because it plays a key role in the algorithm we describe below.

Rader and Steinhardt have computed the condition number of such hyperbolic Householder matrices [31], [32]. Specifically, they show that the largest eigenvalue of U_x has magnitude

$$\rho = |\zeta| + \sqrt{\zeta^2 - 1}$$

where $\zeta = x^*x/x^*Wx$ and the smallest eigenvalue has magnitude ρ^{-1}. Hence, the condition number is ρ^2. As Rader and Steinhardt point out, this quantity is easily computed as a byproduct of forming and applying U_x so that monitoring the condition number of hyperbolic Householders is easy to accomplish. Recently, Bojanczyk and Steinhardt have developed more stable versions of hyperbolic Householder matrices in downdating problems [5], but their applicability to the problem studied here remains unresolved.

3. Displacement rank. Displacement rank is an idea introduced some years ago by Kailath, Kung, and Morf [25] to quantify the properties of matrices that make them Toeplitz or close to being Toeplitz in a structural sense. Toeplitz matrices are matrices that are constant along diagonals parallel to the main diagonal, i.e., $T = (t_{ij})$ is Toeplitz if t_{ij} depends only on $i - j$. We now briefly review the main points of the theory.

Let Z denote the unit shift matrix $Z = (z_{ij})$:

$$z_{ij} = \delta_{i-j-1}$$

where δ_k is the Kronecker delta function. Multiplication on the right by Z has the effect of shifting the columns of a matrix to the right by one column, replacing the first column by the zero column, and shifting the last column out. Multiplication by Z^* on the left has the same effect on the rows of a matrix: it shifts rows down by one, introducing a zero row into the first row and shifting the last row out. Thus, if

$$A = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & p & q \end{bmatrix} \quad \text{then } Z^*AZ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a & b & c \\ 0 & e & f & g \\ 0 & i & j & k \end{bmatrix}.$$

The fundamental concept in displacement rank involves the difference $A - Z^*AZ$, and we now formally state a definition.

DEFINITION 1. The *displacement rank* of the matrix A is the rank of the difference $A - Z^*AZ$.

If A is a Hermitian, Toeplitz matrix, then (without loss of generality, we use a 4×4 example for illustration)

$$A = \begin{bmatrix} a & b & c & d \\ b^* & a & b & c \\ c^* & b^* & a & b \\ d^* & c^* & b^* & a \end{bmatrix} \quad \text{and} \quad Z^*AZ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a & b & c \\ 0 & b^* & a & b \\ 0 & c^* & b^* & a \end{bmatrix}$$

so the difference $A - Z^*AZ$ is zero except for the first row and column. In particular, $A - Z^*AZ$ has rank 2 in general since

$$A - Z^*AZ = \begin{bmatrix} a & b & c & d \\ b^* & 0 & 0 & 0 \\ c^* & 0 & 0 & 0 \\ d^* & 0 & 0 & 0 \end{bmatrix} = \pm u^*u \pm v^*v$$

for some row vectors u and v where the last equality follows from the eigendecomposition of the Hermitian matrix $A - Z^*AZ$. Our nonstandard use of row vectors in such expressions merely simplifies notation in the subsequent derivation.

The key result for displacement ranks is the following theorem due to Kailath, Kung, and Morf.

THEOREM 1 (Kailath, Kung, and Morf [25]). *The Hermitian matrix A has displacement rank α if and only if α is the smallest integer for which A can be written as*

$$A = \sum_{j=1}^{\alpha} \varepsilon_j G^*(x_j) G(x_j) \tag{2}$$

where $\varepsilon_j = \pm 1$ and for an n-dimensional row vector y, $G(y)$ is an upper triangular Toeplitz matrix given by

$$G(y) = \begin{bmatrix} y_1 & y_2 & \cdot & \cdot & y_n \\ 0 & y_1 & y_2 & \cdot & \cdot \\ 0 & 0 & y_1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & y_2 \\ 0 & \cdot & \cdot & \cdot & y_1 \end{bmatrix}.$$

The original proof is due to Kailath, Kung, and Morf and can be found in [25], for instance. We sketch the main points for completeness. If A has the form of (2), then

$$A - Z^*AZ = \sum_{j=1}^{\alpha} \varepsilon_j (G(x_j))^* G(x_j) - \sum_{j=1}^{\alpha} \varepsilon_j (G(x_j)Z)^* G(x_j)Z = \sum_{j=1}^{\alpha} \varepsilon_j x_j^* x_j$$

as claimed. Conversely, consider the telescoping sum

$$A = A - \left(\sum_{i=1}^{n} Z^{*i} A Z^i - Z^{*i} A Z^i \right) = \sum_{i=0}^{n} Z^{*i} (A - Z^*AZ) Z^i \tag{3}$$

where we use the fact that $Z^{*(n+1)} A Z^{n+1} = 0$. Now if A has displacement rank α, then

$$A - Z^*AZ = \sum_{j=1}^{\alpha} \varepsilon_j x_j^* x_j$$

so that substituting the above into (3) gives

$$A = \sum_{j=1}^{\alpha} \varepsilon_j \sum_{i=0}^{n} Z^{*i} x_j^* x_j Z^i = \sum_{j=1}^{\alpha} \varepsilon_j G(x_j)^* G(x_j)$$

as desired.

This result illustrates that matrices structurally close to Toeplitz matrices have a simple parsimonious representation in terms of a relatively small number of vectors, called *generators* of the matrix. Kailath and his colleagues have derived efficient algorithms based on hyperbolic and classical rotations [8] for factoring matrices expressed in terms of their generators. It should be noted that given a general matrix, the determination of its displacement rank would require about as much work as solving a linear system involving that matrix. Thus for a given problem, we need to know a priori the displacement rank and the generators of the matrix. Fortunately, such generators are easy to obtain for an important class of matrices that occur naturally in applications. We give examples in § 5.

The relationship between matrices with small displacement rank and hyperbolic inner products arises by rewriting the basic identity (2). Specifically, let

$$W = \operatorname{diag}(\varepsilon_1 I_n, \varepsilon_2 I_n, \cdots, \varepsilon_\alpha I_n)$$

and

$$G = \begin{bmatrix} G(x_1) \\ G(x_2) \\ \vdots \\ G(x_\alpha) \end{bmatrix}$$

so that W is unitary, idempotent, and Hermitian, and

$$G^* W G = A.$$

The basic idea behind the use of hyperbolic Householder transformations for factoring A can be described as follows. Suppose that U is a W-unitary matrix that triangularizes G (if such a U exists)

$$UG = \begin{bmatrix} R \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where R is upper triangular. Then

$$A = G^* W G = G^* U^* W U G = [R^* 0 \cdots 0] W \begin{bmatrix} R \\ 0 \\ \vdots \\ 0 \end{bmatrix} = R^* R$$

is a Cholesky factorization of A.

The algorithm described in the next section uses hyperbolic Householder transformations to compute such a factorization efficiently. A sequence of hyperbolic Householders is used to triangularize G, and the product of this sequence of Householders is precisely U. Strictly speaking, such a factorization exists only if the underlying A is positive definite so part of the algorithm we derive involves handling the case where A is indefinite.

4. A hyperbolic Householder algorithm. The basic idea behind the algorithm that we present is derived from known algorithms that use hyperbolic rotations instead of hyperbolic Householder matrices [8]. The advantages of using hyperbolic Householders is that their construction greatly simplifies a number of special cases and allows more parallelism and vectorization on large problems.

In order to illustrate the key ideas, we start with a simple example of displacement rank 3 and avoid dealing with anomalies until after the basics are understood. Suppose that $A = G^*WG$ with

$$G = \begin{bmatrix} a & b & c \\ 0 & a & b \\ 0 & 0 & a \\ r & s & t \\ 0 & r & s \\ 0 & 0 & r \\ u & v & w \\ 0 & u & v \\ 0 & 0 & u \end{bmatrix}$$

and $W = \operatorname{diag}(\varepsilon_1 I_3, \varepsilon_2 I_3, \varepsilon_3 I_3)$ with $\varepsilon_i = \pm 1$.

Now construct, according to the previous discussion, a hyperbolic Householder transformation, U_x, so that

$$U_x \begin{bmatrix} a \\ 0 \\ 0 \\ r \\ 0 \\ 0 \\ u \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{a} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

subject, of course, to the condition that $\varepsilon_1 |a|^2 + \varepsilon_2 |r|^2 + \varepsilon_3 |u|^2 = \varepsilon_1 |\tilde{a}|^2$. Then

$$U_x G = \tilde{G}_1 = \begin{bmatrix} \tilde{a} & \tilde{b} & \tilde{c} \\ 0 & \varepsilon_1 a & \varepsilon_1 b \\ 0 & 0 & \varepsilon_1 a \\ 0 & \tilde{s} & \tilde{t} \\ 0 & \varepsilon_2 r & \varepsilon_2 s \\ 0 & 0 & \varepsilon_2 r \\ 0 & \tilde{v} & \tilde{w} \\ 0 & \varepsilon_3 u & \varepsilon_3 v \\ 0 & 0 & \varepsilon_3 u \end{bmatrix}$$

and $\tilde{G}_1^* W \tilde{G}_1 = A$. The fact that x has zero coordinates in entries 2, 3, 5, 6, 8, and 9 implies that $U_x y$ for any vector y has the same 2, 3, 5, 6, 8, and 9 entries as y save for a scaling by one of the ε_j. Moreover, we note that the factors ε_j in $\tilde{G}_1$ can be dropped since they arise from a diagonal, unitary scaling of the rows of the matrix and such scalings

commute with W. Thus defining $\tilde{G}$ according to

$$\tilde{G} = \begin{bmatrix} \tilde{a} & \tilde{b} & \tilde{c} \\ 0 & a & b \\ 0 & 0 & a \\ 0 & \tilde{s} & \tilde{t} \\ 0 & r & s \\ 0 & 0 & r \\ 0 & \tilde{v} & \tilde{w} \\ 0 & u & v \\ 0 & 0 & u \end{bmatrix}$$

still gives $\tilde{G}^* W \tilde{G} = A$. We now duplicate the action of U_x on rows 2, 5, and 8 and then on rows 3, 6, and 9 using U_{Z^*x} and $U_{Z^{*2}x}$, respectively, leading to the matrix

$$G_1 = U_{Z^{*2}x} U_{Z^*x} U_x G = \begin{bmatrix} \tilde{a} & \tilde{b} & \tilde{c} \\ 0 & \tilde{a} & \tilde{b} \\ 0 & 0 & \tilde{a} \\ 0 & \tilde{s} & \tilde{t} \\ 0 & 0 & \tilde{s} \\ 0 & 0 & 0 \\ 0 & \tilde{v} & \tilde{w} \\ 0 & 0 & \tilde{v} \\ 0 & 0 & 0 \end{bmatrix}.$$

This describes one iteration of the basic algorithm, and there are three fundamental observations to be made:

1. The first row of G_1 is the first row of the Cholesky factor of A;
2. The entries of G_1 can be easily inferred from the entries of $U_x G$ so that only the first step needs to be explicitly carried out; and
3. Isolating rows 2, 3, 4, 5, 7, and 8 of G_1 leads to a matrix of the exact same structure as G so that these steps can be repeated iteratively.

These observations indicate that we never need to work with the entire matrix G but only with the submatrix of generators

$$G_0' = \begin{bmatrix} a & b & c \\ r & s & t \\ u & v & w \end{bmatrix} \tag{4}$$

and the reduced signature matrix diag $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$.

Let us now repeat the above step using the matrix of generators G_0' this time. It is easy to see that we require a hyperbolic Householder transformation U_x that takes $[a\ r\ u]^T$ to $[\tilde{a}\ 0\ 0]^T$ so that U_x is hyperbolic with respect to diag $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$. Then we have

$$U_x G_0' = \begin{bmatrix} \tilde{a} & \tilde{b} & \tilde{c} \\ 0 & \tilde{s} & \tilde{t} \\ 0 & \tilde{v} & \tilde{w} \end{bmatrix}$$

and the first row $[\tilde{a}\ \tilde{b}\ \tilde{c}]$ is the first row of the triangular factor we seek to compute. The next iteration is applied to the matrix with this first row shifted to the right by

one column:

$$G_1' = \begin{bmatrix} 0 & \tilde{a} & \tilde{b} \\ 0 & \tilde{s} & \tilde{t} \\ 0 & \tilde{v} & \tilde{w} \end{bmatrix},$$

which simulates the next step of the previously described procedure on rows 2, 3, 4, 5, 7, and 8. Then we proceed to reduce the second column in a similar way.

This simple example illustrates the general principle behind the hyperbolic Householder Algorithm for matrices with small displacement ranks. However, a few anomalous cases need to be addressed.

First, in the algorithm we have presented, the hyperbolic Householder matrix is determined by mapping a column of the generator matrix to a multiple of the first standard basis element e_1. For this to be possible, the sign of the hyperbolic norm of e_1 must be the same as the signs[1] of all columns involved in the reduction; it can be shown that this only holds for strictly positive or strictly negative definite matrices as determined by the generators. In fact, this is the only case described by Chun, Kailath, and Lev-Ari in [8]. One remedy is to determine the Householder matrix by mapping a desired column to one of the standard basis elements with hyperbolic norm of the same sign. Thus, in the example above, if the column $[a\ b\ c]^T$ has negative hyperbolic norm while $W = \text{diag}(1, -1, -1)$, then we have to map the column to a multiple of e_2 or e_3—it makes no difference which is selected providing the resulting U_x is defined. Selecting e_2, we get

$$U_x G_0' = \begin{bmatrix} 0 & \tilde{b} & \tilde{c} \\ \tilde{r} & \tilde{s} & \tilde{t} \\ 0 & \tilde{v} & \tilde{w} \end{bmatrix}$$

and $[\tilde{r}\ \tilde{s}\ \tilde{t}]$ is now the first row of the triangular factor of A with the sign -1, which is stored as the first diagonal entry of a diagonal scaling matrix W'. The next iteration operates on the matrix

$$U_x G_0' = \begin{bmatrix} 0 & \tilde{b} & \tilde{c} \\ 0 & \tilde{r} & \tilde{s} \\ 0 & \tilde{v} & \tilde{w} \end{bmatrix}.$$

This scheme requires tracking the diagonal signature matrix W' as the algorithm proceeds.

The next special case concerns columns with zero hyperbolic norm. Note that the hyperbolic Householder matrix U_x is not defined if $x^*Wx = 0$. Not only is there no hyperbolic Householder matrix that can map x to the zero vector, there can be no hyperbolic matrix at all since such a matrix would have to be singular and as we have already seen, hyperbolic matrices are nonsingular. It can be shown that the algorithm encounters a column with zero hyperbolic norm precisely when the underlying matrix A has a leading principal submatrix that is singular, and so this is avoided in the strictly definite cases, be they positive or negative definite.

We have been able to devise an algorithm that can skip over such singularities providing they have order 1, i.e., one principal leading submatrix is singular but the next one is not. This particular algorithm uses block hyperbolic Householder matrices of block size two. However, we have not been able to devise a method that works for any order singularity and so have decided not to include a discussion of this partial result. In fact,

[1] The sign of column x is defined as the sign (x^*Wx).

we view the development of such algorithms for matrices with arbitrary rank profile as a major outstanding question in this area. It should be pointed out that Levinson-type algorithms for inverse Cholesky factorization of Toeplitz matrices with arbitrary rank profile have been developed by Delsarte, Genin, and Kamp [16], and those results may contain clues about how the hyperbolic Householder approach might be extended to deal with general singular cases. Other work on this singular case can be found in [30].

We now summarize the algorithm:

Assumptions: A is an order n square Hermitian matrix with displacement rank α and nonsingular leading principal submatrices.

Input: $G_0' \in \mathbf{R}^{\alpha \times n}$, the matrix of α generators of A; and
$W = \text{diag}(\varepsilon_1, \cdots, \varepsilon_\alpha)$, the diagonal matrix describing the hyperbolic norm underlying the problem.

Output: $R \in \mathbf{R}^{n \times n}$, the upper triangular matrix in the factorization of A; and
$W' = \text{diag}(\pm 1, \cdots, \pm 1)$, the diagonal signature matrix for which $A = R^* W' R$.

HYPERBOLIC HOUSEHOLDER ALGORITHM
For $i = 1$ to n do
    begin
        $g = i$th column of G_0';
        $\sigma = g^* W g$;
        select e_k so that sign $(e_k^* W e_k) =$ sign (σ);
        define U_x so that $U_x g = -\sigma e_k$;
        form $G_0' = U_x G_0'$;
        set ith row of $R = k$th row of G_0';
        set $W'_{ii} = W_{kk}$;
        shift kth row of G_0' one to the right;
    end

It is important to observe that U_x should not and need not be stored explicitly. Because of its structure in terms of x, a matrix vector product of the form $U_x y$ can be computed in only $O(\alpha)$ operations. By comparison, explicit formation of U_x alone requires $O(\alpha^2)$ operations. Precise operation counts for real and complex cases are given in § 6.

5. Examples. The algorithm we describe is only useful, of course, if the generators of a matrix are explicitly and readily available. In this section, we briefly sketch two important examples in which the generators can be easily obtained.

5.1. Hermitian Toeplitz matrices. Suppose that A is a Hermitian Toeplitz matrix. It is well known that such matrices have displacement rank two in the general case. Moreover, it is well known that the generators are easily obtained from the matrix itself as follows.

Since we assume that the matrix has nonsingular leading submatrices, the diagonal entry is a nonzero real so we can set it to one by normalization. Let U be the strictly upper triangular part of A so that

$$A = I + U + U^* = (I+U)^*(I+U) - U^*U$$

$$= [I + U^* \quad U^*] \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} I+U \\ U \end{bmatrix}$$

where $U = (u_{i,j})$,

$$u_{ij} = \begin{cases} t_{j-i} & \text{if } i < j, \\ 0 & \text{if } i \geqq j. \end{cases}$$

This shows that the $2 \times n$ matrix of generators of A has the form

$$G_0' = \begin{bmatrix} 1 & t_1 & t_2 \cdots t_{n-1} \\ 0 & t_1 & t_2 \cdots t_{n-1} \end{bmatrix}$$

with signature

$$W = \operatorname{diag}(1, -1).$$

The well-known Schur algorithm is the standard technique for factoring A [2], [27], and the hyperbolic Householder construction reduces to Schur's algorithm if we make a few simple observations. First of all, Schur's algorithm can be viewed as applying a sequence of hyperbolic rotations to the matrix of generators. A hyperbolic rotation is defined as follows. To map the vector $[r\ s]^T$ to $[\sigma\ 0]^T$, let

$$t = -\frac{s}{r}$$

and define

$$V_t = \frac{1}{\sqrt{1 - |t|^2}} \begin{bmatrix} 1 & t \\ t & 1 \end{bmatrix}.$$

Then

$$V_t \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \sqrt{|r|^2 - |s|^2} \\ 0 \end{bmatrix}$$

as a direct expansion shows. In the positive-definite Hermitian Toeplitz case, the quantity t is guaranteed to be less than one in absolute value and is commonly known as a *Schur* or *reflection* coefficient. By the same token, defining the hyperbolic Householder matrix U_x based on

$$x = \begin{bmatrix} r \\ s \end{bmatrix} + \sigma \begin{bmatrix} \sqrt{|r|^2 - |s|^2} \\ 0 \end{bmatrix}$$

gives, after some arithmetic reduction $U_x = V_t$. Thus, functionally, these two methods reduce to the same computation in the simple Hermitian Toeplitz case. It should be noted of course that the hyperbolic Householder approach requires considerably more computation if the general form of the Householder matrix is used. This increased complexity is borne out by the numerical experiments performed in the next section.

5.2. Partial QR factorization of Toeplitz matrices. Suppose that T is an $m \times n$ rectangular Toeplitz matrix ($m \geqq n$). There have been a number of algorithms proposed for computing a QR factorization of T [3], [8], [10], [33]. Here we review how the basic notion of displacement rank plays a role in this problem. A much more general description of the relationship between displacement ranks of matrices and the displacement ranks of their products can be found in [8].

The displacement rank of the Toeplitz matrix T is generally two, as we have already seen. The general theory in [8] shows that the displacement rank of $A = T^*T$ is bounded by four. In fact, it is quite easy to explicitly derive the generators of A. Let $T = (t_{i-j})$ for $1 \leqq i \leqq m$ and $1 \leqq j \leqq n$. Then

$$\begin{aligned} a_{ij} &= \sum_{k=1}^{m} \bar{t}_{k-i} t_{k-j} \\ &= \sum_{k=1}^{m} \bar{t}_{k-i+1} t_{k-j+1} + \bar{t}_{1-i} t_{1-j} - \bar{t}_{m-i+1} t_{m-j+1} \\ &= a_{i-1,j-1} + \bar{t}_{1-i} t_{1-j} - \bar{t}_{m-i+1} t_{m-j+1} \end{aligned}$$

for $i, j > 2$. It follows that

$$A - Z^*AZ = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \bar{t}_{-1}t_{-1} - \bar{t}_{m-1}t_{m-1} & \cdots & \bar{t}_{-1}t_{1-n} - \bar{t}_{m-1}t_{m-n+1} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ a_{n1} & \bar{t}_{1-n}t_{-1} - \bar{t}_{m-n+1}t_{m-1} & \cdots & \bar{t}_{1-n}t_{1-n} - \bar{t}_{m-n+1}t_{m-n+1} \end{bmatrix}$$

$$= \alpha^*\alpha - \beta^*\beta + \gamma^*\gamma - \mu^*\mu$$

where

$$\alpha = \frac{1}{\sqrt{a_{11}}}[a_{11} \quad a_{12} \cdots a_{1n}],$$

$$\beta = \frac{1}{\sqrt{a_{11}}}[0 \quad a_{12} \cdots a_{1n}],$$

$$\gamma = [0 \quad t_{-1} \quad t_{-2} \cdots t_{1-n}],$$

$$\mu = [0 \quad t_{m-1} \quad t_{m-2} \cdots t_{m-n+1}].$$

Thus the generator for A is

$$G_0' = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \mu \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{a_{11}}}a_{11} & \frac{1}{\sqrt{a_{11}}}a_{12} & \cdot & \cdot & \frac{1}{\sqrt{a_{11}}}a_{1n} \\ 0 & \frac{1}{\sqrt{a_{11}}}a_{12} & \cdot & \cdot & \frac{1}{\sqrt{a_{11}}}a_{1n} \\ 0 & t_{-1} & t_{-2} & \cdot & t_{-n+1} \\ 0 & t_{m-1} & t_{m-2} & \cdot & t_{m-n+1} \end{bmatrix}$$

with signature matrix

$$W = \operatorname{diag}(1, -1, 1, -1).$$

Note that the generators of A in this case can be computed in $O(mn)$ complex operations. The reader is encouraged to review [6]–[9], [15], [25], and [27] for a general treatment of displacement rank and other situations where such structures arise.

6. Performance. In this section, we present the performance of the hyperbolic Householder algorithm presented in § 4 on the Alliant FX/80, Cray X-MP/48, and

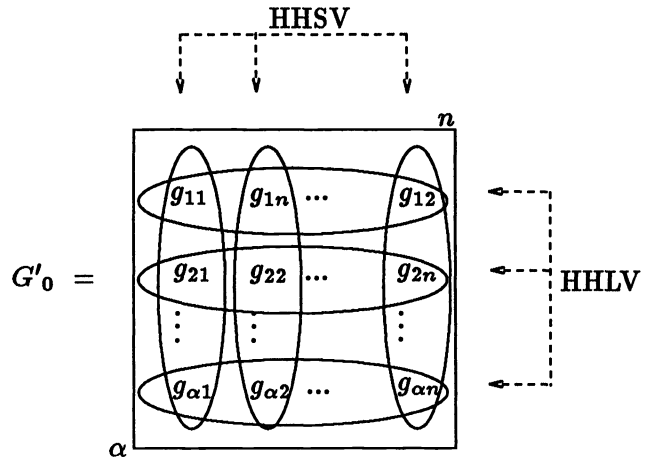


FIG. 1. *Vectorization and concurrency in* HHLV *and* HHSV *methods.*

Cray-2 computer systems.[2] For purposes of comparison, we also present results for the application of hyperbolic Givens rotations to the matrix of generators (Schur's method) as well as for the dense Cholesky factorization of the original matrix A in (2) as implemented by the ZPOFA routine in LINPACK [17]. It should be emphasized that all our experiments are for complex matrices.

In the implementation of hyperbolic Householder matrices to the matrix of generators G'_0 in (4) on vector and multivector processor machines, we can access/update either rows (of length n) or columns (of length α) serially (one CPU of the Cray X-MP/48 and Cray-2) or concurrently (Alliant FX/80). Figure 1 illustrates the two possible implementations that yield either long vector lengths n (HHLV) or short vector lengths α (HHSV), where n is the order of the original matrix A that has displacement rank α. We note that while HHLV can better exploit vectorization, HHSV will yield better parallelism in that n, rather than α vectors, can be processed simultaneously. A straightforward application of hyperbolic Given's rotations (HGIV) to the matrix of generators (Schur's method) will access pairs of rows of the matrix G'_0 in a sequential fashion and hence maintain the same vector lengths of HHLV. We refer the reader to [17] for a description of the Cholesky factorization of a dense matrix (in complex arithmetic).

The total cost in floating-point operations of the three methods we are considering (when implemented in real and complex arithmetic on the Alliant FX/80) is illustrated in Table 1, given the individual floating-point operation costs of basic complex arithmetic operations in Table 2. For our comparison of methods, we assume that all three methods are implemented in 64-bit *complex* arithmetic. We note that the cost of ZPOFA is independent of α since, in this case, we assume the original matrix A has been explicitly formed. As illustrated in Fig. 2, the cost of HGIV will eventually match and then supersede that of HHLV(SV) for a fixed matrix order n as α increases. Specifically, on the Alliant FX/80 the cost of HGIV and HHLV(SV) is identical for $\alpha = 7$ (8) for matrices of order 100 (500). Figure 3 reveals that while the cost differential between the methods remains fairly constant for very small displacement ranks ($\alpha = 2$) and increasing matrix orders

[2] Cray X-MP/48 and Cray-2 at the National Center for Supercomputer Applications (NCSA), University of Illinois at Urbana–Champaign.

TABLE 1
Cost of factorization methods in floating-point operations on the Alliant FX/80.

Method	Floating-point operations	
	Complex arithmetic	Real arithmetic
Hyperbolic Householder (HHLV, HHSV)	$12\alpha n^2 + 63\alpha n + 54n$	$3\alpha n^2 + 13\alpha n + 7n$
Hyperbolic Givens (HGIV)	$14(\alpha - 1)n^2 + 44(\alpha - 1)n$	$3(\alpha - 1)n^2 + 13(\alpha - 1)n$
Dense Cholesky (LINPACK)	$\frac{4}{3}n^3 + \frac{39}{6}n^2 - \frac{35}{6}n$ (ZPOFA)	$\frac{n^3}{3} + \frac{n^2}{2} - \frac{n}{6}$ (DPOFA)

n, the methods approximately require the same number of floating-point operations for larger displacement ranks ($\alpha = 8$).

In all the experiments presented in this section, we deliberately avoid singularity encounters in the factorization of a randomly generated G_0' by defining

- $g_{11} = \sum_{i=2}^{\alpha} g_{i1}$, and
- $\varepsilon_j = 1, j = 1, 2, \cdots, n$ (i.e., $W = I_n$).

Our concern here is to simply demonstrate the raw performance of the methods. The resolution of singularities is certainly an important concern for future work in the general application of hyperbolic Householder matrices.

The CPU times (milliseconds) required on eight processors of the Alliant FX/80 and one CPU of both the Cray X-MP/48 and Cray-2 for factoring matrices of order 100 (with displacement ranks ranging from $\alpha = 2$ to $\alpha = 10$) are given in Figs. 4 and 5. We note that the times indicated in all the figures presented in this section are nondedicated measurements. On the Alliant FX/80, although HGIV is far superior to either HHLV or HHSV for displacement ranks $\alpha \leqq 6$, the performance of HHSV for ranks $\alpha \geqq 7$ is the best. This can easily be attributed to the optimal parallelism (n rather than α) of HHSV mentioned above. For $\alpha = 10$, HHSV is 1.43 times faster than HGIV, 1.64 times faster than HHLV, and 3.07 times faster than the dense Cholesky factorization routine ZPOFA.

On the Cray X-MP/48 and Cray-2, the performance comparisons are quite different. As indicated in Figs. 5 and 6, the HHLV is far superior to HHSV for all displacement ranks. This is not surprising since the vector lengths (α) are extremely small for HHSV (64-element vector registers on Cray X-MP/48 and Cray-2). Note that the crossover point for HHLV relative to HGIV on both Cray machines occurs near $\alpha = 3$. The success

TABLE 2
Cost of basic complex arithmetic operations in floating-point operations on the Alliant FX/80, *Cray* X-MP/48, *and Cray*-2.

Complex operation	Floating-point operations	
	Alliant FX/80	Cray (X-MP/48) (−2)
$a + b$	2	2
$a*b$	6	6
a/b	10	13
$\lvert a \rvert$	13	13

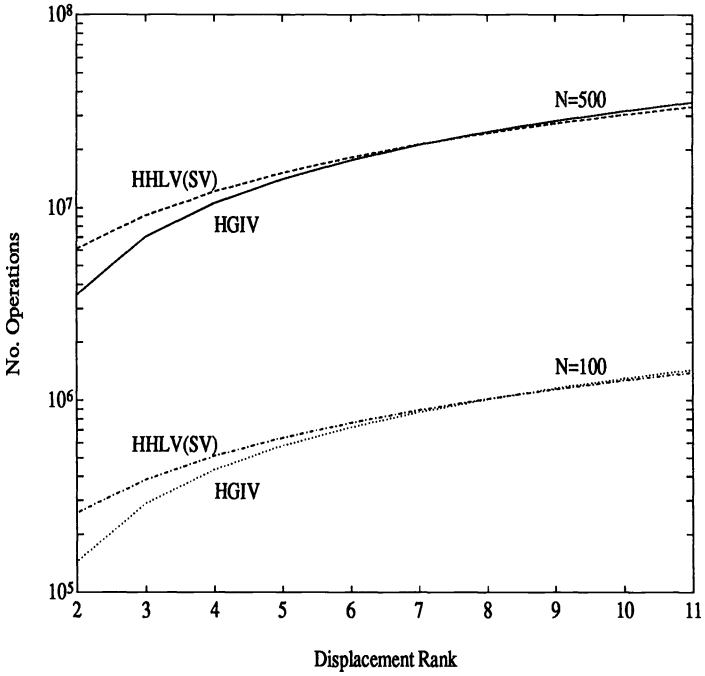


FIG. 2. *Floating-point operations of hyperbolic factorization methods for order* 100 *and* 500 *matrices on the Alliant* FX/80.

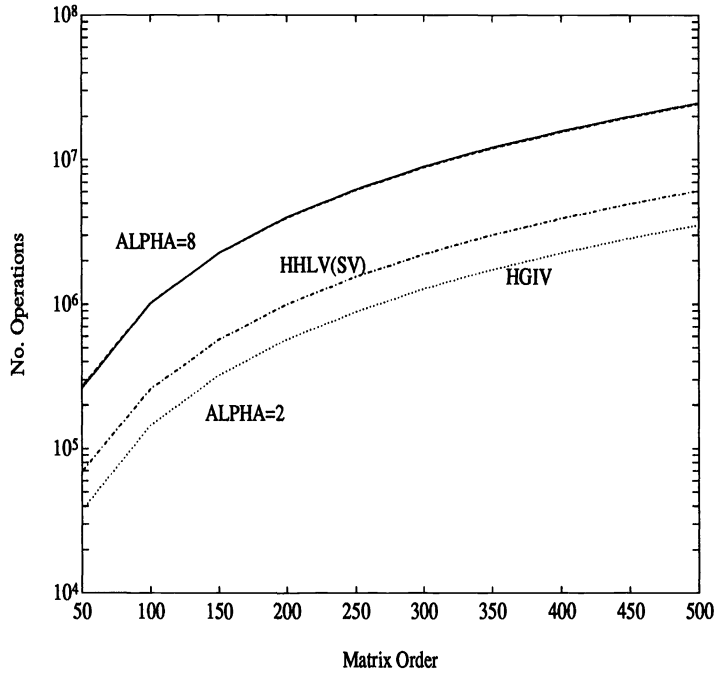


FIG. 3. *Floating-point operations of hyperbolic factorization methods for matrices of displacement rank* 2 *and* 8 *on the Alliant* FX/80 (HHLV(SV) *and* HGIV *are coincident for* ALPHA = 8).

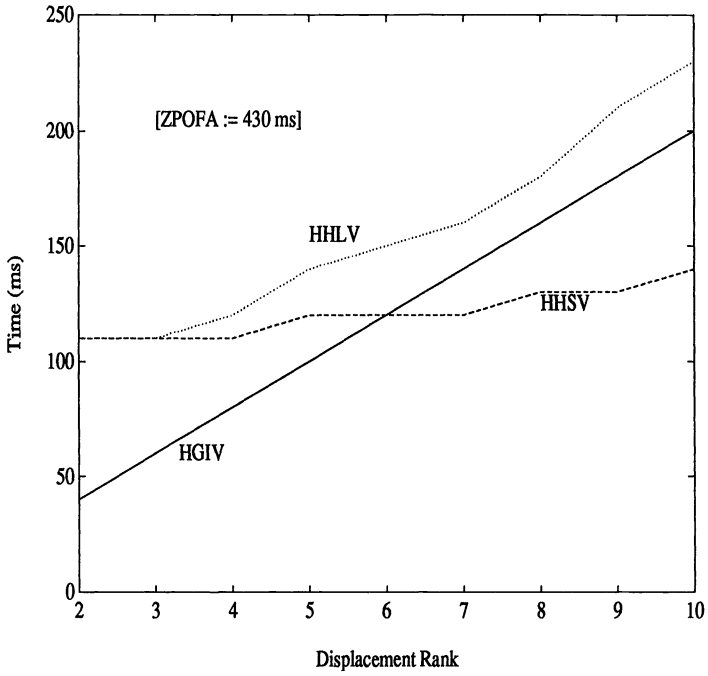


FIG. 4. *Performance of hyperbolic factorization methods for order* 100 *matrices on the Alliant* FX/80.

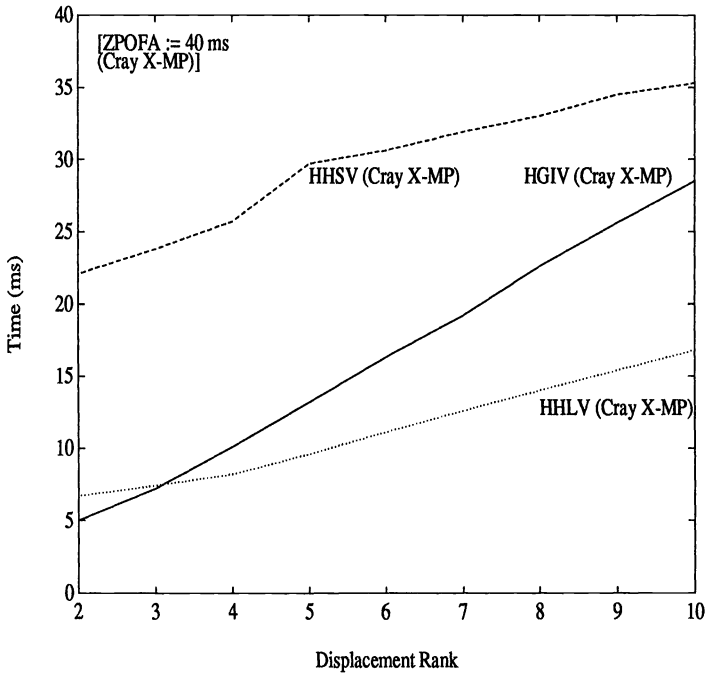


FIG. 5. *Performance of hyperbolic factorization methods for order* 100 *matrices on the Cray* X-MP/48 (1 CPU).

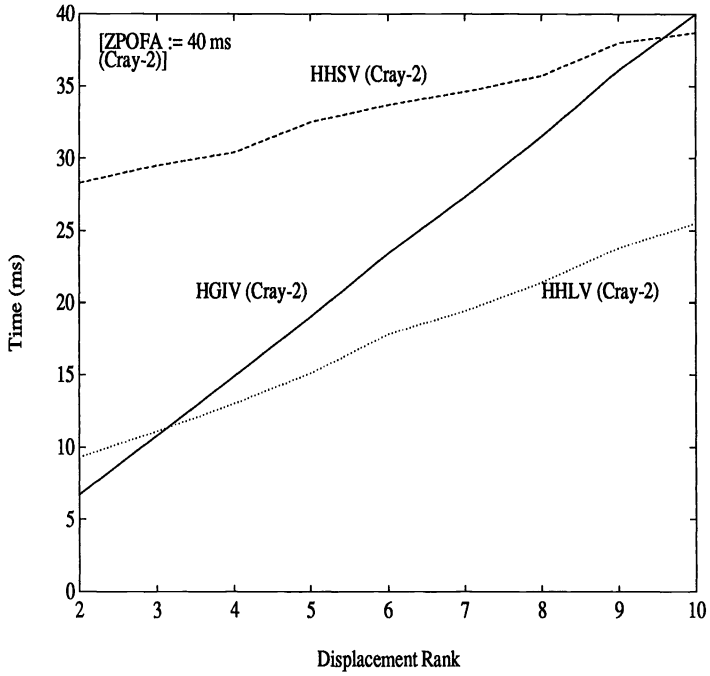


FIG. 6. *Performance of hyperbolic factorization methods for order* 100 *matrices on the Cray*-2 (1 CPU).

of HHLV on the Cray X-MP can be attributed in part to the *chaining* of longer vector operations (see [24]). For displacement rank $\alpha = 10$ on one CPU of the Cray X-MP/48, HHLV is 1.56 times faster than HGIV, 1.52 times faster than HHSV, and 1.56 times faster than ZPOFA. On one CPU of the Cray-2 (no chaining), HHLV is 1.56, 1.52, and 1.56 times faster than HGIV, HHSV, and ZPOFA, respectively. Figures 7 and 8 illustrate the behavior of the methods on the Cray X-MP/48 and Cray-2 for order 500 matrices. Whereas the crossover point in execution time for HHLV relative to HGIV is near 3 for $n = 100$ (see Figs. 5 and 6), the crossover point is either four or five for $n = 500$. Regarding speed improvements for factoring order 500 matrices on the Cray machines, HHLV can be as much as three and six times faster than HHSV and ZPOFA, respectively, on the Cray X-MP/48, and 1.6 and four times faster than HHSV and ZPOFA, respectively, on the Cray-2.

In order to assess machine performance rates of HHSV and HHLV for a fixed displacement rank $\alpha = 4$ in Figs. 9 and 10, respectively, we plot the megaflops (millions of floating-point operations using the operation counts in Table 2) achieved on the three machines for increasing matrix orders ranging from $n = 50$ to $n = 500$. In complex 64-bit arithmetic, we can achieve nearly 100 and 50 megaflops on the Cray X-MP/48 and Cray-2, respectively, when HHLV is used to factor $n = 500$ order matrices. On the Alliant FX/80, we can achieve roughly six megaflops for the more optimal HHSV method. If we consider the speedup of HHSV and HHLV for one to eight processors of the Alliant FX/80 (see Fig. 11), HHSV asymptotically reaches a speedup of six for matrices of displacement rank $\alpha = 4$, whereas the speedup of HHLV (which can only process four rows of the generator matrix G_0' concurrently) stabilizes around 2.5 for matrices of order $n = 50$ to $n = 500$.

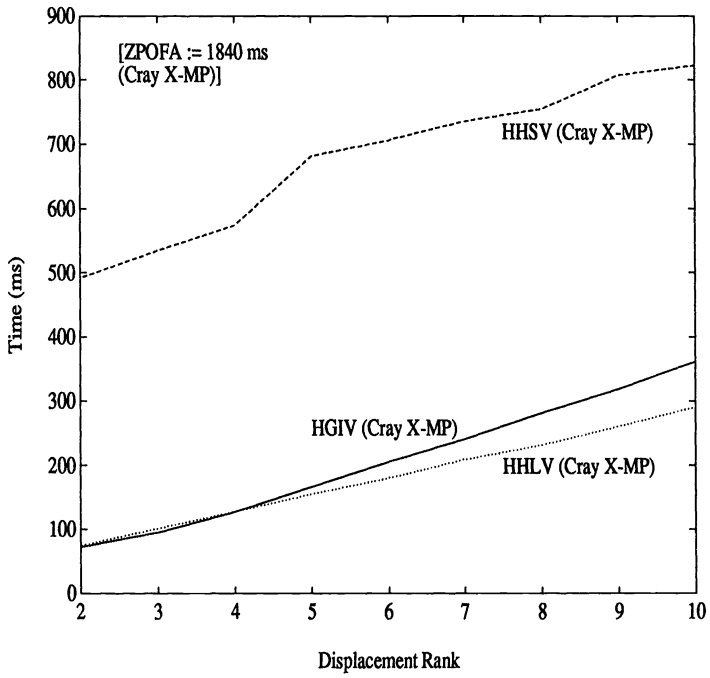


FIG. 7. *Performance of hyperbolic factorization methods for order* 500 *matrices on the Cray* X-MP/48 (1 CPU).

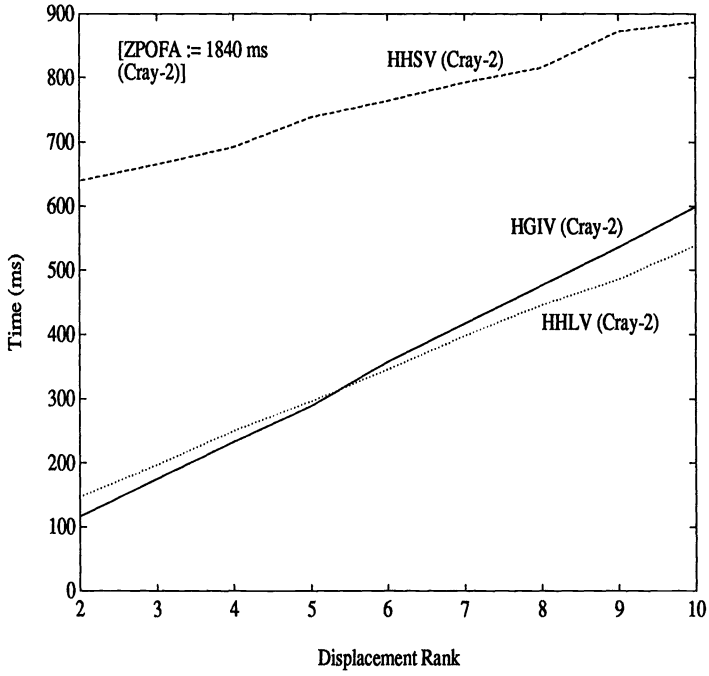


FIG. 8. *Performance of hyperbolic factorization methods for order* 500 *matrices on the Cray*-2 (1 CPU).

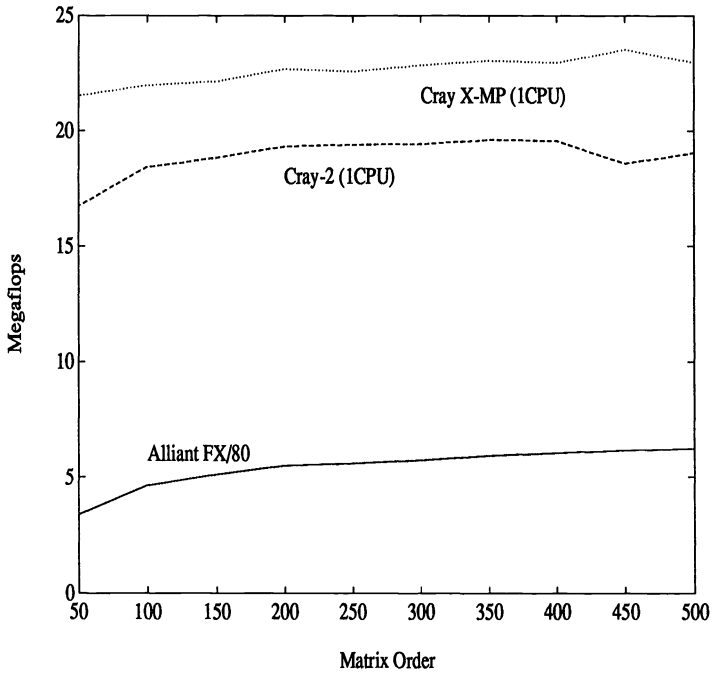


FIG. 9. *Performance of hyperbolic Householder method* HHSV *for matrices of displacement rank* 4.

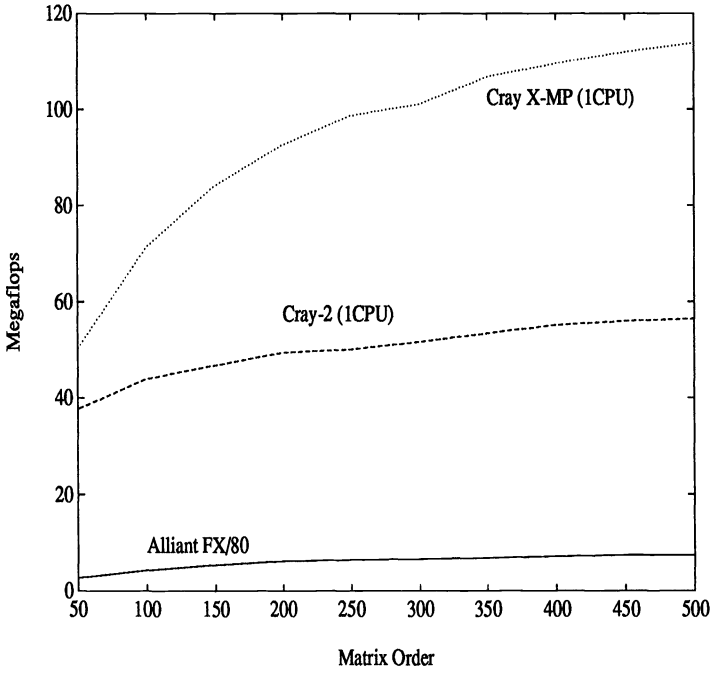


FIG. 10. *Performance of hyperbolic Householder method* HHLV *for matrices of displacement rank* 4.

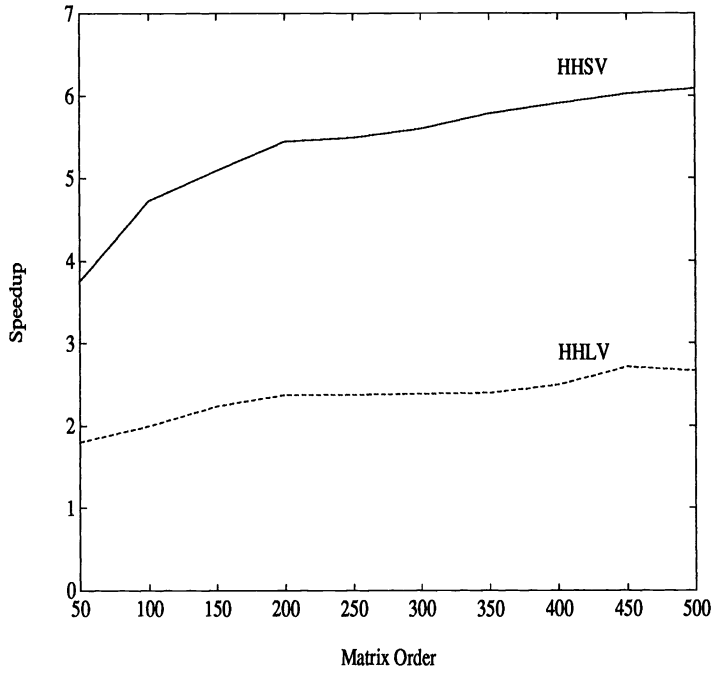


FIG. 11. *Speedup of hyperbolic Householder methods* HHLV *and* HHSV *from one to eight processors of the Alliant* FX/80.

7. Conclusions. An efficient algorithm for the factorization of matrices with small displacement rank has been described and implemented on Alliant FX/80, Cray X-MP/48, and Cray-2 vector computers. The algorithm is based on hyperbolic Householder transformations that were recently introduced by Rader and Steinhardt. For an order n matrix with displacement rank α, the algorithm uses $O(\alpha n^2)$ arithmetic operations and has good vectorized and parallel versions.

The algorithm presented here handles the same problems as Schur's algorithm for Hermitian Toeplitz matrices and the algorithms of Chun, Kailath, and Lev-Ari for Hermitian matrices with small displacement ranks. While there are some results about the numerical properties of hyperbolic rotations [1], [4], we are only aware of preliminary results about the numerical stability of algorithms based on hyperbolic Householder matrices for the downdating problem [5]. Our conjecture, as supported by numerical experiments, is that the algorithm presented here is stable when restricted to the class of positive definite matrices. Since the algorithm solves factorization problems for leading principal submatrices, as do the above mentioned algorithms, indefinite matrices can create numerical difficulties for this whole class of algorithms.

We believe that more research on the numerical stability of algorithms based on hyperbolic Householder matrices is needed. Moreover, algorithms that can handle singular leading submatrices would be valuable if they exist. These are important areas for future research.

Acknowledgments. The authors thank C. T. Pan for helpful discussions on related work, Kyle Gallivan for insights into various computer implementation issues, and Bob Plemmons for the helpful comments and suggestions.

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] G. S. AMMAR AND W. B. GRAGG, *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

[3] A. W. BOJANCZYK, R. P. BRENT, AND F. R. DE HOOG, QR *factorization of Toeplitz matrices*, Numer. Math., 49 (1986), pp. 81–94.

[4] A. W. BOJANCZYK, R. P. BRENT, P. VAN DOOREN, AND F. R. DE HOOG, *A note on downdating the Cholesky factorization*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 210–220.

[5] A. W. BOJANCZYK AND A. O. STEINHARDT, *Stabilized hyperbolic Householder transformations*, IEEE Trans. Acoust. Speech Signal Process., 37 (1989), pp. 1286–1288.

[6] J. CHUN AND T. KAILATH, *Generalized displacement structure for block-Toeplitz, Toeplitz-block, and Toeplitz-derived matrices*, Information Systems Laboratory, Stanford University, Stanford, CA, 1988.

[7] ———, *Divide-and-conquer solutions of least-squares problems for matrices with displacement structure*, Information Systems Laboratory, Stanford University, Stanford, CA, 1988.

[8] 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.

[9] J. CHUN AND T. KAILATH, *Displacement structure for Hankel- and Vandermonde-like matrices*, Information Systems Laboratory, Stanford University, Stanford, CA, 1988.

[10] G. CYBENKO, *Fast Toeplitz orthogonalization using inner products*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 734–740.

[11] F. DE HOOG, *A new algorithm for solving Toeplitz systems of equations*, Linear Algebra Appl., 88/89 (1987), pp. 122–138.

[12] J.-M. DELOSME, Y. GENIN, M. MORF, AND P. VAN DOOREN, Σ*-contractive embeddings and interpretation of some algorithms for recursive estimation*, in Proc. 14th IEEE Asilomar Conference on Circuits, Systems and Computers, New York, 1980, pp. 25–28.

[13] J.-M. DELOSME AND I. IPSEN, *Parallel solution of symmetric positive definite systems with hyperbolic rotations*, Linear Algebra Appl., 77 (1986), pp. 75–111.

[14] J.-M. DELOSME, S. C. EISENSTAT, AND J. R. MASSE, *Toeplitz solvers and vector processing*, in Proc. 11th GRETSI Symposium on Signal and Image Processing, Nice, France, June 1987, pp. 665–668.

[15] P. DELSARTE, Y. GENIN, AND Y. KAMP, *A polynomial approach to the generalized Levinson algorithm based on the Toeplitz distance*, IEEE Trans. Inform. Theory, 29 (1983), pp. 268–278.

[16] ———, *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.

[17] J. DONGARRA, J. BUNCH, C. MOLER, AND G. W. STEWART, LINPACK *Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[18] B. T. DRAINE, *The discrete dipole approximation and its application to interstellar graphite grains*, Astrophysics J., 1990, to appear.

[19] L. ELDÉN, *An efficient algorithm for the regularization of ill-conditioned least squares problems with triangular Toeplitz matrix*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 229–236.

[20] P. J. FLATAU, G. L. STEPHENS, AND B. T. DRAINE, *Light scattering in the discrete dipole approximation: exploiting the block-Toeplitz structure*, J. Opt. Soc. Amer. A, 1990, to appear.

[21] G. H. GOLUB, *Matrix decomposition and statistical computation*, in Statistical Computation, R. C. Milton and J. A. Nelder, eds., Academic Press, New York, 1969, pp. 365–397.

[22] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modeling matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.

[23] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[24] K. HWANG AND F. A. BRIGGS, *Computer Architecture and Parallel Processing*, McGraw-Hill, New York, 1984.

[25] T. KAILATH, S. KUNG, AND M. MORF, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407; Bull. Amer. Math. Soc., 1(1979), pp. 769–773.

[26] T. KAILATH, A. VIERA, AND M. MORF, *Orthogonal Transformation (Square-Root) Implementations of the Generalized Chandrasehkar and Generalized Levinson Algorithms*, Lecture Notes in Control and Information Sciences 14, Springer-Verlag, Berlin, 1978, pp. 81–91.

[27] H. LEV-ARI AND T. KAILATH, *Triangular factorization of structured Hermitian matrices*, in Schur Methods in Operator Theory and Signal Processing, Operator Theory: Advances and Applications, I, Vol. 18, I. Gohberg, ed., Birkhäuser-Verlag, Basel, 1986, pp. 301–324.

[28] B. R. MUSICUS, *Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices*, Report, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[29] V. PAN, *New effective methods for computations with structured matrices*, Tech. Report 88-28, Computer Science Department, State University of New York at Albany, Albany, NY, 1988.

[30] S. POMBRA, H. LEV-ARI, AND T. KAILATH, *Levinson and Schur algorithms for Toeplitz matrices with singular minors*, ICASSP 88, Digital Signal Processing, IEEE, New York, Vol. III, 1988, pp. 1643–1646.

[31] C. M. RADER AND A. O. STEINHARDT, *Hyperbolic Householder transformations*, IEEE Trans. Acoust. Speech Signal Process., 34 (1986), pp. 1589–1602.

[32] ———, *Hyperbolic Householder transforms*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 269–290.

[33] D. R. SWEET, *Fast Toeplitz orthogonalization*, Numer. Math., 43 (1984), pp. 1–21.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 521–530, October 1990

003

BOUNDING THE ERROR IN GAUSSIAN ELIMINATION FOR TRIDIAGONAL SYSTEMS*

NICHOLAS J. HIGHAM†

Abstract. If $\hat{x}$ is the computed solution to a tridiagonal system $Ax = b$ obtained by Gaussian elimination, what is the "best" bound available for the error $x - \hat{x}$ and how can it be computed efficiently? This question is answered using backward error analysis, perturbation theory, and properties of the LU factorization of A. For three practically important classes of tridiagonal matrix, those that are symmetric positive definite, totally nonnegative, or M-matrices, it is shown that $(A + E)\hat{x} = b$ where the backward error matrix E is small componentwise relative to A. For these classes of matrices the appropriate forward error bound involves Skeel's condition number cond (A, x), which, it is shown, can be computed exactly in $O(n)$ operations. For diagonally dominant tridiagonal A the same type of backward error result holds, and the author obtains a useful upper bound for cond (A, x) that can be computed in $O(n)$ operations. Error bounds and their computation for general tridiagonal matrices are discussed also.

Key words. tridiagonal matrix, forward error analysis, backward error analysis, condition number, comparison matrix, M-matrix, totally nonnegative, positive definite, diagonally dominant, LAPACK

AMS(MOS) subject classifications. primary 65F05, 65G05

C.R. classification. G.1.3

1. Introduction.

A natural question to ask when solving a general $n \times n$ linear system $Ax = b$ by Gaussian elimination with partial pivoting (GEPP) is, "how accurate is the computed solution, $\hat{x}$?" The traditional answer begins with Wilkinson's backward error result [22, p. 108]

$$(A+F)\hat{x}=b, \qquad \|F\|_\infty \leqq \rho_n p(n) u \|A\|_\infty, \tag{1.1}$$

where $p(n)$ is a cubic polynomial, u is the unit roundoff, and ρ_n is the *growth factor*, defined in terms of the quantities $a_{ij}^{(k)}$ generated during the elimination by

$$\rho_n = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

Applying standard perturbation theory to (1.1), one obtains the forward error bound

$$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty} \leqq \frac{\kappa_\infty(A)\rho_n p(n) u}{1-\kappa_\infty(A)\rho_n p(n) u} \qquad (\kappa_\infty(A)\rho_n p(n) u < 1), \tag{1.2}$$

where the condition number $\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$. Since the term $p(n)$ can usually be replaced by its square root for practical purposes [22, p. 108], or more crudely can be ignored, and since ρ_n is usually of order 1, this leads to the rule of thumb that $\hat{x}$ has about $-\log_{10} u - \log_{10} \kappa_\infty(A)$ correct decimal digits in its largest component.

In certain circumstances a bound potentially much smaller than (1.2) holds. This can be shown using the following componentwise backward error result, for general A [5]:

$$(A+E)\hat{x}=b, \qquad |E| \leqq c_n u |\hat{L}|\,|\hat{U}|, \tag{1.3}$$

* Received by the editors February 13, 1989; accepted for publication (in revised form) September 1, 1989.

† Department of Computer Science, Upson Hall, Cornell University, Ithaca, New York 14853 (na.nhigham@na-net.stanford.edu). Present address, Department of Mathematics, University of Manchester, Manchester, M13 9PL, United Kingdom.

where $c_n = 2n + O(u)$, and $\hat{L}$ and $\hat{U}$ are the computed LU factors of A (we assume, without loss of generality, that there are no row interchanges). Here, the absolute value operation $|\cdot|$ and the matrix inequality are interpreted componentwise. If $|\hat{L}||\hat{U}| \leqq c_n'|A|$, then (1.3) may be written

$$(A+E)\hat{x} = b, \qquad |E| \leqq c_n'' u |A|, \tag{1.4}$$

which represents the "ideal" situation where E is small componentwise relative to A. Note, in particular, that $e_{ij} = 0$ if $a_{ij} = 0$. The bound in (1.4) holds, at least, when A is triangular (see, e.g., [17]), and when A is *totally nonnegative* [5], assuming no pivoting in both cases. (A is totally nonnegative if all its minors of any order are nonnegative.) The bound also holds, under certain assumptions, if $\hat{x}$ is the result of GEPP followed by one step of iterative refinement in single precision [1], [20].

Perturbation results appropriate to (1.4) render the bound [19]

$$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty} \leqq \frac{\operatorname{cond}(A,x)c_n'' u}{1-\operatorname{cond}(A)c_n'' u} \qquad (\operatorname{cond}(A)c_n'' u < 1), \tag{1.5}$$

where

$$\operatorname{cond}(A,x) = \frac{\| |A^{-1}||A||x| \|_\infty}{\|x\|_\infty}$$

and

$$\operatorname{cond}(A) = \operatorname{cond}(A,e), \qquad e = (1, 1, \cdots, 1)^T.$$

The key difference between (1.5) and (1.2) is in the condition number terms: $\operatorname{cond}(A, x)$ is no larger than $\kappa_\infty(A)$ and is often much smaller. In particular, $\operatorname{cond}(A, x)$ is invariant under row scaling of A, whereas $\kappa_\infty(A)$ is not.

This work focuses on the case where A is tridiagonal, and was partly motivated by the question of what types of error bounds and condition number estimates should be provided in the LAPACK routines for solving tridiagonal systems [3], [9]. (LAPACK is to be a collection of Fortran 77 routines for solving linear equations, linear least squares problems, and matrix eigenvalue problems [6].) The aim of the work is to determine classes of tridiagonal systems for which the bounds (1.4) and (1.5) are valid and to develop efficient methods for estimating or computing the condition numbers in (1.5) and (1.2).

In § 2 we present a specialized version of the backward error bound (1.3) for tridiagonal matrices. The result is known, but we give a short proof since the precise value of the bound is important, and we were unable to find a suitable reference.

In § 3 we show that (1.4) holds for Gaussian elimination without pivoting if the tridiagonal matrix A is symmetric positive definite, totally nonnegative, or an M-matrix. (Thus, for these types of matrices there is no advantage in doing iterative refinement in single precision.) We show that in each case $\operatorname{cond}(A, x)$, and hence also the bound in (1.5), can be computed exactly in $O(n)$ operations. Diagonally dominant matrices also enjoy a relatively small componentwise backward error, and, as we show in § 4, a good upper bound for $\operatorname{cond}(A, x)$ can be obtained in $O(n)$ operations.

We consider general tridiagonal matrices in § 5; we explain which error bounds are applicable and how the corresponding condition numbers may be estimated. In § 6 some further comments are made concerning practical use of the bounds and condition numbers, and some numerical results are presented to illustrate the value of using a componentwise backward error approach when possible.

2. Gaussian elimination and its error analysis. Consider the real $n \times n$, nonsingular tridiagonal matrix

$$A = \begin{bmatrix} d_1 & e_1 & & & \\ c_2 & d_2 & e_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & e_{n-1} \\ & & & c_n & d_n \end{bmatrix}, \tag{2.1}$$

and assume A has an LU factorization $A = LU$, where

$$L = \begin{bmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} u_1 & e_1 & & & \\ & u_2 & e_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & e_{n-1} \\ & & & & u_n \end{bmatrix}. \tag{2.2}$$

Gaussian elimination for computing L and U is described by the recurrence relations

$$u_1 = d_1; \qquad \left.\begin{aligned} l_i &= c_i/u_{i-1} \\ u_i &= d_i - l_i e_{i-1} \end{aligned}\right\} i = 2, \cdots, n. \tag{2.3}$$

To investigate the effects of rounding error, we will employ the model

$$fl(x \text{ op } y) = (x \text{ op } y)(1+\delta), \qquad |\delta| \leqq u, \tag{2.4a}$$

$$fl(x \text{ op } y) = (x \text{ op } y)/(1+\varepsilon), \qquad |\varepsilon| \leqq u, \tag{2.4b}$$

where u is the unit roundoff and $\text{op} \in \{+, -, *, /\}$. Note that (2.4b) is valid under the same assumptions as (2.4a), although usually only (2.4a) is used in a rounding error analysis. Judicious use of (2.4b) simplifies the analysis slightly.

Applying (2.4) to the relations (2.3) and using a hat to denote computed quantities, we have

$$(1+\varepsilon_i)\hat{l}_i = \frac{c_i}{\hat{u}_{i-1}}, \qquad |\varepsilon_i| \leqq u,$$

$$(1+\theta_i)\hat{u}_i = d_i - \hat{l}_i e_{i-1}(1+\delta_i), \qquad |\theta_i|, |\delta_i| \leqq u.$$

Hence

$$|c_i - \hat{l}_i \hat{u}_{i-1}| \leqq u|\hat{l}_i \hat{u}_{i-1}|,$$

$$|d_i - \hat{l}_i e_{i-1} - \hat{u}_i| \leqq u(|\hat{l}_i e_{i-1}| + |\hat{u}_i|).$$

In matrix terms these bounds may be written as

$$A = \hat{L}\hat{U} + E, \qquad |E| \leqq u|\hat{L}||\hat{U}|. \tag{2.5}$$

If the LU factorization is used to solve a system $Ax = b$ by forward and back substitution, then it is straightforward to show that the computed solution $\hat{x}$ satisfies

$$(\hat{L} + \Delta L)(\hat{U} + \Delta U)\hat{x} = b, \qquad |\Delta L| \leqq u|\hat{L}|, \qquad |\Delta U| \leqq (2u + u^2)|\hat{U}|. \tag{2.6}$$

Combining (2.5) and (2.6) we have, overall,

$$(A+F)\hat{x} = b, \qquad |F| \leqq f(u)|\hat{L}||\hat{U}|, \qquad f(u) = 4u + 3u^2 + u^3. \tag{2.7}$$

We have avoided using $O(u^2)$ notation in order to emphasize that there are no large constants in the higher-order terms; in particular, $f(u)$ is independent of n.

3. Componentwise backward error and computation of cond (A, x). The backward error result (2.7) applies to arbitrary nonsingular tridiagonal A having an LU factorization. We are interested in determining classes of tridiagonal A for which the bound $|F| \leqq f(u)|\hat{L}||\hat{U}|$ implies the "ideal bound"

$$|F| \leqq g(u)|A|. \tag{3.1}$$

Certainly, (3.1) holds if

$$|\hat{L}||\hat{U}| = |\hat{L}\hat{U}|, \tag{3.2}$$

for then, using (2.5),

$$|\hat{L}||\hat{U}| = |A-E| \leqq |A| + u|\hat{L}||\hat{U}|,$$

so that

$$|\hat{L}||\hat{U}| \leqq \frac{1}{1-u}|A|. \tag{3.3}$$

Three classes of matrices for which (3.2) holds for the exact L and U are identified in the following theorem. A nonsingular $A \in \mathbf{R}^{n\times n}$ is an M-matrix if $a_{ij} \leqq 0$ for all $i \neq j$ and $A^{-1} \geqq 0$. There are many equivalent conditions for A to be an M-matrix [2, Chap. 6]; for example, the condition $A^{-1} \geqq 0$ can be replaced by the condition that all the principal minors of A are positive.

THEOREM 3.1. *Let $A \in \mathbf{R}^{n\times n}$ be nonsingular and tridiagonal. If any of the following conditions hold then A has an LU factorization and $|L||U| = |LU|$:*

(a) *A is symmetric positive definite;*

(b) *A is totally nonnegative, or equivalently, $L \geqq 0$ and $U \geqq 0$;*

(c) *A is an M-matrix, or equivalently, L and U have positive diagonal elements and nonpositive off-diagonal elements;*

(d) *A is sign equivalent to a matrix B of type* (a)–(c); *that is, $A = D_1BD_2$, where $|D_1| = |D_2| = I$.*

Proof. For (a), it is well known that a symmetric positive definite A has an LU factorization in which $U = DL^T$, where D is diagonal with positive diagonal elements. Hence $|L||U| = |L||D||L^T| = |LDL^T| = |LU|$. In (b) and (c) the equivalences, and the existence of an LU factorization, follow from known results on totally nonnegative matrices [4] and M-matrices [2]; $|L||U| = |LU|$ is immediate from the sign properties of L and U. (d) is trivial. □

THEOREM 3.2. *If the tridiagonal matrix A is of type* (a)–(d) *in Theorem* 3.1, *and if the unit roundoff u is sufficiently small, then Gaussian elimination for solving $Ax = b$ succeeds and the computed solution $\hat{x}$ satisfies*

$$(A+F)\hat{x} = b, \qquad |F| \leqq h(u)|A|, \qquad h(u) = \frac{4u+3u^2+u^3}{1-u}. \tag{3.4}$$

Proof. If u is sufficiently small, then for types (a)–(c) the diagonal elements of $\hat{U}$ will be positive, since $\hat{u}_i \to u_i > 0$ as $u \to 0$. It is easy to see that $\hat{u}_i > 0$ for all i ensures that $|\hat{L}||\hat{U}| = |\hat{L}\hat{U}|$. The argument is similar for type (d). The result therefore follows from (2.7) and (3.3). □

Theorem 3.2 appears to be new in the case of M-matrices. A result of the form (3.4) (with a c_n term in the bound) is valid for any totally nonnegative matrix [5]. The symmetric positive definite case in Theorem 3.2 is also known [8].

A corollary of Theorem 3.2 is that it is not necessary to pivot for the matrices specified in the theorem (and, indeed, pivoting could vitiate the bound (3.4)). Note that large multipliers may occur under the conditions of Theorem 3.2, but they do not affect the stability. (Recall the well-known property [21, p. 412] that arbitrarily large multipliers may occur in LU factorization of a general symmetric positive definite matrix, yet the growth factor $\rho_n \leqq 1$.) We stress this point because in [13], which deals with Gaussian elimination of tridiagonal Toeplitz matrices, it is stated that "the stability of the elimination process is controlled by the size of the multipliers m_j." We also mention that the example given by Harrod [14] of the M-matrix

$$A = \begin{bmatrix} 2 & -2 & 0 \\ \varepsilon - 2 & 2 & 0 \\ 0 & -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ (\varepsilon-2)/2 & 1 & 0 \\ 0 & -1/\varepsilon & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & 3 \end{bmatrix} = LU,$$

for which the multiplier l_{32} is unbounded as $\varepsilon \to 0$, is an example where Gaussian elimination performs very stably, as Theorem 3.2 shows.

We now turn our attention to computing cond (A, x). We show that if $|L||U| = |LU|$ then cond (A, x) can be computed in $O(n)$ operations.

THEOREM 3.3. *If the nonsingular tridiagonal matrix $A \in \mathbf{R}^{n\times n}$ has the LU factorization $A = LU$ and $|L||U| = |A|$, then $|U^{-1}||L^{-1}| = |A^{-1}|$.*

Proof. Using the notation of (2.1) and (2.2), $|L||U| = |A| = |LU|$ if and only if for all i

$$|l_i e_{i-1} + u_i| = |l_i||e_{i-1}| + |u_i|,$$

that is, if

$$\operatorname{sign}\left(\frac{l_i e_{i-1}}{u_i}\right) = 1. \tag{3.5}$$

Using the formulae

$$(U^{-1})_{ij} = \frac{1}{u_j} \prod_{p=i}^{j-1} \left(\frac{-e_p}{u_p}\right) \qquad (j \geqq i), \tag{3.6}$$

$$(L^{-1})_{ij} = \prod_{p=j}^{i-1} (-l_{p+1}) \qquad (i \geqq j), \tag{3.7}$$

we have

$$\begin{aligned}
(U^{-1}L^{-1})_{ij} &= \sum_{k=\max(i,j)}^{n} (U^{-1})_{ik}(L^{-1})_{kj} \\
&= \sum_{k=\max(i,j)}^{n} \frac{1}{u_k} \prod_{p=i}^{k-1} \left(\frac{-e_p}{u_p}\right) \prod_{p=j}^{k-1} (-l_{p+1}) \\
&= \prod_{p=i}^{\max(i,j)-1} \left(\frac{-e_p}{u_p}\right) \cdot \prod_{p=j}^{\max(i,j)-1} (-l_{p+1}) \cdot \sum_{k=\max(i,j)}^{n} \frac{1}{u_k} \prod_{p=\max(i,j)}^{k-1} \left(\frac{e_p l_{p+1}}{u_p}\right) \\
&= \prod_{p=i}^{\max(i,j)-1} \left(\frac{-e_p}{u_p}\right) \cdot \prod_{p=j}^{\max(i,j)-1} (-l_{p+1}) \cdot \frac{1}{u_{\max(i,j)}} \cdot \sum_{k=\max(i,j)}^{n} \prod_{p=\max(i,j)}^{k-1} \left(\frac{e_p l_{p+1}}{u_{p+1}}\right).
\end{aligned}$$

Thus, in view of (3.5), it is clear that $|U^{-1}L^{-1}|_{ij} = (|U^{-1}||L^{-1}|)_{ij}$, as required. □

To see the significance of the property $|U^{-1}||L^{-1}| = |A^{-1}|$, note first that, as is clear from (3.6) and (3.7),

$$|U^{-1}| = M(U)^{-1}, \qquad |L^{-1}| = M(L)^{-1},$$

where for $B \in \mathbf{R}^{n\times n}$ the comparison matrix $M(B)$ is defined by

$$(M(B))_{ij} = \begin{cases} |b_{ii}|, & i = j, \\ -|b_{ij}|, & i \neq j. \end{cases}$$

Thus, if $|A^{-1}| = |U^{-1}||L^{-1}|$ and $y \geqq 0$ then

$$|A^{-1}|y = |U^{-1}||L^{-1}|y = M(U)^{-1}M(L)^{-1}y.$$

By taking $y = |A||x|$ it follows that cond (A, x) can be computed in $O(n)$ operations:

$$\begin{aligned} &\text{form } y = |A||x|, \\ &\text{solve } M(L)v = y, \\ &\text{solve } M(U)w = v, \\ &\text{compute } \|w\|_\infty / \|x\|_\infty. \end{aligned} \tag{3.8}$$

For the special case $y = e$ and A symmetric positive definite, (3.8) was used in [15, § 6] to compute $\|A^{-1}\|_\infty$ in $O(n)$ operations.

Of course, in practice we use the computed $\hat{L}$ and $\hat{U}$ in place of the exact LU factors. If cond (A) is not too large (cond $(A)u < \frac{1}{2}$, say), then we are guaranteed a satisfactory computed value of cond (A, x), that is, one having some correct digits.

4. Diagonally dominant matrices. A in (2.1) is diagonally dominant by rows if

$$|d_i| \geqq |c_i| + |e_i| \quad \text{for all } i \quad (c_1 \equiv e_n \equiv 0),$$

and diagonally dominant by columns if A^T is diagonally dominant by rows. Such A have an LU factorization, but $|L||U| \neq |A|$ in general, and so we cannot apply the results of the last section. However, as the next result shows, $|L||U|$ can be bounded by a small multiple of $|A|$. Combining this result with (2.7), we are able to conclude that the componentwise backward error is small in solving a diagonally dominant tridiagonal system $Ax = b$.

THEOREM 4.1. *Suppose $A \in \mathbf{R}^{n\times n}$ is nonsingular, tridiagonal, and diagonally dominant by rows or columns, and let A have the LU factorization $A = LU$. Then $|L||U| \leqq 3|A|$.*

Proof. If $|i - j| = 1$ then $(|L||U|)_{ij} = |a_{ij}|$, so it suffices to consider the diagonal elements and show that (using the notation of (2.2))

$$|l_i e_{i-1}| + |u_i| \leqq 3|d_i|.$$

The rest of the proof is for the case where A is diagonally dominant by rows; the proof for diagonal dominance by columns is similar.

First, we claim that $|e_i| \leqq |u_i|$ for all i. The proof is by induction. For $i = 1$ the result is immediate, and if it is true for $i - 1$ then from (2.3)

$$\begin{aligned} |u_i| &\geqq |d_i| - |l_i||e_{i-1}| = |d_i| - \frac{|c_i|}{|u_{i-1}|}|e_{i-1}| \\ &\geqq |d_i| - |c_i| \geqq |e_i|, \end{aligned}$$

as required. Note that, similarly, $|u_i| \leqq |d_i| + |c_i|$. Finally,

$$|l_i e_{i-1}| + |u_i| = \left|\frac{c_i}{u_{i-1}} e_{i-1}\right| + |u_i| \leqq |c_i| + |u_i|$$
$$\leqq |c_i| + (|d_i| + |c_i|)$$
$$\leqq 3|d_i|. \qquad \square$$

Unfortunately, it is not generally true for diagonally dominant A that $|A^{-1}| = |U^{-1}||L^{-1}|$, so we cannot compute cond (A, x) using the $O(n)$ operations technique of the last section. However we can compute the upper bound in

$$|A^{-1}|y \leqq |U^{-1}||L^{-1}|y \qquad (y = |A||x|)$$

in $O(n)$ operations. Concentrating, for the moment, on diagonal dominance by rows, a bound for how much of an overestimate this upper bound can be is provided by the following result.

THEOREM 4.2. *Suppose the nonsingular, row diagonally dominant, tridiagonal matrix $A \in \mathbf{R}^{n\times n}$ has the LU factorization $A = LU$. Then, if $y \geqq 0$,*

$$\||U^{-1}||L^{-1}|y\|_\infty \leqq (2n-1)\||A^{-1}|y\|_\infty.$$

Proof. We have $L^{-1} = UA^{-1}$, so

$$|U^{-1}||L^{-1}|y \leqq |U^{-1}||U||A^{-1}|y = |V^{-1}||V||A^{-1}|y,$$

where the bidiagonal matrix $V = \operatorname{diag}(u_{ii})^{-1}U$ has $v_{ii} \equiv 1$ and $|v_{i,i+1}| = |e_i/u_i| \leqq 1$ (see the proof of Theorem 4.1). Thus

$$|U^{-1}||L^{-1}|y \leqq \begin{bmatrix} 1 & 1 & \cdots & 1 \\ & 1 & \cdots & 1 \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & & \\ & 1 & \ddots & \\ & & \ddots & 1 \\ & & & 1 \end{bmatrix}|A^{-1}|y,$$

and the result follows on taking norms. $\square$

Theorem 4.2 says that when A is row diagonally dominant our upper bound for cond (A, x) is too big by a factor at most $2n - 1$. This is somewhat unsatisfactory since n can be very large. For $n = 2$ the bound in Theorem 4.2 is attained as $\alpha \to \infty$ in the example

$$A = \begin{bmatrix} 1 & 1 \\ -\alpha & \alpha+1 \end{bmatrix}, \qquad y = e.$$

For general n we have not been able to construct any examples in which the bound in Theorem 4.2 is attained (except by relaxing the row diagonal dominance assumption). In a wide variety of numerical tests with both random and nonrandom matrices, the upper bound has never exceeded the quantity it bounds by more than a small constant factor (3, say). Moreover, the bound is exact if the row diagonally dominant A happens to be symmetric (so that it is positive definite), nonnegative (that is, $A \geqq 0$, which implies it is totally nonnegative), or an M-matrix—all three cases are common in applications. We therefore regard the upper bound as reliable in practice, and conjecture that the factor $2n - 1$ in Theorem 4.2 can be improved to a constant independent of n.

We mention that Neumaier [18] found that $\||U^{-1}||L^{-1}|y\|_\infty \leqq 2\||A^{-1}|y\|_\infty$ held in a small number of tests with *full* random row diagonally dominant matrices and random $y > 0$, and this inequality is confirmed by our own tests with random matrices.

However, no theoretical bound on the overestimation factor is known in the case of full A.

A weaker analogue of Theorem 4.2 holds when A is diagonally dominant by columns. The inequality $|U^{-1}||L^{-1}|y \leqq |A^{-1}||L||L^{-1}|y$ leads to, for $y > 0$,

$$\||U^{-1}||L^{-1}|y\|_\infty \leqq (2n-1)\theta\||A^{-1}|y\|_\infty, \qquad \theta = \frac{\max_i |y_i|}{\min_i |y_i|}.$$

Despite the unbounded θ term in this inequality, we have not observed or constructed any examples where the upper bound is more than a small constant factor too big. Thus we regard the upper bound as being of practical use also when A is diagonally dominant by columns.

5. General tridiagonal matrices. We turn now to tridiagonal systems $Ax = b$ where A does not fall into any of the classes considered in the previous two sections. Suppose GEPP is used to solve the system. Suppose also that we wish to refer to backward and forward error bounds of the forms (1.1) and (1.2) and to estimate or compute $\kappa_\infty(A)$. Several algorithms for computing $\kappa_\infty(A)$ exactly in $O(n)$ operations are presented in [15]. As explained in [15], these algorithms (except the algorithm for symmetric positive definite A) have the property that the intermediate numbers can have a large dynamic range (the more so, the more diagonally dominant A is), and the algorithms can break down in floating-point arithmetic due to underflow or overflow. These numerical problems can be overcome, but at a nontrivial increase in cost (see [15]). Our preferred approach is to use the matrix norm estimator SONEST from [16]. This provides an *estimate* for $\|B\|_1$ (a lower bound) at the cost of computing a few matrix-vector products Bc and B^Td. Typically four or five products are required; the norm estimate is frequently exact and is almost always correct to within a factor 3. In our application, $B = A^{-T}$, and so we need to solve a few linear systems $A^Ty = c$ and $Az = d$, which can be done using the LU factorization already computed. The SONEST approach has about the same computational cost as the methods in [15].

Next, suppose that GEPP followed by iterative refinement is used to solve the tridiagonal system $Ax = b$. Then, under suitable assumptions, a result of the form (3.4) holds [1], [20], and so the appropriate condition number is $\operatorname{cond}(A, x)$. (See [1] for a discussion of possible violation of the assumptions when x and b are sparse, and for suggested cures.) The techniques of [15] could be adapted to compute $\operatorname{cond}(A, x)$ in $O(n)$ operations, with the same practical numerical difficulties described above. However, as shown in [1], [7], SONEST can be used to estimate $\operatorname{cond}(A, x)$ (even for general A), and this is the approach we recommend.

Finally, note that for GEPP one could use the elementwise backward error result (2.7) (suitably modified to take account of pivoting), for which a forward error bound involving the condition number $\||A^{-1}||\hat{L}||\hat{U}||x|\|_\infty/\|x\|_\infty$ can be derived. Again, this condition number (which is row scaling independent) can be estimated using SONEST.

6. Practical considerations. We discuss several practical issues concerning the condition numbers and algorithms described above.

- For symmetric positive definite A the standard way to solve $Ax = b$ is by using a Cholesky or LDL^T factorization, rather than an LU factorization. The LINPACK routine SPTSL uses a nonstandard "LUB" factorization resulting from the BABE ("burn at both ends") algorithm (see [10], [15]). The results of § 3 are applicable to all of these factorizations, with minor modifications. Note that the LDL^T factorization requires n fewer divisions in the substitution stage than the Cholesky factorization.
- A drawback to the computation or estimation of $\operatorname{cond}(A, x) = \||A^{-1}||A||x|\|_\infty/\|x\|_\infty$ is the need to keep a copy of A in order to form the product $|A||x|$ once x has

been computed. If n is large it may not be possible to store a copy of A. One can circumvent this problem for the matrices in Theorems 3.1 and 4.1, for which $|A| = |L||U|$ and $|A| \leq |L||U| \leq 3|A|$, respectively, by using $|L||U||x|$ in place of $|A||x|$.

- We give the computational costs of the error estimation techniques in two particular cases, in terms of flops [12, p. 32]. For a general tridiagonal $A \in \mathbf{R}^{n\times n}$, factoring $PA = LU$ by GEPP and solving $Ax = b$ by substitution costs $(5 + 2p)n$ flops, where $p \in [0, 1]$ depends on the number of interchanges; estimating $\kappa_\infty(A)$ requires $2n$ flops to compute $\|A\|_\infty$ and, typically, $4(3 + p)n$ or $5(3 + p)n$ flops to estimate $\|A^{-1}\|_\infty$ using SONEST. For a symmetric positive definite $A \in \mathbf{R}^{n\times n}$, factoring $A = LDL^T$ and solving $Ax = b$ requires $5n$ flops, and computing cond (A, x) requires $6n$ flops. Thus these error estimation techniques at least double the cost of solving a linear system.
- Instead of computing cond (A, x) one could compute cond (A) = cond $(A, e) \geq$ cond (A, x). The same cond (A) value could be reused when solving systems with the same A but different right-hand sides. However, this approach reduces the sharpness of the bounds, since cond (A)/cond (A, x) can be arbitrarily large.

Finally, we present a numerical experiment that gives an indication of the sharpness of the various error bounds. We used a tridiagonal matrix given by Dorr [11] that occurs in the solution of a singular perturbation problem by finite differences. With $m = \lfloor (n + 1)/2 \rfloor$, $h = 1/(n + 1)$, and $\varepsilon > 0$, the matrix is defined by (see (2.1))

$$c_i = \begin{cases} -\varepsilon/h^2, & 1 \leq i \leq m, \\ -\varepsilon/h^2 + (\tfrac{1}{2} - ih)/h^2, & m+1 \leq i \leq n, \end{cases}$$

$$e_i = \begin{cases} -\varepsilon/h^2 - (\tfrac{1}{2} - ih)/h^2, & 1 \leq i \leq m, \\ -\varepsilon/h^2, & m+1 \leq i \leq n, \end{cases}$$

and $d_i = -(c_i + e_i)$, $1 \leq i \leq n$ (note that c_1 and e_n are introduced solely to define d_1 and d_n). A is a nonsingular, row diagonally dominant M-matrix. For small values of the parameter ε the matrix is ill-conditioned.

We chose $n = 50$ and $\varepsilon = 0.009$. We solved $Ax = b$ for six different right-hand sides. The computations were done in PC-MATLAB, with simulated single precision arithmetic of unit roundoff $u = 2^{-23} \approx 1.2 \times 10^{-7}$. For each system we computed $\hat{x}$ in single precision and x and the relative error $\|x - \hat{x}\|_\infty / \|x\|_\infty$ in double precision. Since A is an M-matrix, cond (A, x), cond (A), and $\kappa_\infty(A)$ were computed in $O(n)$ flops according to (3.8) (using $y = e$ to compute $\kappa_\infty(A)$). The results are given in Table 6.1.

For our test problem, (1.5) takes the form (using (3.4))

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \leq 10.9 \text{ cond } (A, x)u. \tag{6.1}$$

TABLE 6.1
Numerical results, n = 50.

cond(A) = 1.33E6, $\kappa_\infty(A)$ = 1.85E6
$p = e_n + e_{n-1} + \cdots + e_{n-4}$
$q = (1, \alpha, \alpha^2, \cdots, 10^{-5})$, $\alpha = 10^{-5/(n-1)}$
rand = vector with random elements from uniform $[-1, 1]$ distribution

	$x = p$	$x = e_1$	$x = q$	$x = e$	x = rand	$b = e_n$
cond(A, x)	1.73E2	3.82E0	8.89E3	1.33E6	5.50E5	8.87E5
$\dfrac{\|x - \hat{x}\|_\infty}{u\|x\|_\infty}$	1.25E0	0.00E0	9.92E2	1.42E2	1.75E4	1.09E2

In the traditional bound (1.2) there is a similar constant and cond (A, x) is replaced by $\kappa_\infty(A)$. From Table 6.1 we see that in the first three cases cond (A, x) is significantly smaller than cond (A) and $\kappa_\infty(A)$; this indicates the value of using a condition number that depends on x. The bound (6.1) is of variable sharpness, but it is always smaller than the traditional bound.

Acknowledgment. Des Higham helped to polish the presentation.

REFERENCES

[1] M. ARIOLI, J. W. DEMMEL, AND I. S. DUFF, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 165–190.

[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[3] C. H. BISCHOF, J. W. DEMMEL, J. J. DONGARRA, J. J. DU CROZ, A. GREENBAUM, S. J. HAMMARLING, AND D. C. SORENSEN, Provisional contents, LAPACK Working Note No. 5, Report ANL-88-38, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1988.

[4] C. W. CRYER, *The LU-factorization of totally positive matrices*, Linear Algebra Appl., 7 (1973), pp. 83–92.

[5] C. DE BOOR AND A. PINKUS, *Backward error analysis for totally positive linear systems*, Numer. Math., 27 (1977), pp. 485–490.

[6] J. W. DEMMEL, J. J. DONGARRA, J. J. DU CROZ, A. GREENBAUM, S. J. HAMMARLING, AND D. C. SORENSEN, *Prospectus for the development of a linear algebra library for high-performance computers*, Tech. Memorandum No. 97, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1987.

[7] J. W. DEMMEL, J. J. DU CROZ, S. J. HAMMARLING, AND D. C. SORENSEN, *Guidelines for the design of symmetric eigenroutines, SVD, and iterative refinement and condition estimation for linear systems*, LAPACK Working Note No. 4, Tech. Memorandum 111, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1988.

[8] J. W. DEMMEL AND W. KAHAN, *Computing small singular values of bidiagonal matrices with guaranteed high relative accuracy*, LAPACK Working Note No. 3, Tech. Memorandum 110, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1988; SIAM J. Sci. Statist. Comput., 11 (1990), to appear.

[9] J. J. DONGARRA, Private communication, 1988.

[10] J. J. DONGARRA, J. R. BUNCH, C. B. MOLER, AND G. W. STEWART, LINPACK *Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[11] F. W. DORR, *An example of ill-conditioning in the numerical solution of singular perturbation problems*, Math. Comp., 25 (1971), pp. 271–283.

[12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[13] M. D. GUNZBURGER AND R. A. NICOLAIDES, *Stability of Gaussian elimination without pivoting on tridiagonal Toeplitz matrices*, Linear Algebra Appl., 45 (1982), pp. 21–28.

[14] W. J. HARROD, *LU-decompositions of tridiagonal irreducible H-matrices*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 180–187.

[15] N. J. HIGHAM, *Efficient algorithms for computing the condition number of a tridiagonal matrix*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 150–165.

[16] ———, *Algorithm* 674: FORTRAN *codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation*, ACM Trans. Math. Software, 14 (1988), pp. 381–396.

[17] ———, *The accuracy of solutions to triangular systems*, SIAM J. Numer. Anal., 26 (1989), pp. 1252–1265.

[18] A. NEUMAIER, *On the comparison of H-matrices with M-matrices*, Linear Algebra Appl., 83 (1986), pp. 135–141.

[19] R. D. SKEEL, *Scaling for numerical stability in Gaussian elimination*, J. Assoc. Comput. Mach., 26 (1979), pp. 494–526.

[20] ———, *Iterative refinement implies numerical stability for Gaussian elimination*, Math. Comp., 35 (1980), pp. 817–832.

[21] G. W. STEWART, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[22] J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 531–536, October 1990

004

SIMULTANEOUS DIAGONALISATION OF SEVERAL HERMITIAN MATRICES*

PAUL BINDING†

Abstract. Simultaneous diagonalisation by congruence is shown to be equivalent to solubility of certain linked generalised eigenvalue problems. Various connections are made between related ideas as consequences.

Key words. congruent diagonalisation, unitary diagonalisation, linked eigenvalue problems, generalised inverse

AMS(MOS) subject classification. 15A99

1. Introduction. The $J \times J$ matrices $B_1, \cdots, B_M$ are called simultaneously (real) diagonable by a similarity transformation X if $X^{-1}B_mX$ is real diagonal for $m = 1, \cdots, M$. We abbreviate this to B_m are SDS (by X if we need to specify the transformation). Also B_m are IDS if they are individually diagonable by a similarity transform, i.e., if X is allowed to depend on m. B_m are simultaneously diagonable by congruence (SDC) if there is a nonsingular $J \times J$ matrix X such that X^*B_mX is real diagonal for $m = 1, \cdots, M$. It is our purpose to connect these and related ideas, and in particular to characterize SDC for an arbitrary Hermitian $(M + 1)$-tuple $A_o, \cdots, A_M$. For simplicity we assume the scalars to be complex, although there are other possibilities, cf. [2].

Our key tool is a system of generalised eigenvalue problems

$$A_m\mathbf{x} = \lambda_m A\mathbf{x}, \qquad m = 1, \cdots, M \tag{1}$$

for a given linear combination $A = A(\omega) = A_o + \sum_{j=1}^{M} \omega_j A_j$.

Remark 1.1. Equation (1) is equivalent to the same system but for $m = 0, \cdots, M$, provided we set

$$\lambda_o = 1 - \sum_{j=1}^{M} \omega_j \lambda_j.$$

Systems like (1) and their homogeneous counterparts,

$$\mu_\ell A_m\mathbf{x} = \mu_m A_\ell \mathbf{x}, \qquad 0 \leqq \ell, \qquad m \leqq M, \tag{2}$$

have been used extensively in multiparameter spectral theory for tensor determinants A_m. For example Atkinson [1] gives various extra conditions guaranteeing solubility of (1) and/or (2) and SDC, perhaps by X which is unitary in various inner products. Here we say that (1) (respectively, (2)) is soluble by X if there is a nonsingular matrix X such that the jth column $\mathbf{x}$ of X satisfies (1) (respectively, (2)) for some j-dependent $\lambda \in \mathbb{R}^M$ (respectively, nonzero $\mu \in \mathbb{R}^{M+1}$).

In the following we drop the tensor setting and the extra conditions, and we investigate the relation between the above concepts in their own right. Our results, given in § 2, may be conveniently discussed via two observations, the first being as follows.

* Received by the editors September 11, 1987; accepted for publication (in revised form) December 12, 1989. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

† Department of Mathematics and Statistics, University of Calgary, Alberta, Canada T2N 1N4 (binding@UNCAMULT.BITNET).

THEOREM 1.2. *The following are equivalent* (TFAE):

(i) $A_o, \cdots, A_M$ *are* SDC.

(ii) (1) *is soluble for all* ω *in some dense open set* $\Omega \subset \mathbb{R}$.

(ii') (1) *is soluble for some* $\omega \in \mathbb{R}$.

(iii) (2) *is soluble*.

We remark that Ω is the complement of J hyperplanes. Some special cases of the literature for $k > 1$ are immediately accessible.

COROLLARY 1.3 [4, Thm. 1], [5, Thm. 6.5.3]. *If* A *is nonsingular for some* $\omega \in \mathbb{R}^M$, *then* TFAE:

(i) $A_o, \cdots, A_M$ *are* SDC.

(ii) $A^{-1}A_m$ *are* SDS *for* $m = 1, \cdots, M$.

(iii) $A^{-1}A_m$ *are* SDS *for* $m = 0, \cdots, M$.

Proof.

$$A_m \text{ are SDC} \Leftrightarrow A^{-1}A_m\mathbf{x} = \lambda_m\mathbf{x}, \qquad m = 1, \cdots, M \text{ by (1)},$$

$$\Leftrightarrow A^{-1}A_mX = X\Lambda_m \quad \text{for real diagonal } \Lambda_m.$$

This proves (i) $\Leftrightarrow$ (ii) and (ii) $\Leftrightarrow$ (iii) follows from Remark 1.1. □

Remark 1.4. The conditions in (ii) and (iii) may be expressed as commutativity and IDS, as is well known.

COROLLARY 1.5 [4, Cor. 1] [6, Thm. 2]. *If* A *is positive definite for some* $\omega \in \mathbb{R}$, *then* TFAE:

(i) A_m *are* SDC.

(ii) $A^{-1}A_m$ *are* SDS.

(iii) $A^{-1}A_m$ *commute*.

Proof. Corollary 1.3 gives (ii) $\Rightarrow$ (i), Remark 1.4 gives (i) $\Rightarrow$ (ii), and (iii) $\Rightarrow$ (ii) follows from the spectral theorem for the commuting matrices $A^{-1}A_m$, which are Hermitian in the inner product defined by

$$(\mathbf{x}, \mathbf{y})_A = \mathbf{x}^*A\mathbf{y}. \tag{3}$$

□

We shall extend this result in Corollary 2.6.

If any of the conditions of Theorem 1.2 is satisfied, then they all are, and we thus have: each $X^*A_mX = \Lambda_m$, say (a diagonal matrix) for some $X = X_{(i)}$, (1) is soluble by some $X = X_\omega$ and (2) is soluble by some $X = X_{(2)}$.

Our second observation concerns the relation between these transformation matrices X and the subspace $N = \cap_{m=0}^{M} N(A_m)$.

THEOREM 1.6. *Suppose that the conditions* (i)–(iv) *of Theorem* 1.2 *are satisfied as above. Then they are also satisfied, with the same* Λ_m, λ, *and* μ, *by a common transformation matrix* $X = Z$ *which is independent of* $\omega \in \Omega$. *Moreover the columns of* Z *may be split into subsets* U *and* V *where*

(i) *the columns of* U *form an orthonormal basis of* N,

(ii) *the columns of* U *are orthogonal to those of* V,

(iii) *two columns of* V *are orthogonal if the corresponding* λ (*respectively*, μ) *are equal* (*respectively*, *proportional*).

This enables us to give conditions equivalent to SDC by a matrix that is unitary, or $_A$-unitary in the sense of (3) (see Corollaries 2.5 and 2.6). In general we may split off N by means of $Z = [U{:}V]$ giving

$$Z^*A_mZ = \begin{bmatrix} 0 & 0 \\ 0 & B_m \end{bmatrix},$$

where B_m satisfy Corollary 1.3. A more precise statement follows.

COROLLARY 1.7. TFAE:

(i) A_m *are* SDC.

(ii) *For all nonsingular* $Z = [U:V]$ *where the columns of* U *span* N, *and for all* $\omega \in \Omega$,

(a) $B = V^*AV$ *is nonsingular*,

(b) $B^{-1}B_m$ *are* SDS, *where* $B_m = V^*A_mV$.

(ii′) *As* (ii) *but with "all" replaced by "some."*

(iii) *For all* $\omega \in \Omega$,

(a) $N(A) = N$,

(b) *for all* A^- *satisfying* $AA^-A = A$, A^-A_m *are* SDS.

(iii′) *As* (iii) *but with "all" replaced by "some."*

By virtue of Remark 1.4, (i) $\Leftrightarrow$ (iii′) strengthens a result stated by Rao and Mitra [6, Cor., p. 134]. Replacing Corollary 1.3 by Corollary 1.5 in Corollary 1.7, we obtain the following.

COROLLARY 1.8. *If for some* $\omega \in \Omega$, A *is nonnegative definite then* TFAE:

(i) A_m *are* SDC.

(ii) $B^{-1}B_m$ *commute*.

(iii) A^-A_m *commute*.

An equivalent result is given in [5, Thm. 6.5.2]. In the special case when *all* the A_m are nonnegative definite, one may take $\omega = (1, 1, \cdots, 1)$, i.e., $A = \sum_{m=0}^{M} A_m$ (see [5, Rem., p. 133], [6, Thm. 1]).

We conclude this introduction with some comments on the well-studied case $k = 1$. Then the equivalence (i) $\Leftrightarrow$ (iii) of Theorem 1.2 is due to Au-Yeung [2]. Corollary 1.3 (i) $\Leftrightarrow$ (ii) is well known and Corollary 1.5 (where (iii) is vacuous, so (i) is automatic) is standard. The actual statement of Corollary 1.7 may be new, but (ii)(a)–(b) correspond, respectively, to vanishing minimal indices and linear elementary divisors, i.e., to the classical conditions of Kronecker for (i). For further results and references, we cite [3], [4], and [5].

2. Results and proofs. In this section we prove enough to substantiate the results of § 1.

LEMMA 2.1. *Suppose* $X^*A_mX = \Lambda_m$ *where* $\Lambda_m = \operatorname{diag}(\lambda_m^1, \cdots, \lambda_m^J)$.

(i) *If* $\nu = (\lambda_o^j, \cdots, \lambda_M^j) \neq \mathbf{0}$, *then* (2) *is satisfied by* $\mu = \nu$ *and the* j*th column* $\mathbf{x}$ *of* X.

(ii) *If* $\nu = \mathbf{0}$, *then* (2) *is satisfied by any* μ *and the* j*th column* $\mathbf{x}$ *of* X.

Proof.

(i) $\mu_\ell A_m\mathbf{x} = \lambda_\ell^j$ (jth column of $X^{-*}\Lambda_m$)
$= \lambda_\ell^j\lambda_m^j$ (jth column of X^{-*}),

which is obviously symmetric in ℓ and m.

(ii) For each m, $A_m\mathbf{x} = X^{-*}\mathbf{0} = \mathbf{0}$ so $\mu_mA_\ell\mathbf{x} = \mu_\ell A_m\mathbf{x} = \mathbf{0}$ for all $\mu \in \mathbb{R}^{M+1}$. □

LEMMA 2.2. *If* (2) *is satisfied by* $\mu = \mu^j$ *and the* j*th column* $\mathbf{x}$ *of* X, *and if*

$$\mu^j = \mu^j(\omega) = \mu_o^j + \sum_{m=1}^{M} \omega_m\mu_m^j \tag{4}$$

is nonzero for some $\omega \in \mathbb{R}^M$, *then* (1) *is satisfied by* $\lambda = (\mu_1^j, \cdots, \mu_M^j)/\mu^j$ *with the same* $\mathbf{x}$.

Proof. Equation (2) yields

$$\left(\mu_o^j + \sum_{\ell=1}^{M} \omega_\ell\mu_\ell^j\right)A_m\mathbf{x} = \mu_m^j\left(A_o + \sum_{\ell=1}^{M} \omega_\ell A_\ell\right)\mathbf{x},$$

i.e.,

$$\mu^j A_m \mathbf{x} = \mu^j_m A \mathbf{x}. \qquad \square$$

Now define Ω as the complement of the J hyperplanes $0 = \mu^j$ of (4), i.e., $\omega \notin \Omega \Leftrightarrow \mu^j = 0$ for some $j = 1, \cdots, M$. Then Lemmas 2.1–2.2 yield the following corollary.

COROLLARY 2.3. *If A_m are* SDC *by X, then* (2) (*and* (1) *for all $\omega \in \Omega$*) *are soluble by X.*

To establish Theorem 1.2, it is enough therefore to prove (ii′) $\Rightarrow$ (i). Accordingly suppose that (1) is soluble by X, with jth column $\mathbf{x}^j$ corresponding to $\lambda = \lambda^j$. We write $\mathbf{y}^j$ for the orthogonal projection of $\mathbf{x}^j$ onto $N^\perp$, provided $\mathbf{x}^j \notin N$, and Y^k for the span of all $\mathbf{y}^j$ such that $\lambda^j = \lambda^k$. We use the notation (3) regardless of whether A is definite, and we call two subspaces U and V $_A$-orthogonal if $(\mathbf{u}, \mathbf{v})_A = 0$ for all $\mathbf{u} \in U$, $\mathbf{v} \in V$.

LEMMA 2.4. *The subspaces Y^k are $_A$-orthogonal and together span $N^\perp$.*

Proof. If $\lambda^j \neq \lambda^k$, then $\lambda^j_m \neq \lambda^k_m$ for some m, so

$$\lambda^j_m(\mathbf{y}^k, \mathbf{y}^j)_A = \lambda^j_m(\mathbf{x}^k, \mathbf{x}^j)_A = (\mathbf{x}^k)^* A_m \mathbf{x}^j$$

$$= (\mathbf{x}^j)^* A_m \mathbf{x}^k = \lambda^k_m(\mathbf{x}^j, \mathbf{x}^k)_A = \lambda^k_m(\mathbf{y}^j, \mathbf{y}^k)_A.$$

Thus $(\mathbf{y}^j, \mathbf{y}^k)_A = 0$, and so Y^j and Y^k are $_A$-orthogonal.

Let $\mathbf{y}^j = \mathbf{x}^j + \mathbf{n}^j$ where $\mathbf{n}^j \in N$ and $\mathbf{y}^j = \mathbf{0}$ if $\mathbf{x}^j \in N$. Given $\mathbf{x} \in N^\perp$,

$$\mathbf{x} = \Sigma^j_\alpha(\mathbf{y}^j - \mathbf{n}^j)$$

for some $\alpha^j \in \mathbb{C}$. Thus

$$\Sigma^j_\alpha \mathbf{n}^j \in N \cap N^\perp$$

and so indeed $\mathbf{x}$ is in the span of the Y^j. $\square$

We now construct a matrix Z whose columns corresponding to $\mathbf{x}^j \in N$ form an orthonormal and $_A$-orthogonal basis of N, and whose remaining columns make up corresponding bases of the Y^k. By Lemma 2.4, Z is nonsingular and Z^*AZ is diagonal, so by (1), Z^*A_mZ is also diagonal. This completes the proof of Theorem 1.2.

Proof of Theorem 1.6. By Corollary 2.3, (1) is soluble by X, so we may construct Z (perhaps ω-dependent) as above. Now Lemma 2.2 shows that if $\lambda^j = \lambda^k$ for some $\omega \in \Omega$ then $\lambda^j = \lambda^k$ for all $\omega \in \Omega$. Thus the Y^k are ω-independent, so Z may be chosen independently of ω.

A simple computation shows that $A\mathbf{x}^j = A\mathbf{z}^j$ and $A_m\mathbf{x}^j = A_m\mathbf{z}^j$ for all j and m. Thus (1) is soluble by Z with the same vectors λ. Similarly, A_m are SDC by X and Z with the same Λ_m, and so (2) is soluble by X and Z with the same vectors μ. Finally, conclusions (i)–(iii) follow from Lemma 2.2 and the construction of Z. $\square$

COROLLARY 2.5. TFAE:

(i) *A_m are* SDC *by a unitary matrix.*

(ii) (2) (*or* (1) *for all $\omega \in \Omega$*) *is soluble by a unitary matrix.*

(iii) *A_m commute.*

Proof. (i) $\Leftrightarrow$ (iii) as for Corollary 1.5, and (i) $\Rightarrow$ (ii) by Corollary 2.3. If (ii) holds then the Y^k are orthogonal, so it suffices to choose orthonormal bases of N and the Y^k for the columns of Z. $\square$

We call vectors μ^j $_A$-orthonormal if $(\mathbf{u}^j, \mathbf{u}^j)_A = \delta_{ij}$. A $J \times J$ matrix U is $_A$-unitary if its columns are $_A$-orthonormal. Arguing as above but with $_A$-orthonormal bases for the Y^k, we can augment Corollary 1.5 as follows.

COROLLARY 2.6. TFAE:

(i) A_m *are* SDC *by an* $_A$*-unitary matrix for some* $\omega \in \Omega$.

(ii) A_m *are* SDC *and* A *is positive definite for some* $\omega \in \Omega$.

(iii) $A^{-1}A_m$ *commute and* A *is positive definite for some* $\omega \in \Omega$.

Proof of Corollary 1.7. (i) $\Rightarrow$ (iii) by construction $N \subset N(A)$ and the converse follows from Theorem 1.2 (i) $\Rightarrow$ (ii). This proves (a) and to prove (b) we shall assume that (1) is soluble by X with columns ordered so that

$$X^*AX = \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix}, \qquad X^*A_mX = \begin{bmatrix} 0 & 0 \\ 0 & D_m \end{bmatrix},$$

where D and D_m are real diagonal matrices with D nonsingular. (If the partition is improper then either each $A_m = 0$ or else A is nonsingular and the results follow from Corollary 1.3.)

An easy computation shows that

$$A^- = X \begin{bmatrix} Q & R \\ S & D^{-1} \end{bmatrix} X^*$$

for some Q, R, and S of appropriate sizes. Thus

$$A^-A_m = XW_mX^{-1},$$

where

$$W_m = \begin{bmatrix} 0 & RD_m \\ 0 & D^{-1}D_m \end{bmatrix},$$

and it is clear that W_m has eigenvalues consisting of zero and the diagonal entries of $D^{-1}D_m$. Moreover W_m has dim N zero columns, and hence has rank at most J-dim N. If $D^{-1}D_m$ has an eigenvalue λ repeated ℓ times then $W_m - \lambda I$ has ℓ zero rows, and hence has rank at most $J - \ell$. As a result, W_m are IDS and since they clearly commute, they are SDS by Remark 1.4, so A^-A_m are SDS.

(iii′) $\Rightarrow$ (ii) Let $Z^*AZ = \left[\begin{smallmatrix} 0 & 0 \\ 0 & B \end{smallmatrix}\right]$, $Z^*A_mZ = \left[\begin{smallmatrix} 0 & 0 \\ 0 & B_m \end{smallmatrix}\right]$, so (iii′)(a) $\Rightarrow$ (ii)(a). Writing

$$A^- = Z \begin{bmatrix} Q & R \\ S & B^{-1} \end{bmatrix} Z^*,$$

we have

$$A^-A_m = ZC_mZ^{-1},$$

where

$$C_m = \begin{bmatrix} 0 & RB_m \\ 0 & B^{-1}B_m \end{bmatrix}.$$

Thus C_m are SDS, say

$$C_m \begin{bmatrix} F & G \\ H^1 & H^2 \end{bmatrix} = \begin{bmatrix} F & G \\ H^1 & H^2 \end{bmatrix} \begin{bmatrix} D_m^1 & 0 \\ 0 & D_m^2 \end{bmatrix},$$

where D_m^1 and D_m^2 are real diagonal. We obtain $B^{-1}B_m[H^1H^2] = [H^1D_m^1\ H^2D_m^2]$, and if H is a nonsingular matrix whose columns are a subset of those of H^1 and H^2, we have

$$B^{-1}B_mH = HD_m,$$

where D_m is a diagonal matrix whose diagonal entries are a subset of those of D_m^1 and D_m^2.

(ii′) $\Rightarrow$ (i) By Corollary 1.3 (iii) $\Rightarrow$ (i), there are nonsingular T and real diagonal D_m such that $T^*B_mT = D_m$. Thus

$$Y^*A_mY = \begin{bmatrix} 0 & 0 \\ 0 & D_m \end{bmatrix},$$

where

$$Y = \left[U \vdots \begin{matrix} S \\ \vdots \\ T \end{matrix} \right]$$

for any S of the appropriate size. It suffices to choose S so that Y is nonsingular. □

REFERENCES

[1] F. V. ATKINSON, *Multiparameter Eigenvalue Problems*, Vol. 1, Academic Press, New York, 1972.

[2] Y. H. AU-YEUNG, *A necessary and sufficient condition for the simultaneous diagonalization of two Hermitian matrices and its application*, Glasgow Math. J., 11 (1970), pp. 81–83.

[3] R. I. BECKER, *Necessary and sufficient conditions for the simultaneous diagonability of two quadratic forms*, Linear Algebra Appl., 30 (1980), pp. 129–139.

[4] C. R. JOHNSON, *Simultaneous diagonalization of several matrices by congruence*, preprint.

[5] C. R. RAO AND S. K. MITRA, *Generalized Inverse of Matrices and its Applications*, John Wiley, New York, 1971.

[6] W. QUIGUAUG, *Necessary and sufficient conditions for simultaneous diagonalization of several matrices*, Sci. Exploration, 3 (1983), pp. 41–44.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 537–552, October 1990

005

SOLVING THE GENERALIZED EIGENVALUE PROBLEM FOR RATIONAL TOEPLITZ MATRICES*

DARIO BINI† AND FABIO DI BENEDETTO‡

Abstract. The generalized eigenvalue problem $\mathbf{Ax} = \lambda\mathbf{Bx}$ for Toeplitz matrices generated by a rational function is considered from a computational point of view. Three different functions having the same zeros as the polynomial $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{B})$ are introduced and fast methods for the evaluation at a point of these functions and their derivatives are presented. Such methods, which can be used for the numerical computation of the eigenvalues as zeros of $p(\lambda)$, generalize some results holding in the particular case of banded Toeplitz matrices.

Key words. Toeplitz matrices, generalized eigenvalue problem, computational complexity

AMS(MOS) subject classifications. 15A18, 65F15

1. Introduction.

Let $R(z)$ be a complex rational function defined by

$$R(z) = \frac{C(z)}{A(z)B(z^{-1})},$$

$$A(z) = \sum_{\mu=0}^{r} a_\mu z^\mu, \quad B(z) = \sum_{\nu=0}^{s} b_\nu z^\nu, \quad C(z) = \sum_{i=-q}^{p} c_i z^i, \tag{1.1}$$

where a_r, b_s, c_{-q}, and c_p are nonzero. The formal Laurent series $\sum_{j=-\infty}^{+\infty} r_j z^j$, associated to the function $R(z)$, is defined by the condition

$$R(z)A(z)B(z^{-1}) = C(z).$$

This series defines, for any positive integer n, the $n \times n$ Toeplitz matrix $\mathbf{R}_n = (r_{j-i})_{i,j=0}^{n-1}$. We refer to the matrix $\mathbf{R}_n$ as the rational Toeplitz matrix generated by the function $R(z)$. Observe that if $A(z) = 1$ and $B(z) = 1$, the matrix $\mathbf{R}_n$ is a banded Toeplitz matrix.

Given the rational Toeplitz matrices $\mathbf{T}_n$ and $\mathbf{S}_n$ generated by the functions

$$T(z) = \frac{\gamma(z)}{\alpha(z)\beta(z^{-1})}, \qquad S(z) = \frac{\eta(z)}{\delta(z)\varepsilon(z^{-1})},$$

respectively, we consider the following generalized eigenvalue problem

$$\mathbf{T}_n\mathbf{u} = \lambda\mathbf{S}_n\mathbf{u}, \mathbf{u} \neq 0. \tag{1.2}$$

Problems of this kind occur in several applications: for instance, in the optimization of rejection filters the maximization of the signal-to-noise ratio leads to the computation of the minimum λ in (1.2), where $\mathbf{T}_n$ is the interference covariance matrix and $\mathbf{S}_n$ is the signal covariance matrix [5].

Recently most interest has been devoted to the standard eigenvalue problem (where $\mathbf{S}_n = \mathbf{I}$) for rational Toeplitz matrices (see [10], [11], [12]) and to the specific case of banded block Toeplitz matrices ([1], [2], [3], [9], [13]). Explicit formulae have been given for the polynomial $p_n(\lambda) = \det(\mathbf{T}_n - \lambda\mathbf{I}_n)$ and efficient computational methods have been devised.

* Received by the editors June 6, 1988; accepted for publication (in revised form) June 22, 1989.
† Dipartimento di Matematica, Università di Pisa, via Buonarroti 2, Pisa, Italy (bini@icnucevm.bitnet).
‡ Dipartimento di Matematica, Università di Genova, via Leon Battista Alberti 4, 16132 Genova, Italy.

In [13] a method for computing $p_n(\lambda)$, which uses the approximation of the zeros of an auxiliary polynomial of degree independent of n, is proposed for the case where $\mathbf{T}_n$ is a banded Toeplitz matrix. This method has been extended to the case of Toeplitz matrices generated by a rational function in [10], [11], [12]. As shown in [11] this approach can be used together with root-finders (such as bisection or "regula falsi") applied to the equation $p_n(\lambda) = 0$, which use only the value that $p_n(\lambda)$ takes on at a point λ. On the other hand, since computable formulae have not been provided for the ratio $p_n(\lambda)/p_n'(\lambda)$, Newton's method has never been taken into account. Moreover numerical stability problems may arise if the degree of the auxiliary polynomial is big enough.

In [3] these restrictions have been overcome by devising new methods for the evaluation of $p_n(\lambda)$, in the case of a banded Toeplitz matrix T_n. These methods do not need any asymptotic stage, such as the approximation of the zeros of a polynomial, so that they work over any field of numbers (not necessarily algebraically closed as in [10], [11], [12], [13]). Moreover, these methods also compute, at a cost which is roughly doubled, the ratio $p_n(\lambda)/p_n'(\lambda)$, so that Newton's method can be applied efficiently. Finally, they can be easily extended to the case of block matrices.

In this paper we are interested in generalizing the algorithms given in [3] to the case of (1.2), where the matrices $\mathbf{T}_n$ and $\mathbf{S}_n$ are rational Toeplitz matrices.

Since λ is a solution of (1.2) if and only if

$$\det \mathbf{R}_n = 0, \tag{1.3}$$

where $\mathbf{R}_n = \mathbf{T}_n - \lambda \mathbf{S}_n$ is the Toeplitz matrix generated by the rational function

$$\begin{aligned} R(z,\lambda) &= \frac{C(z,\lambda)}{A(z)B(z^{-1})}, \\ C(z,\lambda) &= \gamma(z)\delta(z)\varepsilon(z^{-1}) - \lambda\eta(z)\alpha(z)\beta(z^{-1}), \\ A(z) &= \alpha(z)\delta(z), \\ B(z) &= \beta(z)\varepsilon(z), \end{aligned} \tag{1.4}$$

we can restate (1.2) as a singularity problem for a Toeplitz matrix generated by a rational function. By using Greville and Trench's lemma [6] we rewrite condition (1.3) in terms of the solution of a homogeneous constant-coefficient linear difference equation (§ 2). Following [13] and [3], we can reduce the computation of $\det \mathbf{R}_n$ in (1.3) to the computation of the determinant of a $k \times k$ matrix whose order k is independent of n and whose entries are computable with a low computational cost (§ 3). This can be achieved by solving the difference equation in either of three different ways: representing the solution of the difference equation by means of the zeros of a suitable polynomial as in [13]; representing the solution in terms of the integer powers of a companion matrix as in [3]; or applying the cyclic reduction method that is customarily used in the solution of certain block tridiagonal systems. This way we obtain three functions, $\Delta_n(\lambda)$, $\tilde{\Delta}_n(\lambda)$, $\hat{\Delta}_n(\lambda)$, having the same zeros as $\det \mathbf{R}_n$.

In § 4 we apply the results of § 3 to devise computational methods for the numerical evaluation of the functions $\Delta_n(\lambda)$, $\tilde{\Delta}_n(\lambda)$, $\hat{\Delta}_n(\lambda)$. The case of $\Delta_n(\lambda)$ is treated as in [10], [11], [12], while for $\tilde{\Delta}_n(\lambda)$ and $\hat{\Delta}_n(\lambda)$ we use a generalization, to the case of rational matrices, of the methods given in [3] for the case of banded Toeplitz matrices. An outline of the procedures to compute the ratios $\Delta_n(\lambda)/\Delta_n'(\lambda)$, $\tilde{\Delta}_n(\lambda)/\tilde{\Delta}_n'(\lambda)$ and $\hat{\Delta}_n(\lambda)/\hat{\Delta}_n'(\lambda)$ is also given.

In § 5 we present an explicit formula for the characteristic polynomial $p_n(\lambda) = \det(\mathbf{T}_n - \lambda \mathbf{I}_n)$ of a rational Toeplitz matrix $\mathbf{T}_n$. It generalizes an analogous formula given in [3] for the particular case where $\mathbf{T}_n$ is banded. Unlike the representation of $p_n(\lambda)$ given in [12], this formula gives $p_n(\lambda)$ in terms of quantities that are computable with a finite number of arithmetic operations.

In § 6, considering a concrete example occurring in the optimization of rejection filters, we show how the conditions under which the results of § 3 hold can be removed.

2. Reduction to a linear difference problem. We want to state the singularity condition of a rational Toeplitz matrix in terms of the solution of a linear difference equation. So let $\mathbf{R}_n$ be the $n \times n$ Toeplitz matrix generated by the rational function $R(z)$ defined in (1.1), set

$$M = \max(p, r), N = \max(q, s), k = M + N,$$

and assume that

$$\begin{aligned} &a_0 b_0 \neq 0, \\ &\gcd(A(z), z^s B(z^{-1})) = 1, \\ &\gcd(z^q C(z), z^s B(z^{-1})) = 1, \\ &\gcd(A(z), z^q C(z)) = 1. \end{aligned} \tag{2.1}$$

That is, the polynomials $A(z)$, $z^s B(z^{-1})$ and $z^q C(z)$, are pairwise relatively prime.

Throughout the paper we will denote by $\theta_{-s}, \cdots, \theta_r$ the coefficients of the function $\theta(z) = A(z)B(z^{-1})$, that is, $\theta(z) = \theta_{-s} z^{-s} + \cdots + \theta_r z^r$; moreover, we assume that every coefficient of $A(z)$, $B(z)$, $C(z)$, $\theta(z)$ is zero if its subscript is out of range.

Consider the matrices

$$\begin{aligned} \hat{\mathbf{A}} &= \begin{pmatrix} a_0 & \cdots & a_{M-1} \\ & \ddots & \vdots \\ & & a_0 \end{pmatrix} \in \mathbf{C}^{M \times M}, \\ \hat{\mathbf{B}} &= \begin{pmatrix} b_0 & & \\ \vdots & \ddots & \\ b_{N-1} & \cdots & b_0 \end{pmatrix} \in \mathbf{C}^{N \times N}, \\ \mathbf{A} &= \begin{pmatrix} a_0 & \cdots & a_M & & \\ & \ddots & & \ddots & \\ & & a_0 & \cdots & a_M \end{pmatrix} \in \mathbf{C}^{N \times k}, \\ \mathbf{B} &= \begin{pmatrix} b_N & \cdots & b_0 & & \\ & \ddots & & \ddots & \\ & & b_N & \cdots & b_0 \end{pmatrix} \in \mathbf{C}^{M \times k}, \end{aligned} \tag{2.2}$$

and observe that, since $a_0 b_0 \neq 0$, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are nonsingular matrices.

The following result allows us to express the singularity condition of the matrix $\mathbf{R}_n$ in terms of the solution of a linear difference equation of order k. The formulation of this result is slightly different from that given originally in [6] and is more useful for our purposes.

LEMMA 1 (Greville and Trench). *Under the hypotheses* (2.1) *there exists a unique sequence* $\{\phi_i\}_{i=-\infty}^{+\infty}$ *such that*

$$\sum_{\mu=0}^{r} a_\mu \phi_{i-\mu} = b_0^{-1}\delta_{i,0}, \qquad \sum_{\nu=0}^{s} b_\nu \phi_{\nu-i} = a_0^{-1}\delta_{i,0}, \text{ for } i \geqq 0,$$

where $\delta_{i,j}$ *is Kronecker's symbol. Moreover the following properties hold*:

(i) $\sum_{i=-\infty}^{+\infty} \phi_i z^i$ *is the formal Laurent series of the rational function*

$$R_0(z) = \frac{1}{A(z)B(z^{-1})};$$

(ii) $\mathbf{R}_n = \mathbf{C}^{(n)}\mathbf{\Phi}^{(n)}$, *where*

$$\mathbf{C}^{(n)} = \begin{pmatrix} c_{-N} & \cdots & c_0 & \cdots & c_M & & \\ & \ddots & & \ddots & & \ddots & \\ & & c_{-N} & \cdots & c_0 & \cdots & c_M \end{pmatrix} \in \mathbf{C}^{n\times(n+k)},$$

$$\mathbf{\Phi}^{(n)} = \begin{pmatrix} \phi_N & \cdots & \phi_0 & \cdots & \phi_{-M} & \cdots & \phi_{-n-M+1} \\ \vdots & \ddots & & \ddots & & \ddots & \vdots \\ \phi_{N+n-1} & \cdots & \phi_N & \cdots & \phi_0 & \cdots & \phi_{-M} \end{pmatrix}^T \in \mathbf{C}^{(n+k)\times n};$$

(iii) *the matrix* $\mathbf{\Phi}_{n+k} = (\phi_{j-i})_{i,j=0}^{n+k-1}$ *is nonsingular and its inverse* $\mathbf{H}_{n+k}$ *has the following structure*

$$\mathbf{H}_{n+k} = \begin{pmatrix} \mathbf{H}^{(1)} \\ \mathbf{H}^{(2)} \\ \mathbf{H}^{(3)} \end{pmatrix},$$

$$\mathbf{H}^{(1)} = \hat{\mathbf{B}}(\mathbf{A} \quad \mathbf{0}) \in \mathbf{C}^{N\times(n+k)},$$

$$\mathbf{H}^{(2)} = \begin{pmatrix} \theta_{-N} & \cdots & \theta_0 & \cdots & \theta_M & & \\ & \ddots & & \ddots & & \ddots & \\ & & \theta_{-N} & \cdots & \theta_0 & \cdots & \theta_M \end{pmatrix} \in \mathbf{C}^{n\times(n+k)},$$

$$\mathbf{H}^{(3)} = \hat{\mathbf{A}}(\mathbf{0} \quad \mathbf{B}) \in \mathbf{C}^{M\times(n+k)},$$

where $\mathbf{A}$, $\mathbf{B}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$ *are the matrices defined in* (2.2).

Now we can state the condition $\det \mathbf{R}_n = 0$ in terms of a difference equation. In fact, if $\mathbf{u} \in \mathbf{C}^n$ is such that $\mathbf{R}_n\mathbf{u} = \mathbf{0}$, then from Lemma 1, part (ii), we deduce $\mathbf{C}^{(n)}\mathbf{\Phi}^{(n)}\mathbf{u} = \mathbf{0}$, hence the $(n+k)$-vector $\mathbf{v} = \mathbf{\Phi}^{(n)}\mathbf{u}$ belongs to the kernel of the rectangular Toeplitz matrix $\mathbf{C}^{(n)}$, that is

$$\sum_{j=-N}^{M} c_j v_{j+i} = 0, \qquad 0 \leqq i \leqq n-1; \tag{2.3}$$

moreover, $\mathbf{v} \neq \mathbf{0}$ since the matrix $\mathbf{\Phi}^{(n)}$ has full rank. This is a homogeneous constant-coefficient linear difference equation of order k, which can be completed by adding a set of boundary conditions as follows. First observe that $\mathbf{\Phi}^{(n)}$ is a submatrix of the matrix $\mathbf{\Phi}_{n+k}$ defined in Lemma 1 so that we have

$$\mathbf{v} = \mathbf{\Phi}_{n+k}\begin{pmatrix} \mathbf{0}_N \\ \mathbf{u} \\ \mathbf{0}_M \end{pmatrix}, \mathbf{H}_{n+k}\mathbf{v} = \begin{pmatrix} \mathbf{0}_N \\ \mathbf{u} \\ \mathbf{0}_M \end{pmatrix},$$

where $\mathbf{0}_m$ denotes the null vector in $\mathbf{C}^m$. Moreover from Lemma 1, part (iii), we deduce

$$\hat{\mathbf{B}}(\mathbf{A} \quad \mathbf{0})\mathbf{v} = \mathbf{0}_N; \tag{2.4}$$

$$\begin{pmatrix} \theta_{-N} & \cdots & \theta_0 & \cdots & \theta_M & & \\ & \ddots & & \ddots & & \ddots & \\ & & \theta_{-N} & \cdots & \theta_0 & \cdots & \theta_M \end{pmatrix} \mathbf{v} = \mathbf{u}; \tag{2.5}$$

$$\hat{\mathbf{A}}(\mathbf{0} \quad \mathbf{B})\mathbf{v} = \mathbf{0}_M. \tag{2.6}$$

Now the matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{A}}$ are nonsingular; therefore, from (2.4) and (2.6) it follows that

$$\sum_{\mu=0}^{M} a_\mu v_{\mu+i} = 0, \qquad -N \leqq i \leqq -1 \tag{2.7}$$

and

$$\sum_{\nu=0}^{N} b_\nu v_{\nu+i} = 0, \qquad n \leqq i \leqq n+M-1. \tag{2.8}$$

Relation (2.7) yields initial conditions for the solution of the difference equation (2.3), while (2.8) yields terminal conditions.

Conversely, if $\mathbf{v}$ is a nonzero solution of the problem given by (2.3), (2.7), and (2.8), then it is easy to check that the vector $\mathbf{u} = \mathbf{H}^{(2)}\mathbf{v}$ belongs to the kernel of the matrix $\mathbf{R}_n$.

3. Solving the difference problem. It is well known that the general solution of the homogeneous linear difference equation with constant coefficient (2.3) can be expressed as a linear combination of elementary solutions given in terms of the roots of the associate algebraic equation $z^N C(z) = 0$ (see for instance [8]). Imposing the boundary conditions (2.7), (2.8) to the general solution expressed in this form leads directly to the following result proved in [12].

THEOREM 1. *Let $\mathbf{R}_n$ be a rational Toeplitz matrix satisfying the condition* (2.1) *and assume $c_M \neq 0$. If $z_1, \cdots, z_L$ are the distinct zeros of the polynomial $z^N C(z)$ and $\sigma_1, \cdots, \sigma_L$ their multiplicities, then the equation* (2.3) *has a nonzero solution satisfying* (2.7) *and* (2.8) *if and only if the matrix $\mathbf{\Omega}_n = (\omega_{i,h})_{i,h=0}^{k-1}$ is singular, where*

$$\omega_{i,h} = \begin{cases} \dfrac{d^\nu}{dz^\nu}(z^i A(z))|_{z=z_j} & \text{if} \quad 0 \leqq i \leqq N-1; \\ \dfrac{d^\nu}{dz^\nu}(z^{n+i} B(z^{-1}))|_{z=z_j} & \text{if} \quad N \leqq i \leqq k-1; \end{cases}$$

and $\nu = \nu(h, j)$ is defined by

$$\nu = h - \sum_{l=1}^{j-1} \sigma_l, \qquad 0 \leqq \nu \leqq \sigma_j - 1.$$

By expressing the solution of (2.3) in terms of the companion matrix associated to the polynomial $z^N C(z)$, it is possible to extend a result that has been proved in [3] in the case of banded Toeplitz matrices to rational Toeplitz matrices. We have, in fact, the following result.

THEOREM 2. *Let* $\mathbf{R}_n$ *be a rational matrix satisfying the condition* (2.1) *and assume* $c_M \neq 0$. *Then the equation* (2.3) *has a nonzero solution satisfying* (2.7) *and* (2.8) *if and only if the matrix*

$$\tilde{\Omega}_n = \begin{pmatrix} \mathbf{A} \\ \mathbf{B}\mathbf{F}^n \end{pmatrix}$$

is singular, where $\mathbf{A}$ *and* $\mathbf{B}$ *are the matrices defined in* (2.2) *and*

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ \beta_{-N} & \cdots & \cdots & \beta_{M-1} \end{pmatrix}, \qquad \beta_j = \frac{-c_j}{c_M}$$

is the companion matrix associated to the polynomial $z^N C(z)$.

Proof. Any solution $\mathbf{v} \neq \mathbf{0}$ of (2.3) is such that

$$\begin{pmatrix} v_{-N+i+1} \\ \vdots \\ v_{M+i} \end{pmatrix} = \mathbf{F} \begin{pmatrix} v_{-N+i} \\ \vdots \\ v_{M+i-1} \end{pmatrix}, \qquad i = 0, 1, \cdots, n-1; \tag{3.1}$$

therefore we have

$$\begin{pmatrix} v_{-N+i} \\ \vdots \\ v_{M+i-1} \end{pmatrix} = \mathbf{F}^i \mathbf{w}, \qquad \mathbf{w} = \begin{pmatrix} v_{-N} \\ \vdots \\ v_{M-1} \end{pmatrix};$$

moreover, since $\mathbf{F}$ is nonsingular, we have that $\mathbf{v} \neq \mathbf{0}$ if and only if $\mathbf{w} \neq \mathbf{0}$. Setting $i = n$, (2.7) and (2.8) take the form $\mathbf{A}\mathbf{w} = \mathbf{0}$, $\mathbf{B}\mathbf{F}^n\mathbf{w} = \mathbf{0}$, $\mathbf{w} \neq \mathbf{0}$, whence

$$\det \begin{pmatrix} \mathbf{A} \\ \mathbf{B}\mathbf{F}^n \end{pmatrix} = 0.$$

Reversely, if $\tilde{\Omega}_n$ is singular there exists a vector $\mathbf{w} \in \mathbf{C}^k$, $\mathbf{w} \neq \mathbf{0}$, such that $\tilde{\Omega}_n \mathbf{w} = \mathbf{0}$, that is $\mathbf{A}\mathbf{w} = \mathbf{0}$, and that $\mathbf{B}\mathbf{F}^n\mathbf{w} = \mathbf{0}$. Therefore the vector $\mathbf{v} \in \mathbf{C}^{n+k}$, recursively defined by (3.1), is nonzero and satisfies (2.3), (2.7), and (2.8). □

A third approach to solve the linear difference equation (2.3) is cyclic reduction. This method, devised for solving certain block tridiagonal systems, has been used in [3] to compute the determinant of a banded Toeplitz matrix.

Assume, for simplicity, that $n = m\rho$, where $\rho = \max(N, M)$, $m = 2^h - 1$, and h is a positive integer, so that the linear difference equation (2.1) with the boundary conditions (2.7), (2.8) can be rewritten as a three term matrix difference equation

$$\tilde{\mathbf{C}}\mathbf{v} = \mathbf{0}, \qquad \mathbf{v} \neq \mathbf{0} \tag{3.2a}$$

$$(\mathbf{A}_0 \quad \mathbf{A}_1) \begin{pmatrix} \mathbf{v}_{-1} \\ \mathbf{v}_0 \end{pmatrix} = \mathbf{0} \tag{3.2b}$$

$$(\mathbf{B}_1 \quad \mathbf{B}_0) \begin{pmatrix} \mathbf{v}_m \\ \mathbf{v}_{m+1} \end{pmatrix} = \mathbf{0} \tag{3.2c}$$

where

$$\tilde{\mathbf{C}} = \tilde{\mathbf{C}}^{(0)} = \begin{pmatrix} \mathbf{C}_{-1} & \mathbf{C}_0 & \mathbf{C}_1 & & & & \\ & \mathbf{C}_{-1} & \mathbf{C}_0 & \mathbf{C}_1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \mathbf{C}_1 & \\ & & & & \mathbf{C}_{-1} & \mathbf{C}_0 & \mathbf{C}_1 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} \mathbf{v}_{-1} \\ \mathbf{v}_0 \\ \vdots \\ \mathbf{v}_m \\ \mathbf{v}_{m+1} \end{pmatrix},$$

$\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i \in \mathbf{C}^{\rho \times \rho}$, $\mathbf{v}_i \in \mathbf{C}^{\rho}$, and we suppose that $\det \mathbf{C}_0 \neq 0$.

At the first step of cyclic reduction, interchanging block-rows and block-columns of the matrix $\tilde{\mathbf{C}}$ according to the permutations $(1, 3, 5, \cdots, 2^h - 1, 2, 4, \cdots, 2^h - 2)$, $(1, 2, 4, \cdots, 2^h, 3, 5, \cdots, 2^h + 1)$, respectively, we get the equation

$$\begin{pmatrix} \mathbf{C}_{-1} & \mathbf{C}_0 & & & & \mathbf{C}_1 & & & \\ & & \mathbf{C}_0 & & & \mathbf{C}_{-1} & \ddots & & \\ & & & \ddots & & & \ddots & \ddots & \\ & & & & \mathbf{C}_0 & & & \mathbf{C}_{-1} & \mathbf{C}_1 \\ & \mathbf{C}_{-1} & \mathbf{C}_1 & & & \mathbf{C}_0 & & & \\ & & \ddots & \ddots & & & \ddots & & \\ & & & \ddots & \mathbf{C}_1 & & & \mathbf{C}_0 & \\ & & & & \mathbf{C}_{-1} & & & \mathbf{C}_0 & \mathbf{C}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{-1} \\ \mathbf{v}_0 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{m-1} \\ \mathbf{v}_1 \\ \mathbf{v}_3 \\ \vdots \\ \mathbf{v}_m \\ \mathbf{v}_{m+1} \end{pmatrix} = \mathbf{0}. \tag{3.3}$$

Eliminating the $(m + 1)/2$ unknowns $\mathbf{v}_0, \mathbf{v}_2, \cdots, \mathbf{v}_{m-1}$ in the last $(m + 1)/2$ equations of the above system and applying one step of block Gaussian elimination, we obtain

$$\begin{pmatrix} \tilde{\mathbf{C}}_{-1}^{(1)} & \tilde{\mathbf{C}}_0^{(1)} & \tilde{\mathbf{C}}_1^{(1)} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \tilde{\mathbf{C}}_1^{(1)} & \\ & & & \tilde{\mathbf{C}}_{-1}^{(1)} & \tilde{\mathbf{C}}_0^{(1)} & \mathbf{C}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{-1} \\ \mathbf{v}_1 \\ \mathbf{v}_3 \\ \vdots \\ \mathbf{v}_m \\ \mathbf{v}_{m+1} \end{pmatrix} = \mathbf{0}, \tag{3.4}$$

where

$$\tilde{\mathbf{C}}_{-1}^{(1)} = -\mathbf{C}_{-1}\mathbf{C}_0^{-1}\mathbf{C}_{-1},$$

$$\tilde{\mathbf{C}}_0^{(1)} = \mathbf{C}_0 - \mathbf{C}_{-1}\mathbf{C}_0^{-1}\mathbf{C}_1 - \mathbf{C}_1\mathbf{C}_0^{-1}\mathbf{C}_{-1},$$

$$\tilde{\mathbf{C}}_1^{(1)} = -\mathbf{C}_1\mathbf{C}_0^{-1}\mathbf{C}_1.$$

It is important to point out that equation (3.4) is still a three term matrix difference equation as (3.2a), but its size has been reduced by a factor of 2. Moreover, from the first $(m + 1)/2$ equations of the system (3.3) we can deduce that $\mathbf{v} \neq \mathbf{0}$ if and only if $\mathbf{v}_i \neq \mathbf{0}$, $i = -1, 1, 3, \cdots, m, m + 1$, provided that $\det \mathbf{C}_0 \neq 0$.

In order to get a new boundary condition that does not involve the vector $\mathbf{v}_0$, as (3.2b), we can replace $\mathbf{v}_0$ in (3.2b) by the expression $\mathbf{v}_0 = -\mathbf{C}_0^{-1}(\mathbf{C}_{-1}\mathbf{v}_{-1} + \mathbf{C}_1\mathbf{v}_1)$ obtained from the first block-row of (3.3). We get the new initial condition

$$\tilde{\mathbf{A}}_0^{(1)}\mathbf{v}_{-1} + \tilde{\mathbf{A}}_1^{(1)}\mathbf{v}_1 = \mathbf{0}, \tag{3.5}$$

where

$$\tilde{\mathbf{A}}_0^{(1)} = \mathbf{A}_0 - \mathbf{A}_1\mathbf{C}_0^{-1}\mathbf{C}_{-1},$$

$$\tilde{\mathbf{A}}_1^{(1)} = -\mathbf{A}_1\mathbf{C}_0^{-1}\mathbf{C}_1.$$

Thus we can apply again a cyclic reduction step to the problem consisting of (3.4), (3.5), and (3.2c); in general, at the ith step of cyclic reduction we get

$$\begin{aligned}
\tilde{\mathbf{C}}_{-1}^{(i+1)} &= -\tilde{\mathbf{C}}_{-1}^{(i)}\tilde{\mathbf{C}}_{0}^{(i)-1}\tilde{\mathbf{C}}_{-1}^{(i)},\\
\tilde{\mathbf{C}}_{0}^{(i+1)} &= \tilde{\mathbf{C}}_{0}^{(i)} - \tilde{\mathbf{C}}_{-1}^{(i)}\tilde{\mathbf{C}}_{0}^{(i)-1}\tilde{\mathbf{C}}_{1}^{(i)} - \tilde{\mathbf{C}}_{1}^{(i)}\tilde{\mathbf{C}}_{0}^{(i)-1}\tilde{\mathbf{C}}_{-1}^{(i)},\\
\tilde{\mathbf{C}}_{1}^{(i+1)} &= -\tilde{\mathbf{C}}_{1}^{(i)}\tilde{\mathbf{C}}_{0}^{(i)-1}\tilde{\mathbf{C}}_{1}^{(i)},
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
\tilde{\mathbf{A}}_{0}^{(i+1)} &= \tilde{\mathbf{A}}_{0}^{(i)} - \tilde{\mathbf{A}}_{1}^{(i)}\tilde{\mathbf{C}}_{0}^{(i)-1}\tilde{\mathbf{C}}_{-1}^{(i)},\\
\tilde{\mathbf{A}}_{1}^{(i+1)} &= -\tilde{\mathbf{A}}_{1}^{(i)}\tilde{\mathbf{C}}_{0}^{(i)-1}\tilde{\mathbf{C}}_{1}^{(i)}.
\end{aligned} \tag{3.7}$$

After $h-1$ steps of this process (3.4) and (3.5) reduce to

$$\begin{pmatrix} \tilde{\mathbf{C}}_{-1}^{(h-1)} & \tilde{\mathbf{C}}_{0}^{(h-1)} & \tilde{\mathbf{C}}_{1}^{(h-1)} & \\ & \tilde{\mathbf{C}}_{-1}^{(h-1)} & \tilde{\mathbf{C}}_{0}^{(h-1)} & \mathbf{C}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{-1} \\ \mathbf{v}_{(m+1)/2} \\ \mathbf{v}_m \\ \mathbf{v}_{m+1} \end{pmatrix} = \mathbf{0},$$

$$\tilde{\mathbf{A}}_{0}^{(h-1)}\mathbf{v}_{-1} + \tilde{\mathbf{A}}_{1}^{(h-1)}\mathbf{v}_{(m+1)/2} = \mathbf{0},$$

$$\mathbf{B}_1\mathbf{v}_m + \mathbf{B}_0\mathbf{v}_{m+1} = \mathbf{0}.$$

Thus we obtain the following result.

THEOREM 3. *Under the hypothesis* (2.1), *if the matrices* $\tilde{\mathbf{C}}_0^{(i)}$, $i = 1, \cdots, h-1$, *generated by* (3.6) *and by* $\tilde{\mathbf{C}}_j^{(0)} = \mathbf{C}_j$, *are nonsingular, then the equation* (2.1) *has a nonzero solution satisfying* (2.7) *and* (2.8) *if and only if the matrix*

$$\hat{\Omega}_n = \begin{pmatrix} \tilde{\mathbf{A}}_{0}^{(h-1)} & \tilde{\mathbf{A}}_{1}^{(h-1)} & & \\ \tilde{\mathbf{C}}_{-1}^{(h-1)} & \tilde{\mathbf{C}}_{0}^{(h-1)} & \tilde{\mathbf{C}}_{1}^{(h-1)} & \\ & \tilde{\mathbf{C}}_{-1}^{(h-1)} & \tilde{\mathbf{C}}_{0}^{(h-1)} & \mathbf{C}_1 \\ & & \mathbf{B}_1 & \mathbf{B}_0 \end{pmatrix} \in \mathbf{C}^{4p\times 4p}$$

is singular, where $\tilde{\mathbf{A}}_0^{(h-1)}$, $\tilde{\mathbf{A}}_1^{(h-1)}$, $\tilde{\mathbf{C}}_{-1}^{(h-1)}$, $\tilde{\mathbf{C}}_0$, $\tilde{\mathbf{C}}_1^{(h-1)}$ *are defined recursively by* (3.6), (3.7) *and by* $\tilde{\mathbf{C}}_j^{(0)} = \mathbf{C}_j$, $\tilde{\mathbf{A}}_j^{(0)} = \mathbf{A}_j$.

4. Computational methods for the generalized eigenvalue problem. We can apply the results of the theorems in § 3 to the generalized eigenvalue problem (1.2), stating three different equivalent conditions for the existence of $\lambda \in \mathbf{C}$ that solves (1.2). In fact observe that, denoting Δ_n, $\tilde{\Delta}_n$, and $\hat{\Delta}_n$, the determinants of the matrices Ω_n, $\tilde{\Omega}_n$, and $\hat{\Omega}_n$ defined in Theorems 1, 2, and 3, respectively, we get three scalar functions of λ having the same zeros as the polynomial $p_n(\lambda) = \det(\mathbf{T}_n - \lambda\mathbf{S}_n)$. Moreover, for any given λ, the entries of the matrices Ω_n, $\tilde{\Omega}_n$, and $\hat{\Omega}_n$ can be computed with a low computational cost, and the evaluation of their determinants has a cost independent of n. This leads us to compute the generalized eigenvalues of (2.3) by applying to the functions Δ_n, $\tilde{\Delta}_n$, and $\hat{\Delta}_n$, any root-finding method that uses only values of the function (such as the secant method, false position, etc.). Even though this is, in general, not the method recommended for numerical computations of the eigenvalues, in this case the computational saving can make it competitive.

It is interesting to point out that $\tilde{\Delta}_n$ and $\hat{\Delta}_n$ are rational functions of λ that can be computed by a rational algorithm, that is an algorithm that outputs the exact result in a finite number of arithmetic operations. Moreover Δ_n and $\tilde{\Delta}_n$ can be used to give an explicit expression of $p_n(\lambda)$. The case of Δ_n has been proved in [12] for the standard eigenvalue problem; the case of $\tilde{\Delta}_n$ is dealt with in the next section.

We observe that higher-order convergence can be obtained if Newton's method is applied for approximating the roots of the equations $\Delta_n = 0$, $\tilde{\Delta}_n = 0$ and $\hat{\Delta}_n = 0$. Any iteration of Newton's method applied to a general equation $f(\lambda) = 0$, requires the computation of the ratio $f(\lambda)/f'(\lambda)$, where $f'(\lambda)$ is the first derivative of $f(\lambda)$. So we need computable formulae for this ratio in the case where $f(\lambda)$ is any one of the functions Δ_n, $\tilde{\Delta}_n$, and $\hat{\Delta}_n$.

In this section we describe three algorithms for the computation of Δ_n, $\tilde{\Delta}_n$, and $\hat{\Delta}_n$, based on the theorems of § 3, and we analyze their computational cost. Then we extend these algorithms in such a way that Newton's method can be applied with about double the computational cost.

We observe that the structure of the matrices Ω_n, $\tilde{\Omega}_n$, and $\hat{\Omega}_n$ seems hardly exploitable to compute their determinants with low computational cost. Hence Gaussian elimination seems to be the most effective procedure for this task, so we assume that the cost of this computation is given by $O(k^3)$ arithmetic operations.

The cost of computing the entries of Ω_n grows logarithmically with n. In fact the cost for finding the zeros $z_1, z_2, \cdots, z_L$ of the polynomial $z^N C(z)$, of degree k, is independent of n; while powering each z_j, for $j = 1, \cdots, L$, for the computation of the Nth row of Ω_n requires $O(\log n)$ multiplications if the repeated squaring technique is used.

We can divide the computation of $\Delta_n(\lambda)$ at a point λ, into the following steps.

(1) Compute the zeros $z_1, z_2, \cdots, z_L$ of the polynomial $z^N C(z)$ in (1.4) together with their multiplicities $\sigma_1, \sigma_2, \cdots, \sigma_L$.

(2) Compute the rows $0, 1, \cdots, N-1$ of the matrix Ω_n.

(3) Compute $z_1^n, z_2^n, \cdots, z_L^n$ with the method of repeated squaring.

(4) Compute the rows $N, N+1, \cdots, k-1$ of the matrix Ω_n.

(5) Use Gaussian elimination to compute Δ_n.

The cost of the above algorithm is at most $O(k^3) + O(k \log n) + O(k^2 \log k \log d)$ operations, where $O(k^2 \log k \log d)$ is the number of arithmetic operations sufficient to approximate all the zeros of a k-degree polynomial with the precision of d binary digits (see [7]).

Concerning the evaluation of $\tilde{\Delta}_n(\lambda)$, we observe that also in this case the entries of $\tilde{\Omega}_n(\lambda)$ can be computed with a cost growing logarithmically with n. In fact we may use the same technique as in [3], computing the coefficients of the polynomial $\psi(z) = z^n \bmod \beta(z)$, where $\beta(z) = z^k - \sum_{j=-N}^{M-1} \beta_j z^{j+N}$ (compare Theorem 2), by means of repeated squaring modulo $\beta(z)$. Then it is sufficient to compute $\psi(\mathbf{F}) = \mathbf{F}^n$ (observe that $\beta(\mathbf{F}) = 0$) with a cost independent of n. We have, in fact, the following algorithm for the evaluation of $\tilde{\Delta}_n(\lambda)$ at a given point λ, where, for simplicity, we suppose that $n = 2^h$, h positive integer.

(1) Compute $\psi(z) = z^n \bmod \beta(z)$ as follows

$$\psi_0(z) = z,$$

$$\psi_{i+1}(z) = \psi_i(z)^2 \bmod \beta(z), i = 0, 1, \cdots, h-1.$$

$$\psi(z) = \psi_h(z).$$

(2) Compute $\psi(\mathbf{F}) = \mathbf{F}^n$ and therefore $\tilde{\Omega}_n(\lambda)$, by means of Horner's rule.

(3) Compute $\tilde{\Delta}_n(\lambda) = \det \tilde{\Omega}_n(\lambda)$ by means of Gaussian elimination.

Stage 1 of this algorithm costs $O(k^2 \log n)$ if each step of polynomial squaring and polynomial division is performed with customary algorithms. If the ground field supports FFT, then the cost of polynomial arithmetic can be reduced to $O(k \log k)$ operations

[4], so that stage 1 costs $O(k \log k \log n)$. Stages 2 and 3 have a cost independent of n that is dominated by $O(k^3)$ operations needed in Gaussian elimination. The overall cost of this algorithm is $O(k \log k \log n + k^3)$ for any field supporting FFT and $O(k^2 \log n + k^3)$, otherwise.

From the proof of Theorem 3 we obtain a third algorithm for the evaluation of $\tilde{\Delta}_n(\lambda)$. Assume $n = \rho m$, $m = 2^h - 1$, set

$$\mathbf{C}_{-1} = \begin{pmatrix} c_{-\rho} & \cdots & c_{-1} \\ & \ddots & \vdots \\ & & c_{-\rho} \end{pmatrix}, \qquad \mathbf{C}_0 = \begin{pmatrix} c_0 & \cdots & c_{\rho-1} \\ \vdots & \ddots & \vdots \\ c_{-\rho+1} & \cdots & c_0 \end{pmatrix},$$

$$\mathbf{C}_1 = \begin{pmatrix} c_\rho & \ddots & \\ \vdots & & \\ c_1 & \cdots & c_\rho \end{pmatrix},$$

$$\mathbf{A}_0 = \begin{pmatrix} a_0 & \cdots & a_{\rho-1} \\ & \ddots & \vdots \\ & & a_0 \end{pmatrix}, \qquad \mathbf{B}_0 = \begin{pmatrix} b_0 & & \\ \vdots & \ddots & \\ b_{\rho-1} & \cdots & b_0 \end{pmatrix},$$

$$\mathbf{A}_1 = \begin{pmatrix} a_\rho & \ddots & \\ \vdots & & \\ a_1 & \cdots & a_\rho \end{pmatrix}, \qquad \mathbf{B}_1 = \begin{pmatrix} b_\rho & \cdots & b_1 \\ & \ddots & \vdots \\ & & b_\rho \end{pmatrix},$$

and perform the following stages.

(1) For $i = 1$ to $h - 1$ compute

$$\tilde{\mathbf{C}}_{-1}^{(i+1)} = -\tilde{\mathbf{C}}_{-1}^{(i)} \tilde{\mathbf{C}}_0^{(i)-1} \tilde{\mathbf{C}}_{-1}^{(i)},$$

$$\tilde{\mathbf{C}}_{-0}^{(i+1)} = \tilde{\mathbf{C}}_0^{(i)} - \tilde{\mathbf{C}}_{-1}^{(i)} \tilde{\mathbf{C}}_0^{(i)-1} \tilde{\mathbf{C}}_1^{(i)} - \tilde{\mathbf{C}}_1^{(i)} \tilde{\mathbf{C}}_0^{(i)-1} \tilde{\mathbf{C}}_{-1}^{(i)},$$

$$\tilde{\mathbf{C}}_1^{(i+1)} = -\tilde{\mathbf{C}}_1^{(i)} \tilde{\mathbf{C}}_0^{(i)-1} \tilde{\mathbf{C}}_1^{(i)};$$

$$\tilde{\mathbf{A}}_0^{(i+1)} = \tilde{\mathbf{A}}_0^{(i)} - \tilde{\mathbf{A}}_1^{(i)} \tilde{\mathbf{C}}_0^{(i)-1} \tilde{\mathbf{C}}_{-1}^{(i)},$$

$$\tilde{\mathbf{A}}_1^{(i+1)} = -\tilde{\mathbf{A}}_1^{(i)} \tilde{\mathbf{C}}_0^{(i)-1} \tilde{\mathbf{C}}_1^{(i)};$$

where $\tilde{\mathbf{C}}_j^{(0)} = \mathbf{C}_j$, $\tilde{\mathbf{A}}_j^{(0)} = \mathbf{A}_j$.

(2) Compute the determinant $\hat{\Delta}_n(\lambda)$ of the matrix

$$\tilde{\Omega}_n = \begin{pmatrix} \tilde{\mathbf{A}}_0^{(h-1)} & \tilde{\mathbf{A}}_1^{(h-1)} & & \\ \tilde{\mathbf{C}}_{-1}^{(h-1)} & \tilde{\mathbf{C}}_0^{(h-1)} & \tilde{\mathbf{C}}_1^{(h-1)} & \\ & \tilde{\mathbf{C}}_{-1}^{(h-1)} & \tilde{\mathbf{C}}_0^{(h-1)} & \mathbf{C}_1 \\ & & \mathbf{B}_1 & \mathbf{B}_0 \end{pmatrix}.$$

The cost of this algorithm is $O(k^3 \log n)$ operations.

Concerning Newton's method we have to compute the ratio $f(\lambda)/f'(\lambda)$, where $f = \det \mathbf{X}(\lambda)$ and $\mathbf{X}(\lambda)$ is one of the matrices Ω_n, $\tilde{\Omega}_n$, and $\hat{\Omega}_n$. In this case we make use of the following identity

$$\frac{f'(\lambda)}{f(\lambda)} = \operatorname{trace}\left(\mathbf{X}(\lambda)^{-1} \mathbf{X}'(\lambda)\right)$$

where $\mathbf{X}'(\lambda)$ is the componentwise derivative of $\mathbf{X}(\lambda)$.

Case 1. $\mathbf{X}(\lambda) = \Omega_n(\lambda)$. The matrix $\Omega_n(\lambda)$ has a first derivative if the zeros $z_1, \cdots, z_k$ of the polynomial $z^N C(z)$ are pairwise distinct. Moreover $\Omega_n'(\lambda)$ can be evaluated by

computing the values z_i' through the following linear system of equations.

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ s_{1,1} & s_{1,2} & \cdots & s_{1,k} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,k} \\ \vdots & \vdots & & \vdots \\ s_{k-1,1} & s_{k-1,2} & \cdots & s_{k-1,k} \end{pmatrix} \begin{pmatrix} z_1' \\ z_2' \\ z_3' \\ \vdots \\ z_k' \end{pmatrix} = \begin{pmatrix} \gamma_1' \\ \gamma_2' \\ \gamma_3' \\ \vdots \\ \gamma_k' \end{pmatrix}, \tag{4.1}$$

where $\gamma_i(\lambda) = (-1)^i c_{M-j}(\lambda)/c_M(\lambda)$ and $s_{i,j}$ is the symmetric function of degree i in the variables $z_1, \cdots, z_{j-1}, z_{j+1}, \cdots, z_k$. This system is obtained by taking the first derivative, with respect to λ, of the relations $s_i(z_1, \cdots, z_k) = \gamma_i$, where $s_i(z_1, \cdots, z_k)$ is the symmetric function of degree i in the variables $z_1, \cdots, z_k$. The cost of computing the symmetric functions in (4.1) and of solving the linear system (4.1) amounts to $O(k^2)$ operations. So the asymptotic overall cost for computing $\Delta_n'(\lambda)/\Delta_n(\lambda) = \text{trace}\,(\Omega_n^{-1}(\lambda)\Omega_n'(\lambda))$ is still $O(k \log n + k^3)$.

Observe that if for a given λ the zeros $z_1, \cdots, z_k$ are not pairwise distinct, the derivatives $z_1', \cdots, z_k'$ may not exist at λ.

Case 2. $\mathbf{X}(\lambda) = \tilde{\mathbf{\Omega}}_n(\lambda)$. It suffices to compute $\frac{d}{d\lambda}(\mathbf{F}^n)$, since the matrices $\mathbf{A}$ and $\mathbf{B}$ are constant with respect to λ. This computation can be performed by means of a suitable modification of the second algorithm given in this section, obtaining at the end the matrices $\mathbf{F}^n$ and $\frac{d}{d\lambda}(\mathbf{F}^n)$. A complete description of this modification is given in [3] in the case of banded Toeplitz matrices. The overall cost is still $O(k \log k \log n + k^3)$ for any field supporting FFT.

Case 3. $\mathbf{X}(\lambda) = \hat{\mathbf{\Omega}}_n(\lambda)$. In order to compute $\hat{\mathbf{\Omega}}_n'(\lambda)$, it is sufficient to take the first derivatives in the relations at step 1 of the third algorithm of this section recalling that $(\mathbf{Y}^{-1})' = -\mathbf{Y}^{-1}\mathbf{Y}'\mathbf{Y}^{-1}$ for any nonsingular matrix $\mathbf{Y}$. The overall asymptotic cost is $O(k^3 \log n)$.

We observe that, even though the cost of one iteration of Newton's method is roughly doubled in all the three cases, this method is preferable for approximating simple eigenvalues, due to its quadratic convergence.

5. Explicit formulae for the characteristic polynomial. In the case of the standard eigenvalue problem where $\mathbf{S}_n$ is the identity matrix, Theorems 1 and 2 of § 3 allow us to give explicit formulae for the characteristic polynomial $p_n(\lambda) = \det(\mathbf{T}_n - \lambda\mathbf{I}_n)$ of a rational matrix. In [12] the following result is proved:

$$\det(\lambda\mathbf{I}_n - \mathbf{T}_n) = (-1)^{n(M-1)}(a_0 b_0)^{-n}\frac{1}{W}(c_M - \lambda\theta_M)^n \frac{\Delta_n(\lambda)}{V(\lambda)}, \tag{5.1}$$

where W is the determinant of the $k \times k$ matrix $\binom{\mathbf{A}}{\mathbf{B}}$, with $\mathbf{A}$ and $\mathbf{B}$ defined in (2.2), and $V(\lambda)$ is the generalized Vandermonde determinant of $z_1, \cdots, z_k$. In this section we give an analogous result involving the determinant $\tilde{\Delta}_n(\lambda)$. The proof of (5.1) given in [12] is based on the following facts. The function $(c_M - \lambda\theta_M)^n(\Delta_n(\lambda)/V(\lambda))$ is a polynomial in λ of degree n, and its leading coefficient is $(-1)^{n(M-1)}(a_0 b_0)^n W$. If the eigenvalues $\lambda_1, \cdots, \lambda_n$, of $\mathbf{T}_n$ are pairwise different, the relation (5.1) holds since in the view of Theorem 1 the left-hand and the right-hand sides of (5.1) are both monic polynomials of degree n having the same zeroes. The generalization to the case where the eigenvalues are not distinct is obtained by a continuity argument.

We can follow the same technique to prove the following formula involving $\tilde{\Delta}_n(\lambda)$:

$$\det(\lambda\mathbf{I}_n - \mathbf{T}_n) = (-1)^{n(M-1)}(a_0 b_0)^{-n}\frac{1}{W}(c_M - \lambda\theta_M)^n \tilde{\Delta}_n(\lambda). \tag{5.2}$$

We outline the proof of this result. It is sufficient to prove that $(c_M - \lambda\theta_M)^n\tilde{\Delta}_n(\lambda)$ is a polynomial of degree n having leading coefficient $(-1)^{n(M-1)}(a_0b_0)^nW$. To this purpose consider a $k \times k$ matrix of the type $\begin{pmatrix}\mathbf{A}\\ \mathbf{XF}^m\end{pmatrix}$, where $\mathbf{X} \in \mathbf{C}^{M\times k}$, and show that its determinant is a polynomial of degree m; the proof is completed by setting $\mathbf{X} = \mathbf{B}$ and $m = n$.

Let $\mathbf{x}_1^T, \cdots, \mathbf{x}_M^T$ be the rows of the matrix $\mathbf{XH}$ where

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix}$$

is the shift matrix of order k, and let $\tau = (\tau_1, \cdots, \tau_M)^T$ be the kth column of $\mathbf{X}$. First we want to relate $\det\begin{pmatrix}\mathbf{A}\\ \mathbf{XF}^m\end{pmatrix}$ and $\det\begin{pmatrix}\mathbf{A}\\ \mathbf{XF}^{m+1}\end{pmatrix}$. Since $\mathbf{F} = \mathbf{H} + \mathbf{e}^{(k)}\beta^T$, where $\beta^T = (\lambda\theta^T - \mathbf{c}^T)/\Lambda$, $\theta^T = (\theta_{-N}\cdots\theta_{M-1})$, $\mathbf{c}^T = (c_{-N}\cdots c_{M-1})$, $\Lambda = c_M - \lambda\theta_M$, and $\mathbf{e}^{(k)}$ is the kth column of the $k \times k$ identity matrix, we have

$$\det\begin{pmatrix}\mathbf{A}\\ \mathbf{XF}^{m+1}\end{pmatrix} = \det\begin{pmatrix}\mathbf{A}\\ \mathbf{XHF}^m\end{pmatrix} + \frac{1}{\Lambda}\sum_{i=1}^{M}\tau_i\left[\lambda\det\begin{pmatrix}\mathbf{A}\\ \mathbf{X}_\theta^{(i)}\mathbf{F}^m\end{pmatrix} - \det\begin{pmatrix}\mathbf{A}\\ \mathbf{X}_c^{(i)}\mathbf{F}^m\end{pmatrix}\right], \tag{5.3}$$

where $\mathbf{X}_\theta^{(i)} = (\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}, \theta, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_M)^T$ and $\mathbf{X}_c^{(i)} = (\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}, \mathbf{c}, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_M)^T$. This can be proved by using the multilinearity of the function determinant M times, being

$$\mathbf{XF}^{m+1} = \mathbf{XHF}^m + \tau\beta^T\mathbf{F}^m.$$

For instance, at the first step we have

$$\det\begin{pmatrix}\mathbf{A}\\ \mathbf{XF}^{m+1}\end{pmatrix} = \det\begin{pmatrix}\mathbf{A}\\ \left[\begin{pmatrix}\mathbf{x}_1^T\\ \vdots\\ \mathbf{x}_{M-1}^T\end{pmatrix} + \begin{pmatrix}\tau_1\\ \vdots\\ \tau_{M-1}\end{pmatrix}\beta^T\right]\mathbf{F}^m\\ \mathbf{x}_M^T\mathbf{F}^m\end{pmatrix}$$

$$+ \tau_M \det\begin{pmatrix}\mathbf{A}\\ \left[\begin{pmatrix}\mathbf{x}_1^T\\ \vdots\\ \mathbf{x}_{M-1}^T\end{pmatrix} + \begin{pmatrix}\tau_1\\ \vdots\\ \tau_{M-1}\end{pmatrix}\beta^T\right]\mathbf{F}^m\\ \beta^T\mathbf{F}^m\end{pmatrix}.$$

Moreover, a suitable Gaussian elimination allows us to rewrite the second determinant as

$$\frac{1}{\Lambda}\det\begin{pmatrix}\mathbf{A}\\ \begin{pmatrix}\mathbf{x}_1^T\\ \vdots\\ \mathbf{x}_{M-1}^T\end{pmatrix}\mathbf{F}^m\\ (\lambda\theta^T - \mathbf{c}^T)\mathbf{F}^m\end{pmatrix}.$$

Another application of the multilinearity of the determinant to the last row yields the Mth term of the sum in (5.3).

Relation (5.3) can be easily used to prove by induction on m that for every $\mathbf{X} \in \mathbf{C}^{M\times k}$ (which is constant with respect to λ) one has

$$\det\begin{pmatrix}\mathbf{A}\\ \mathbf{X}\mathbf{F}^m\end{pmatrix}=\frac{\alpha_m(\mathbf{X})\lambda^m+Q_{m-1}(\mathbf{X},\lambda)}{\Lambda^m},$$

where $Q_{m-1}(\mathbf{X}, \lambda)$ is a polynomial in λ of degree at most $m-1$ and $\alpha_m(\mathbf{X})$ is a scalar satisfying the recurrence relation $\alpha_{m+1}(\mathbf{X}) = -\theta_M\alpha_m(\mathbf{XH}) + \sum_{i=1}^{M}\tau_i\alpha_m(\mathbf{X}_\theta^{(i)})$. This relation can be used to prove by induction on m the following formula:

$$\alpha_m(\mathbf{X})=(-1)^{m(M-1)}(a_0b_0)^m\det\begin{pmatrix}\mathbf{A}\\ \mathbf{X}\end{pmatrix}. \tag{5.4}$$

This can be obtained computing the determinant of the $(k+1)\times(k+1)$ matrix $\begin{pmatrix}\tilde{\mathbf{A}}\\ {}_0\tilde{\mathbf{X}}\end{pmatrix}$, where

$$\tilde{\mathbf{A}}=\begin{pmatrix}a_0 & \cdots & a_M & & \\ & \ddots & & \ddots & \\ & & a_0 & \cdots & a_M\end{pmatrix}\in\mathbf{C}^{(N+1)\times(k+1)},$$

by means of Laplace's rule applied to the first and to the last column of the matrix. Setting $\mathbf{X} = \mathbf{B}$ and $m = n$, we obtain from (5.3) that the function $\Lambda^n\det\begin{pmatrix}\mathbf{A}\\ \mathbf{B}\mathbf{F}^n\end{pmatrix} = (c_M-\lambda\theta_M)^n\tilde{\Delta}_n(\lambda)$ is a polynomial of degree n, and from (5.4) that its leading coefficient is $(-1)^{n(M-1)}(a_0b_0)^nW$.

6. An example of degeneration. Theorems 1 and 2 in § 3 require that the leading coefficient c_M of the polynomial $C(z, \lambda)$ is nonzero. In many cases c_M is a nontrivial linear function of λ so that the condition holds for any λ except at most one value.

In some special case it may happen that c_M vanishes identically, so that Theorems 1 and 2 cannot be directly applied. However in this case it is often possible to manage the difference equation defined by (2.3), (2.7), and (2.8) to find analogous singularity conditions involving either the zeros of the polynomial $z^NC(z)$ or the companion matrix associated to (2.3).

We will examine in this section an example of degeneration occurring in a concrete situation.

In the problem of optimization of rejection filters [5], the maximization of the signal/noise ratio leads to the generalized eigenvalue problem

$$\mathbf{T}_n\mathbf{u}=\lambda\mathbf{S}_n\mathbf{u},$$

where the minimum eigenvalue must be computed. In this setting $\mathbf{T}_n$ is the interference covariance matrix, and $\mathbf{S}_n$ is the signal covariance matrix. An interesting situation occurs when $\mathbf{T}_n$ and $\mathbf{S}_n$ are hermitian Toeplitz matrices defined by $t_{i-j} = \rho_1^{|i-j|}\exp(2\pi\mathbf{i}\theta_1(i-j))$ and $s_{i-j} = \rho_2^{|i-j|}\exp(2\pi\mathbf{i}\theta_2(i-j))$, $i, j = 0, \cdots, n-1$, and $\rho_1, \rho_2, \theta_1, \theta_2$, are real numbers with

$$\rho_1\neq\rho_2, 0<\rho_1,\rho_2<1. \tag{6.1}$$

It is easy to check that $\mathbf{T}_n$ and $\mathbf{S}_n$ are both rational matrices generated by the functions

$$T(z)=\frac{\gamma(z)}{\alpha(z)\beta(z^{-1})},\qquad S(z)=\frac{\eta(z)}{\delta(z)\varepsilon(z^{-1})},$$

respectively, where

$$\gamma(z)=1-\rho_1^2, \qquad \alpha(z)=1-\xi_1 z, \qquad \beta(z)=1-\bar{\xi}_1 z,$$
$$\eta(z)=1-\rho_2^2, \qquad \delta(z)=1-\xi_2 z, \qquad \varepsilon(z)=1-\bar{\xi}_2 z,$$
$$\xi_i=\rho_i \exp(2\pi \mathbf{i}\theta_i), \qquad i=1,2.$$

The generalized eigenvalue problem is reduced to the computation of λ and $\mathbf{u}\in\mathbf{C}^n$, $\mathbf{u}\neq\mathbf{0}$, such that $\mathbf{R}_n\mathbf{u}=\mathbf{0}$, where $\mathbf{R}_n=\mathbf{T}_n-\lambda\mathbf{S}_n$ is the Toeplitz matrix generated by the rational function

$$R(z,\lambda)=\frac{C(z,\lambda)}{A(z)B(z^{-1})},$$
$$C(z,\lambda)=\bar{c}_1 z^{-1}+c_0+c_1 z,$$
$$A(z)=a_0+a_1 z+a_2 z^2,$$
$$B(z)=\bar{a}_0+\bar{a}_1 z+\bar{a}_2 z^2,$$

where

$$c_0=(1-\rho_1^2)(1+\rho_2^2)-\lambda(1-\rho_2^2)(1+\rho_1^2),$$
$$c_1=\lambda(1-\rho_2^2)\bar{\xi}_1-(1-\rho_1^2)\bar{\xi}_2,$$
$$a_0=1, \qquad a_1=-(\xi_1+\xi_2), \qquad a_2=\xi_1\xi_2.$$

With the notation of § 1 we have for this case $p=q=1$, $r=s=M=N=2$, $k=4$.

The conditions (2.1) are satisfied since $a_0=b_0=1$ and

$$A(z)=\xi_1\xi_2\left(z-\frac{1}{\xi_1}\right)\left(z-\frac{1}{\xi_2}\right), \qquad z^2B(z^{-1})=(z-\bar{\xi}_1)(z-\bar{\xi}_2).$$

Moreover, if $A(z)$ and $z^2B(z^{-1})$ had a common zero, then we would have $\bar{\xi}_i=1/\xi_j$, for some integers i,j, whence $\rho_i\rho_j=1$, which is not consistent with (6.1). Then the Greville and Trench lemma holds, and we can characterize the solution of the generalized eigenvalue problem by means of the difference equation

$$\bar{c}_1 v_{i-1}+c_0 v_i+c_1 v_{i+1}=0, \qquad 0\leqq i\leqq n-1 \tag{6.2}$$

with the boundary conditions

$$a_0 v_{-2}+a_1 v_{-1}+a_2 v_0=0, \tag{6.3a}$$
$$a_0 v_{-1}+a v_0+a_2 v_1=0, \tag{6.3b}$$
$$\bar{a}_2 v_{n-2}+\bar{a}_1 v_{n-1}+\bar{a}_0 v_n=0, \tag{6.4a}$$
$$\bar{a}_2 v_{n-1}+\bar{a}_1 v_n+\bar{a}_0 v_{n+1}=0. \tag{6.4b}$$

Observe that this problem is degenerate, since $c_M=c_2=0$ for any $\lambda\in\mathbf{C}$.

- Solution by means of the characteristic equation.

The general solution of equation (6.2) is given by $v_i=\alpha_1 z_1^{i+1}+\alpha_2 z_2^{i+1}$, $i=-1,\cdots,n$, assuming that z_1 and z_2 are the distinct roots of the characteristic equation

$$\bar{c}_1+c_0 z+c_1 z^2=0.$$

Now observe that conditions (6.3a) and (6.4b) involve the entries v_{-2} and v_{n+1}, which do not appear in the difference equation (6.2), and they can be viewed as implicit definitions of these additional components. Conditions (6.3b) and (6.4a) are the boundary conditions that can be used to determine α_1 and α_2. In fact from (6.2), (6.3b), and (6.4a) we obtain the condition,

$$\det \begin{pmatrix} A(z_1) & A(z_2) \\ z_1^{n+1}B(z_1^{-1}) & z_2^{n+1}B(z_2^{-1}) \end{pmatrix} = 0, \tag{6.5}$$

that represents the extension of Theorem 1 to this degenerate case.

A similar argument leads to the condition

$$\det \begin{pmatrix} a_0 + a_1 z_1 + a_2 z_1^2 & a_1 + 2a_2 z_1 \\ \bar{a}_0 z_1^2 + \bar{a}_1 z_1 + \bar{a}_2 & (n+1)\bar{a}_0 z_1 + n\bar{a}_1 + (n-1)\bar{a}_2 z_1^{-1} \end{pmatrix}$$

in the case where $z_1 = z_2$.

• Solution by means of powering a companion matrix.

Consider the 2×2 companion matrix associated to (6.2)

$$\mathbf{F} = \begin{pmatrix} 0 & 1 \\ -\dfrac{\bar{c}_1}{c_1} & -\dfrac{c_0}{c_1} \end{pmatrix},$$

so that the general solution of (6.2) can be written as

$$\begin{pmatrix} v_{i-1} \\ v_i \end{pmatrix} = F^i \begin{pmatrix} v_{-1} \\ v_0 \end{pmatrix}, \qquad 0 \leqq i \leqq n.$$

Imposing the boundary conditions (6.3b) and (6.4a), we obtain

$$(a_0 \quad a_1 \quad a_2) \begin{pmatrix} 1 & 0 \\ & \mathbf{F} \end{pmatrix} \begin{pmatrix} v_{-1} \\ v_0 \end{pmatrix} = 0,$$

$$(\bar{a}_2 \quad \bar{a}_1 \quad \bar{a}_0) \begin{pmatrix} 1 & 0 \\ & \mathbf{F} \end{pmatrix} \mathbf{F}^{n-1} \begin{pmatrix} v_{-1} \\ v_0 \end{pmatrix} = 0,$$

that is the condition

$$\det \begin{pmatrix} (a_0 \quad a_1 \quad a_2) \begin{pmatrix} 1 & 0 \\ & \mathbf{F} \end{pmatrix} \\ (\bar{a}_2 \quad \bar{a}_1 \quad \bar{a}_0) \begin{pmatrix} 1 & 0 \\ & \mathbf{F} \end{pmatrix} \mathbf{F}^{n-1} \end{pmatrix} = 0. \tag{6.6}$$

Observe that the matrices in (6.5) and (6.6) replace the matrices Ω_n and $\tilde{\Omega}_n$, respectively, of Theorems 1 and 2. It is interesting to point out that the matrices (6.5) and (6.6) have dimension 2 while the size k of the original problem is 4. In this case the degeneracy of the problem has brought a further reduction of the complexity.

• Solution by means of cyclic reduction.

Theorem 3 does not require the condition $c_2 \neq 0$; however, it is worth pointing out that solving equation (6.2) with conditions (6.3b) and (6.4a), by means of a cyclic reduction, does not involve block matrices as in the general case (see § 4). Even in this case, the degeneracy of the problem brings a simplification of the computational method.

REFERENCES

[1] D. BINI AND M. CAPOVANI, *Spectral and computational properties of band symmetric Toeplitz matrices*, Linear Algebra Appl., 52 (1983), pp. 99–126.

[2] ———, *Fast parallel and sequential computations and spectral properties concerning band Toeplitz matrices*, Calcolo, 20 (1983), pp. 177–189.

[3] 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.

[4] ———, *Polynomial division and its computational complexity*, J. Complexity, 2 (1986), pp. 179–203.

[5] F. CHIUPPESI, G. GALATI, AND P. LOMBARDI, *Optimisation of rejection filters*, IEEE Proc., Vol. 127, Pt. F, No. 5, October 1980, pp. 354–360.

[6] T. N. E. GREVILLE AND W. F. TRENCH, *Band matrices with Toeplitz inverses*, Linear Algebra Appl., 27 (1979), pp. 199–209.

[7] V. PAN, *Algebraic complexity of computing polynomial zeros*, Comput. Math. Appl., 14 (1987), pp. 285–304.

[8] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer-Verlag, Berlin, NY, 1980.

[9] M. TISMENETSKY, *Determinant of block-Toeplitz band matrices*, Linear Algebra Appl., 85 (1987), pp. 165–184.

[10] W. F. TRENCH, *Characteristic polynomials of symmetric rationally generated Toeplitz matrices*, Linear and Multilinear Algebra, 21 (1987), pp. 289–296.

[11] ———, *Numerical solution of the eigenvalue problem for symmetric rationally generated Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 291–303.

[12] ———, *On the eigenvalue problem for a class of band matrices including those with Toeplitz inverses*, SIAM J. Algebraic Discrete Methods, 2 (1986), pp. 167–179.

[13] ———, *On the eigenvalue problem for Toeplitz band matrices*, Linear Algebra Appl., 64 (1985), pp. 199–214.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 553–564, October 1990

006

EIGENPROBLEM ERROR BOUNDS WITH APPLICATION TO SYMMETRIC DYNAMIC SYSTEM MODIFICATION*

YITSAK M. RAM†‡, JOAB J. BLECH†, AND SIMON G. BRAUN†

Abstract. Suppose A and B are two $m \times m$ symmetric matrices. Let $C = A + B$. Some of C's lowest eigenvalues together with their corresponding invariant subspace are bound in terms of B, a subspectrum of A, and an invariant subspace of A.

An application demonstrating the usefulness of the presented theorems is given. The application chosen is related to the frequently encountered engineering problem of the influence of a structural modification on the dynamic behaviour of a structure.

Key words. modified eigenvalue problem, truncation error, structural modification, modal analysis, vibration test

AMS(MOS) subject classifications. 15A42, 65F15

1. Introduction.

Suppose A and B are two $m \times m$ symmetric matrices. Let $C = A + B$. Let $A = \Phi\Lambda\Phi^t$ be the spectral decomposition of A, where Φ is an $m \times m$ ortho normal matrix and $\Lambda = \text{diag}\{\lambda_i(A);\ i = 1, \cdots, m\}$.[1] Partition Φ and Λ in the form $\Phi = [\Phi_1\Phi_2]$ and

$$\Lambda = \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix},$$

where $\Phi_1 \in R^{m\times n}$ and $\Lambda_1 \in R^{n\times n}$. Suppose Λ_1, Φ_1, and B are given, whereas A, Φ_2, and Λ_2 are unknown. We consider here the problem of bounding some of the lowest eigenvalues of C. Error bounds on the angle between the invariant subspace of C and the subspace, which is spanned by its corresponding Ritz vectors from span (Φ_1) are also given.

This problem arises in the field of vibration analysis. The dynamic behaviour of an engineering structure is determined by the symmetric definite generalized eigenvalue problem $(K - \lambda M)x = 0$ (see, e.g, Strang [23, pp. 261–263], Weinberger [25, pp. 6–17]) where $K \in R^{m\times m}$ and $M \in R^{m\times m}$ are the stiffness and the mass matrices of the structure, respectively, and $x \in R^m$. In some applications, due to the complexity of the structure no reasonable analytical model of the stiffness matrix can be evaluated, whereas the mass matrix is known. Additional information on the dynamic behaviour of the structure is available from a vibration test, where the excitation and the response of the structure at many points are measured experimentally. Identification techniques (Chu [6], Link [15], Braun and Ram [4], [5]) extract a part of the eigenpairs of the structure from the measurements. Since the measured data form a discrete time series, an inherent limitation of the vibration test is that the identified eigenpairs are restricted by the sampling rate. Thus, a vibration test usually results in an incomplete set of eigenpairs (Berman and Flannelly [2]).

A frequently encountered engineering problem is one in which the designer would like to change the dynamic behaviour of an existing structure by means of stiff-

* Received by the editors December 5, 1988; accepted for publication (in revised form) July 13, 1989.

† Faculty of Mechanical Engineering, Technion-Israel Institute of Technology, Haifa 32000, Israel (MERSBRY@TECHNION.BITNET).

‡ Present address, Manufacturing Engineering Department, Hong Kong Polytechnic, 83 Tat chee Avenue, Kowloon, Hong Kong (MERAMITZ@CPHKVX.BITNET).

[1] Throughout this paper the eigenvalues are labeled in an increasing order, $\lambda_1 \leqq \lambda_2 \leqq \cdots \leqq \lambda_m$.

ness modifications. In this problem the following are assumed to be known: (a) the mass matrix M; (b) a subset of the eigenpairs of $(K - \lambda M)x = 0$ corresponding to the smallest eigenvalues (from a vibration test); and (c) the design incremental stiffness matrix $\Delta K \in R^{m \times m}$. The objective is then to bound a part of the eigenpairs of $((K + \Delta K) - \lambda M)x = 0$.

In § 4 we shall show that this structural modification problem is congruently equivalent to the above-mentioned problem that we wish to consider.

There are many publications concerning bounds for the eigenproblem that cover a very wide range of applications. In Stewart [20], [21], [22], Crawford [7], Kahan, Parlett, and Jiang [13], and Weinberger [25] the sensitivity of the eigenpairs to random perturbation is considered. For this purpose it is assumed that an upper bound on the norm of the perturbation matrix is given. Although we may note that in our problem B is perturbed by the (unknown) matrix A, the above-mentioned perturbation bounds are not very appropriate for our case since, in practice, the norm of A is very large. Moreover, in these approaches the information concerning the given eigenvectors of A would not be taken into account. Other publications (e.g., Davis, Kahan, and Weinberger [9], Kahan [12], Lehmann [14], and Thompson [24]) are motivated by the desire to reduce the size of the eigenproblem. In this case a large matrix is given explicitly and the objective is to bound the eigensolution by considering certain submatrices. Since A is unknown, our problem does not belong to this category, and a separate analysis is needed to solve the problem.

Arbenz and Golub [1] have developed a very efficient procedure for finding the eigenvalues and eigenvectors of $C = A + B$, where B is of low rank and the (complete) spectral decomposition of A is given. It is well known that this problem arises in many applications. However, occasionally, as in our problem, only a subspectrum of A is known. This occurs in the following cases: (a) If $\tilde{A}$ is a finite-element approximation for A, then sometimes $\tilde{A}$ differs significantly from A, but still the eigenpairs of $\tilde{A}$ corresponding to the smallest eigenvalues approximate eigenpairs of A accurately (Berman [3]). (b) If the eigenvalues and the eigenvectors of A are experimentally measured, as in the case of the structural modification, then the eigenvectors corresponding to the high frequencies are very sensitive to small uncertainty in the location of the measurement points. Also the sampling rate and the use of anti-aliasing filters restrict the possibility of extracting the largest eigenvalues.

In the case where *all* the eigenpairs of A are given and B is of low rank, it is recommended to calculate the eigenvalues of C by Arbenz and Golub's procedure. Indeed, one of the applications mentioned in their paper is the work of Simpson [18] on structural modification.

It should be noted that our development is restricted to the symmetric-definite eigenvalue problem. Although the generalized eigenvalue problem arises in many fields (see, e.g., Haley [11] for applications in the linear system analysis and control), the requirements in our problem of symmetry and definition exclude all controlled systems. But this approach may be useful in the analysis of modified passive dynamic systems.

This work is organized as follows. The bounds for the eigenvalues are derived in § 2. Bounds for the eigenvectors are given in § 3. The application concerning the structural modification problem and a detailed example demonstrating the procedures are given in § 4. Conclusions are summarized in the last paragraph.

2. Error bounds for eigenvalues. We introduce a matrix E, which is orthonormally similar to C, and partition it in the following form:

$$E \equiv \Phi^t C \Phi \equiv \begin{bmatrix} E_1 & E_2 \\ E_2^t & E_4 \end{bmatrix} = \begin{bmatrix} \Phi_1^t C \Phi_1 & \Phi_1^t C \Phi_2 \\ \Phi_2^t C \Phi_1 & \Phi_2^t C \Phi_2 \end{bmatrix}, \qquad E_1 \in \mathbf{R}^{n \times n}. \tag{2.1}$$

Using the orthogonality relation

$$\Phi_i^t A \Phi_j = \begin{cases} \Lambda_i, & i=j \\ 0, & i \neq j \end{cases}, \qquad (i=1,2; j=1,2), \tag{2.2}$$

we get

$$E = \begin{bmatrix} \Lambda_1 + \Phi_1^t B \Phi_1 & \Phi_1^t B \Phi_2 \\ \Phi_2^t B \Phi_1 & \Lambda_2 + \Phi_2^t B \Phi_2 \end{bmatrix}. \tag{2.3}$$

Note that only the leading principal submatrix E_1 is known explicitly.

A similar problem was treated by Lehmann [14] and Kahan [12]. Their results are expounded in detail in Parlett [16]. There are, however, basic differences between the present work and the one just mentioned. In their problem only E_4 is unknown, whereas in our problem only E_1 is given explicitly. Our problem contains additional information on the structure of E_2 and on the first eigenpairs of A.

Lehmann and Kahan's results enable us to get intervals that contain eigenvalues. Each interval depends on an arbitrary parameter. There is no outline on how to choose the parameters such that the intervals be minimized in length. In addition, it is impossible to determine from their work which eigenvalue lies within each interval.

The following theorem, which essentially combines the Lehmann interval and the monotonicity principle for eigenvalues, enables us, in certain circumstances, to obtain a greatest lower bound on a *specific* eigenvalue of C. A sufficient condition for the realization of those circumstances is then presented in Theorem 2.2.

THEOREM 2.1. *Suppose that $A \in R^{m\times m}$ and $B \in R^{m\times m}$ are symmetric matrices and that E is any matrix orthonormally similar to $A + B$. Let E be partitioned as in* (2.1). *Denote by S an $n \times q$ matrix that satisfies the following relationship*:

$$SS^t = E_2 E_2^t \tag{2.4}$$

where $q = \operatorname{rank}(E_2)$. *Introduce an auxiliary matrix* $Y[X(\mu)]$:

$$Y[X(\mu)] = \begin{bmatrix} E_1 & S \\ S^t & X(\mu) \end{bmatrix} \tag{2.5}$$

where for any real number $\mu \notin \operatorname{Spec}(E_1)$ the matrix $X(\mu) \in R^{q\times q}$ is defined by

$$X(\mu) = \mu I_q + S^t (E_1 - \mu I_n)^{-1} S. \tag{2.6}$$

Let α_i ($i = 1, \cdots, n-1$; $n \geqq 2$) be defined by

$$\alpha_i = \max_{j=1,\cdots,i+1} [\lambda_j(A) + \lambda_{i-j+2}(B)]. \tag{2.7}$$

If $\lambda_i(E_1) < \alpha_i$, then for any μ that satisfies

$$\lambda_i(E_1) < \mu \leqq \alpha_i, \qquad i = 1, \cdots, n-1, \tag{2.8}$$

the following inequalities hold

$$\lambda_i(Y[X(\mu)]) \leqq \lambda_i(E) = \lambda_i(A+B) \leqq \lambda_i(E_1). \tag{2.9}$$

Moreover, if the only known data are E_1, the product $E_2E_2^t$ and the subspectrum of A and B, then the largest number which guarantees a lower bound is $\lambda_i(Y[X(\alpha_i)])$.

Proof. From the monotonicity principle [16, p. 192] it follows that

$$\alpha_i \leqq \lambda_{i+1}(A+B) = \lambda_{i+1}(E), \qquad i = 1, \cdots, n-1. \tag{2.10}$$

The equality on the right-hand side of (2.10) is due to the similarity between E and $A + B$. By Cauchy's interlace theorem we have

$$\lambda_i(E) \leqq \lambda_i(E_1), \qquad i = 1, \cdots, n. \tag{2.11}$$

Since $\lambda_i(E_1) < \alpha_i$, any μ that satisfies $\lambda_i(E_1) < \mu \leqq \alpha_i$ must also satisfy $\lambda_i(E) < \mu \leqq \lambda_{i+1}(E)$.

Lehmann's Theorem [16, p. 199] states that there is at least one eigenvalue of E in the interval $[\lambda_i(Y[X(\mu)]), \mu]$. Therefore $\lambda_i(E)$ must lie within Lehmann's interval. The following inequality must, therefore, hold:

$$\lambda_i(Y[X(\mu)]) \leqq \lambda_i(E). \tag{2.12}$$

From (2.11) and (2.12) it follows that

$$\lambda_i(Y[X(\mu)]) \leqq \lambda_i(E) \leqq \lambda_i(E_1). \tag{2.13}$$

For any $\lambda_i(E_1) < \mu < \lambda_{i+1}(E_1)$ and for any unit vector $u \in R^{n+q}$ the product $u^t Y[X(\mu)]u$ is a continuous function of μ. In this case, $\lambda_i(Y[X(\mu)])$ is a monotonic nondecreasing function of μ [16, p. 199]. Hence, the largest number which assures a lower bound on $\lambda_i(E)$ is $\lambda_i(Y[X(\alpha_i)])$. □

Remark 2.1. The application of Theorem 2.1 to our problem requires the formation of the matrix S. In general the construction of S requires finding an orthonormal basis of E_2^t [10, pp. 150–153]. In our problem E_2 is not given explicitly. We next show how to construct S. Noting that $\Phi_1\Phi_1^t$ and $\Phi_2\Phi_2^t$ are orthogonal projectors onto complementary subspaces it follows that $\Phi_2\Phi_2^t = I_m - \Phi_1\Phi_1^t$, hence an explicit formula for the product $E_2E_2^t$ is given by

$$E_2E_2^t = \Phi_1^t B\Phi_2\Phi_2^t B\Phi_1 = \Phi_1^t B(I_m - \Phi_1\Phi_1^t)B\Phi_1. \tag{2.14}$$

Let $E_2E_2^t = QDQ^t$ be the spectral decomposition of $E_2E_2^t$, where $Q \in R^{n\times n}$ is an orthonormal matrix and $D = \text{DIAG}\{d_i;\ i = 1, \cdots, n\}$. Then $d_i = 0$ for $i = 1, \cdots, n - q$ and $d_i > 0$ for $i = n - q + 1, \cdots, n$. The columns of the matrix S can be taken as the last q columns of $QD^{1/2}$ (in any order). Consider now the numerical problem of determining q. Apparently, q is determined by the number of the nonzero diagonal entries of D and it is equivalent to the nontrivial problem of determining the numerical rank of $E_2E_2^t$. However, note that $q \leqq \min[n, m - n, \text{rank}(B)]$; also note that Theorem 2.1 still holds for any q that satisfies $\text{RANK}(E_2) \leqq q \leqq m - n$. Thus the problem of determining the numerical rank of E_2 in our case is not critical. □

To find the greatest lower bound on the ith eigenvalue in the sense of Theorem 2.1 it is necessary that $\lambda_i(E_1) < \alpha_i$. Theorem 2.2 gives a sufficient condition for this inequality to hold.

THEOREM 2.2. *Consider E as given in* (2.3) (*together with the definitions in* §1). *Then the inequality*

$$\lambda_{i+1}(A) - \lambda_i(A) > 2\|B\|_2 \tag{2.15}$$

is a sufficient condition for the inequality

$$\alpha_i \equiv \max_{j=1,\cdots,i+1} [\lambda_j(A) + \lambda_{i-j+2}(B)] > \lambda_i(E_i), \qquad i = 1, \cdots, n-1, \quad n \geqq 2 \tag{2.16}$$

to hold.

Proof. By definition (2.16)

$$\alpha_i \geqq \lambda_{i+1}(A) + \lambda_1(B) \geqq \lambda_{i+1}(A) - \|B\|_2. \tag{2.17}$$

From the monotonicity principle

$$\lambda_i(E_1) \equiv \lambda_i(\Lambda_1 + \Phi_1^t B \Phi_1) \leqq \lambda_i(\Lambda_1) + \lambda_n(\Phi_1^t B \Phi_1). \tag{2.18}$$

But, $\lambda_i(\Lambda_1) = \lambda_i(A)$ (see problem definition in §1) and since Φ_1 is an orthonormal matrix $\|\Phi_1^t B \Phi_1\|_2 \leqq \|B\|_2$ and it follows that

$$\lambda_i(E_1) \leqq \lambda_i(A) + \|B\|_2. \tag{2.19}$$

Combining (2.17) and (2.19) we obtain the following inequality:

$$\lambda_{i+1}(A) - \lambda_i(A) > 2\|B\|_2$$

as a sufficient condition for $\alpha_i > \lambda_i(E_1)$ to hold. □

Example. To demonstrate the bounds for the eigenvalues we consider the following four-by-four symmetric matrices:

$$A = \begin{bmatrix} 4.0 & 3.0 & 0.0 & 0.0 \\ 3.0 & 7.0 & 2.0 & 0.0 \\ 0.0 & 2.0 & 9.0 & 5.0 \\ 0.0 & 0.0 & 5.0 & 11.0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & k & 0.0 \\ 0.0 & k & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}.$$

The eigenvalues of A are

$$\lambda(A) = \{1.8681, 4.7344, 9.0780, 15.3194\},$$

with corresponding normalized eigenvectors

$$\Phi = [\Phi_1 \Phi_2] = \left[\begin{array}{cc:cc} 0.78063 & 0.40063 & -0.47754 & 0.04533 \\ -0.55474 & 0.09808 & -0.80832 & 0.17102 \\ 0.25248 & -0.71204 & -0.12354 & 0.64341 \\ -0.13824 & 0.56822 & 0.32140 & 0.74479 \end{array}\right].$$

Suppose only $\Lambda_1 \equiv \operatorname{diag}\{1.8681, 4.7344\}$, Φ_1, and B are given.

Fig. 2.1 demonstrates the monotonicity lower bound for $\lambda_1(A + B)$ (i.e., $\lambda_1(A) + \lambda_1(B) \leqq \lambda_1(A + B)$) and the lower bound for $\lambda_1(A + B)$ in view of Theorem 2.1, as a function of the parameter k. Note that for $-1.9 < k < 2.6$ the lower bound obtained by

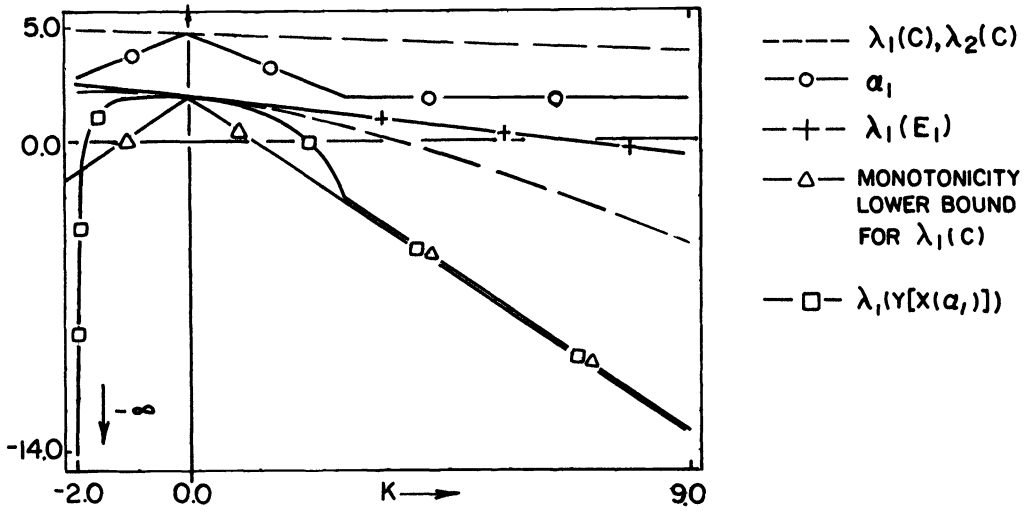


FIG. 2.1. *Eigenvalue bounds.*

the present method is better than the monotonicity lower bound. Also note that for $\lambda_2(A) - \lambda_1(A) = 2.8663 > 2\|B\|_2 = 2|k|$, i.e., $|k| < 1.43315$, the lower bound exists (according to Theorem 2.2). As seen in the figure, when $\alpha_1 \to +\lambda_1(E_1)$, we get $\lambda_1(Y[X(\alpha_1)]) \to -\infty$, since $\det(E_1 - \alpha_1 I) \to -0$.

3. Error bounds for the eigenvectors. Suppose that the orthonormal matrix $U \in R^{n\times p}$ $(n \geqq p)$ spans an invariant subspace for E_1. Then the Ritz matrix $W = \Phi_1 U$ approximates a p-dimensional invariant subspace of C from span (Φ_1). Associated with W there is defined a residual matrix $R(C, W) \equiv CW - WW^tCW$.

We adopt the approach taken by Davis and Kahan [8], which gives a bound on the angle between the eigenvectors of C and their Ritz approximation from a subspace. This approach requires having the residual matrix and the spectrum gap. It is possible, in our problem, to obtain the gap by Theorem 2.1, however the residual matrix, $R(C, W)$, cannot be constructed directly from its definition since C is not given explicitly.

In what follows it is shown how that difficulty is circumvented. We start with the following proposition.

PROPOSITION 3.1. *Suppose that the columns of* $\Phi_1 \in R^{m\times n}$ *are orthogonal eigenvectors of the symmetric matrix* A, *such that* $\Phi_1^t A\Phi_1 = \Lambda_1$, *where* $\Lambda_1 = \operatorname{diag}\{\lambda_i; i = 1, \cdots, n\}$. *Let* $U \in R^{n\times p}$ $(n \geqq p)$ *be any orthonormal matrix. Denote* $W = \Phi_1 U$. *Then*

$$R(A, W) = \Phi_1(I_n - UU^t)\Lambda_1 U \tag{3.1}$$

where $R(A, W) \equiv AW - WW^tAW$.

Proof.

$$\begin{aligned} R(A, W) &= A\Phi_1 U - \Phi_1 U(\Phi_1 U)^t A\Phi_1 U \\ &= (A\Phi_1 - \Phi_1 UU^t\Phi_1^t A\Phi_1)U \\ &= (A\Phi_1 - \Phi_1 UU^t\Lambda_1)U. \end{aligned} \tag{3.2}$$

Since the columns of Φ_1 are eigenvectors of A we have

$$A\Phi_1 - \Phi_1\Lambda_1 = 0. \tag{3.3}$$

By using (3.2) and (3.3) we get

$$R(A, W) = \Phi_1(I_n - UU^t)\Lambda_1 U. \tag{3.4}$$ □

The bound on the angle between the invariant subspace and the subspace, which is spanned by the Ritz vectors, follows from the next theorem.

THEOREM 3.2. *Suppose the orthonormal matrix* $U \in R^{n\times p}$ $(p \leqq n)$ *spans an invariant subspace for* $\Phi_1^t C\Phi_1$, *such that* $U^t\Phi_1^t C\Phi_1 U = \Theta$, *where*

$$\Theta = \operatorname{diag}\{\theta_i; i = t, \cdots, t+p; t \geqq 1, t+p \leqq n\}$$

and θ_i *is the ith smallest Ritz value of* C *from span* (Φ_1). *Denote the Ritz matrix* $W = \Phi_1 U$. *Then the angle* π *between the subspace that is formed by* span (W) *and the corresponding invariant subspace of* C *satisfies the following inequality*:

$$\sin\pi \leqq \frac{\|\Phi_1(I_n - UU^t)\Lambda_1 U + BW - WW^tBW\|_2}{\tau} \tag{3.5}$$

where τ *is the gap between the spectrum of* Θ *and the complementary spectrum of* C.

Proof. Using the Davis and Kahan result [8, p. 225], we have

$$\sin\pi \leqq \frac{\|R(C, W)\|_2}{\tau}.$$

But

$$R(C, W) = R(A, W) + R(B, W).$$

In Proposition 3.1 we have shown that $R(A, W) = \Phi_1(I_n - UU^t)\Lambda_1 U$, and by definition we have

$$R(B, W) \equiv BW - WW^tBW.$$

The proof is then completed by back substitutions. □

The desired gap can be bounded by using Theorem 2.1.

4. Application. Consider a linear vibratory system with m degrees of freedom that is characterized by the following generalized eigenvalue problem:

$$K\Psi = M\Psi\Omega^2, \qquad M, K, \Psi \in R^{m\times m}, \qquad \Omega^2 = \operatorname{diag}\{\omega_i^2, i = 1, \cdots, m\}, \tag{4.1}$$

where K and M are the stiffness and the mass matrices respectively, and ω_i is the ith lowest natural frequency of the system. Here K is a nonnegative symmetric matrix and M is a positive definite symmetric matrix. Partition Ψ and Ω^2 in the form $\Psi = [\Psi_1 \Psi_2]$ and

$$\Omega^2 = \begin{bmatrix} \Omega_1^2 & 0 \\ 0 & \Omega_2^2 \end{bmatrix}$$

where $\Psi_1 \in R^{m\times n}$ and $\Omega_1^2 \in R^{n\times n}$. Let the symmetric matrix $\Delta K \in R^{m\times m}$ be the modification of K. By a vibration test it is possible to extract Ψ_1 and Ω_1^2 experimentally from excitation and response measurements [17], [19]. Suppose we design a modification ΔK of the stiffness matrix, leaving the mass matrix practically unchanged.

Summarizing, Ψ_1, Ω_1^2, M, and ΔK are given, whereas K is unknown. Denote the following:

$$\Lambda_1 = \Omega_1^2, \qquad \Phi_1 = M^{1/2}\Psi_1(\Psi_1^t M\Psi_1)^{-1/2}, \qquad B = M^{-1/2}\Delta K M^{-1/2}. \tag{4.2}$$

Theorem 2.1 gives bounds on the lowest eigenvalues of the pencil $(K + \Delta K, M)$, whereas Theorem 3.2 gives a bound on the angle between the invariant subspace of $M^{-1/2}(K + \Delta K)M^{-1/2}$ and its Ritz approximation from span (Φ_1).

Example 4.1.

Complete system description. Consider the three degrees of freedom vibratory system shown in Fig. 4.1(a). This system is characterized by the eigenproblem of (4.1).

In this case the mass and stiffness matrices are

$$M = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.25 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, \qquad K = \begin{bmatrix} 2000.0 & -1000.0 & -1000.0 \\ -1000.0 & 2000.0 & -1000.0 \\ -1000.0 & -1000.0 & 2000.0 \end{bmatrix}.$$

The solution of eigenproblem (4.1) yields

$$\Omega^2 \equiv \begin{bmatrix} \Omega_1^2 & 0 \\ 0 & \Omega_2^2 \end{bmatrix} = \left[\begin{array}{cc:c} 0.0 & 0.0 & 0.0 \\ 0.0 & 3000.0 & 0.0 \\ \hdashline 0.0 & 0.0 & 9000.0 \end{array}\right]$$

and

$$\Psi \equiv [\Psi_1 | \Psi_2] = \left[\begin{array}{cc:c} 1.0 & -1.0 & 1.0 \\ 1.0 & 0.0 & -8.0 \\ 1.0 & 1.0 & 1.0 \end{array}\right].$$

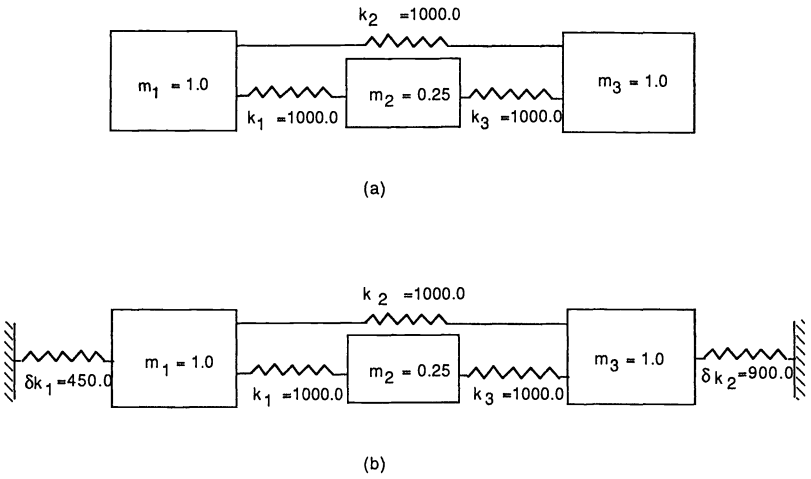


FIG. 4.1. (a) *Unmodified system.* (b) *Modified system.*

The vibratory system is now modified by the addition of two springs as shown by Fig. 4.1(b). The modification is represented by the matrix ΔK, where

$$\Delta K = \begin{bmatrix} 450.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 900.0 \end{bmatrix}.$$

The eigenproblem of the modified system is

$$(K+\Delta K)\bar{\Psi} = M\bar{\Psi}\bar{\Omega}^2. \tag{4.3}$$

The solution of eigenproblem (4.3) yields

$$\bar{\Omega}^2 = \begin{bmatrix} 580.41 & 0.0 & 0.0 \\ 0.0 & 3688.09 & 0.0 \\ 0.0 & 0.0 & 9081.50 \end{bmatrix}$$

and

$$M^{1/2}\bar{\Psi}(\bar{\Psi}^t M\bar{\Psi})^{-1/2} = \begin{bmatrix} -0.70714 & 0.66437 & -0.24201 \\ -0.35539 & -0.03806 & 0.93394 \\ -0.61128 & -0.74643 & -0.26302 \end{bmatrix}.$$

Problem definition (partial system). Suppose that Ω_1^2, Ψ_1, M, and ΔK are given (whereas K, Ω_2^2, and Ψ_2 are unknown).

(a) Find upper and lower bounds for the smallest eigenvalue of the pencil $(K + \Delta K, M)$.

(b) Find a bound for the angle between the eigenvector of $M^{-1/2}(K + \Delta K)M^{-1/2}$ corresponding to the smallest eigenvalue and its Ritz approximation from span $(M^{1/2}\Psi_1)$.

Transformation to the standard eigenvalue problem. The transformation of this problem to the standard eigenvalue problem via (4.2) yields

$$\Lambda_1 \equiv \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 3000.0 \end{bmatrix}, \quad \Phi_1 = \begin{bmatrix} 2/3 & -\sqrt{2}/2 \\ 1/3 & 0.0 \\ 2/3 & \sqrt{2}/2 \end{bmatrix}, \quad B = \begin{bmatrix} 450.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 900.0 \end{bmatrix}.$$

Bounds for eigenvalues. We will first show how to use Theorem 2.1 to get upper and lower bounds for the smallest eigenvalue of the modified system.

Using (2.7) we get

$$\alpha_1 = \max\,[\lambda_1(A)+\lambda_2(B), \lambda_2(A)+\lambda_1(B)] = \max\,[0.0+450.0, 3000.0+0.0] = 3000.0.$$

Since

$$\lambda_2(A)-\lambda_1(A) = 3000.0 > 2\|B\|_2 = 1800,$$

it follows by Theorem 2.2 that there exists a sufficient condition for having the lower bound for $\lambda_1(C)$.

The leading principal submatrix E_1 is calculated by (2.3)

$$E_1 = \Lambda_1 + \Phi_1^t B \Phi_1 = \begin{bmatrix} 600.0 & 150\sqrt{2} \\ 150\sqrt{2} & 3675 \end{bmatrix}.$$

The product $E_2 E_2^t$ can be evaluated using (2.14), resulting in

$$E_2 E_2^t = \begin{bmatrix} 45000 & 11250\sqrt{2} \\ 11250\sqrt{2} & 5625 \end{bmatrix}.$$

The spectral decomposition of $E_2 E_2^t$ is

$$E_2 E_2^t = \begin{bmatrix} 1/3 & 2\sqrt{2}/3 \\ -2\sqrt{2}/3 & 1/3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 50625 \end{bmatrix} \begin{bmatrix} 1/3 & -2\sqrt{2}/3 \\ 2\sqrt{2}/3 & 1/3 \end{bmatrix}.$$

Hence, by Remark 2.1 the matrix S can be taken as

$$S = \sqrt{50625} \begin{bmatrix} 2\sqrt{2}/3 \\ 1/3 \end{bmatrix} = \begin{bmatrix} 150\sqrt{2} \\ 75 \end{bmatrix}.$$

By using (2.6) we get $X(\alpha_1) = 2993.92$. At this stage the matrix $Y[X(\alpha_1)]$ is completely defined as

$$Y[X(\alpha_1)] = \begin{bmatrix} 600 & 150\sqrt{2} & 150\sqrt{2} \\ 150\sqrt{2} & 3675 & 75 \\ 150\sqrt{2} & 75 & 2993.92 \end{bmatrix}.$$

Finally, from (2.9) we obtain the desired bound for $\lambda_1(A+B)$:

$$\lambda_1(Y[X(\alpha_1)]) = 567.84 \leqq \lambda_1(A+B) \leqq \lambda_1(E_1) = 585.43.$$

Check. The exact smallest eigenvalue of C is $\lambda_1(C) = 580.41$.
The monotonicity lower bound for $\lambda_1(A+B)$ is $\lambda_1(A)+\lambda_1(B) = 0.0 < \lambda_1(A+B)$. Therefore, using the principle of monotonicity and the interlacing property for eigenvalues, we bound $\lambda_1(A+B)$ in [0.0, 585.43], whereas the technique presented here bounds $\lambda_1(A+B)$ in [567.84, 585.43].

A bound for an eigenspace. We denote the spectral decomposition of E_1 as follows:

$$E_1 = U\Theta U^t \quad \text{where } U \equiv [u_1 | u_2] = \begin{bmatrix} 0.99765 & 0.06850 \\ -0.06850 & 0.99765 \end{bmatrix}$$

and

$$\Theta = \begin{bmatrix} 585.43 & 0.0 \\ 0.0 & 3689.56 \end{bmatrix}.$$

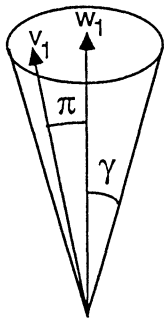


FIG. 4.2. *A bound for an eigenvector.*

The Ritz vector

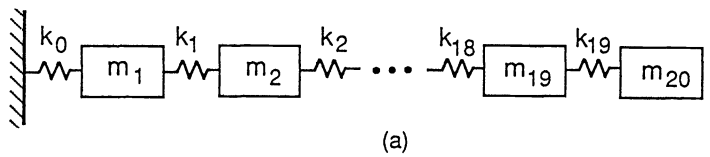
$$w_1 \equiv \Phi_1 u_1 = \begin{bmatrix} 0.71354 \\ 0.33255 \\ 0.61666 \end{bmatrix}$$

approximates an eigenvector of C (denoted here by v_1).

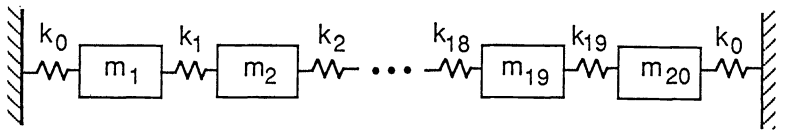
Since α_1 is a lower bound for $\lambda_2(C)$ and since $\lambda_1(E_1)$ is an upper bound for $\lambda_1(C)$, we deduce that the desired gap τ can be taken as $\tau = \alpha_1 - \lambda_1(E_1) = 2414.57$. It is impossible to evaluate the residual matrix $R(C, w_1)$ directly from its definition since C is unknown. Hence we make use of Proposition 3.1 to get

$$R(C, w_1) = \Phi_1(I - u_1 u_1^t)\Lambda_1 u_1 + Bw_1 - w_1 w_1^t Bw_1 = \begin{bmatrix} 48.68 \\ -194.69 \\ 48.68 \end{bmatrix}.$$

The residual norm is $\| R(C, w_1) \|_2 = 206.5$.



(a)



(b)

$m_i = 1$, $i = 1, \ldots, 20$; $k_0 = 3000$; $k_i = 15000$, $i = 1, \ldots, 19$

FIG. 4.3. (a) *The original system.* (b) *The modified system.*

TABLE 4.1
Upper and lower bounds for the eigenvalues.

(a) Bounds for $\lambda_1(K + \Delta K, M)$

n	4	6	8	10	12	14	16	18
Upper bounds	191.79	187.48	185.69	184.80	184.34	184.09	183.98	183.93
Lower bounds	105.18	113.34	124.09	137.34	152.20	166.52	177.31	182.74

(b) Bounds for $\lambda_2(K + \Delta K, M)$

n	4	6	8	10	12	14	16	18
Upper bounds	836.54	821.76	816.13	813.43	812.03	811.30	810.96	810.83
Lower bounds	686.66	701.27	720.49	743.02	766.58	787.56	802.25	809.30

Since $\| R(C, w_1) \|_2 < \tau$, the use of Theorem 3.2 is possible and we obtain that the bound on the angle between v_1 and w_1 is

$$\pi \leqq \arcsin \| R(C, w_1) \|_2 / \tau = \arcsin 0.085 = 4.87°.$$

Check. The exact eigenvector of C corresponding to $\lambda_1(C)$ is

$$v_1 = \begin{bmatrix} 0.70714 \\ 0.35539 \\ 0.61128 \end{bmatrix}.$$

Hence

$$\pi \equiv \arccos | w_1^t v_1 | / \| w_1 \|_2 \cdot \| v_1 \|_2 = \arccos 0.9997 = 1.39°.$$

In this case a geometrical interpretation is possible as shown in Fig. 4.2. It can be seen that the sought eigenvector lies in a cone of an apex angle $\gamma = 4.87°$ whose axis of symmetry is the Ritz vector w_1.

Example 4.2. Consider the 20 degrees of freedom system shown in Fig. 4.3(a). The system is modified by the addition of a spring between m_{20} and the ground, as shown in Fig. 4.3(b). The upper and the lower bounds for the two smallest eigenvalues of the modified system as a function of n are shown in Table 4.1. Note that the two smallest eigenvalues of the original system are $\lambda_1(K, M) = 62.2393$ and $\lambda_2(K, M) = 592.1683$. The two smallest eigenvalues of the modified system are $\lambda_1(K + \Delta K, M) = 183.9273$ and $\lambda_2(K + \Delta K, M) = 810.8037$.

We thus note that the present method provides a systematic way to define the possible location of several natural frequencies of the modified system, based on vibration test data and on the analytical model of the incremental stiffness matrix.

5. Summary. Upper and lower bounds for the eigenvalues of the sum of two symmetric matrices $A + B$ where part of the eigenpairs of A are unknown have been developed. Based on the given data, the angle between an invariant subspace of $A + B$ and the subspace, which is spanned by certain Ritz vectors, has been bounded. An application concerning the possible location of part of the natural frequencies of a modified structure based on vibration test data of the original structure and on the analytical incremental stiffness matrix has been presented. The results are of engineering interest.

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] A. BERMAN AND W. G. FLANNELLY, *Theory of incomplete models of dynamic structures*, AIAA J., 9 (1971), pp. 1481–1487.

[3] A. BERMAN, *System identification of structural dynamic models—theoretical and practical bounds*, AIAA Paper 84-0929, 1984, pp. 123–129.

[4] S. G. BRAUN AND Y. M. RAM, *Determination of structural modes via the Prony model; system order and noise induced poles*, J. Acoust. Soc. Amer., 5 (1987), pp. 1447–1459.

[5] ———, *Structural parameters identification in the frequency domain: The use of overdetermined systems*, Trans. ASME J. Dynamic System Measurement and Control, 109 (1987), pp. 120–123.

[6] D. H. F. CHU, *Modal Testing and Modal Refinement*, American Society of Mechanical Engineers, New York, 1983.

[7] C. R. CRAWFORD, *A stable generalized eigenvalue problem*, SIAM J. Numer. Anal., 13 (1976), pp. 854–860.

[8] C. DAVIS AND W. KAHAN, *The rotation of eigenvectors by a perturbation* III, SIAM J. Numer. Anal., 7(1970), pp. 1–46.

[9] C. DAVIS, W. KAHAN, AND H. F. WEINBERGER, *Norm-preserving dilations and their applications to optimal error bounds*, SIAM J. Numer. Anal., 19 (1982), pp. 445–469.

[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations,* Johns Hopkins University Press, Baltimore, MD, 1983.

[11] S. B. HALEY, *The generalized eigenproblem: Pole-zero computation*, Proc. IEEE, 76 (1988), pp. 103–119.

[12] W. KAHAN, *Inclusion theorems for clusters of eigenvalues of Hermitian matrices*, Tech. Report No. CS42, Computer Science Department, University of Toronto, Toronto, Ontario, Canada, 1967.

[13] W. KAHAN, B. N. PARLETT, AND E. JIANG, *Residual bounds on approximate eigensystem of nonnormal matrices*, SIAM J. Numer. Anal., 19 (1982), pp. 470–484.

[14] N. J. LEHMANN, *On optimal eigenvalue localization in the solution of symmetric matrix problems*, Numer. Math., 8 (1966), pp. 42–55.

[15] M. LINK, *Theory of a method for identifying incomplete system matrices from vibration test data*, Z. Flugwiss. Weltraumforsch., 9 (1985), pp. 76–88.

[16] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[17] Y. M. RAM, S. G. BRAUN, AND J. J. BLECH, *Structural modification in truncated systems by the Rayleigh–Ritz approach*, J. Sound and Vibration, 125 (1988), pp. 203–209.

[18] A. SIMPSON, *Scanning Kron's determinant*, Quart. J. Mech. Appl. Math., 27 (1974), pp. 27–43.

[19] R. SNOEYS, P. SAS, W. HEYLEN, AND H. VAN.DER AUWERAER, *Trends in Experimental Modal Analysis*, Mechanical Systems and Signal Processing, 1 (1987), pp. 5–27.

[20] G. W. STEWART, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.

[21] ———, *Perturbation bounds for the definite generalized eigenvalue problem*, Linear Algebra Appl., 23 (1979), pp. 69–85.

[22] ———, *On the sensitivity of the eigenvalue problem* $Ax = \lambda Bx$, SIAM J. Numer. Anal., 9 (1972), pp. 669–686.

[23] G. STRANG, *Linear Algebra and Its Applications*, Academic Press, New York, 1980.

[24] R. C. THOMPSON, *The characteristic polynomial of a principal subpencil of a Hermitian matrix pencil*, Linear Algebra Appl., 14 (1976), pp. 135–177.

[25] H. F. WEINBERGER, *Variational Methods for Eigenvalue Approximation*, CBMS–NSF Regional Conference Series in Applied Mathematics 12, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1974.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 565–574, October 1990

007

ON THE INERTIA OF INTERVALS OF MATRICES*

DANIEL HERSHKOWITZ† AND HANS SCHNEIDER‡

Abstract. The inertia of intervals and lines of matrices is investigated. For complex $n \times n$ matrices A and B it is shown that, under mild nonsingularity conditions, $A + tB$ changes inertia at no more than n^2 real values of t. Conditions are given for the constancy of the inertia of $A + tB$, where t lies in a real interval. These conditions generalize and organize some known results.

Key words. inertia, constant inertia, inertia change point, interval of matrices, matrix stability, Lyapunov operators, Z-matrices

AMS(MOS) subject classification. 15

1. Introduction. Bialas [1], Johnson and Rodman [4], Väliaho [7], and Fu and Barmish [2], [3] have recently studied the inertia of intervals and lines of matrices. We extend these investigations under nonsingularity conditions. While some of our results are not difficult and are related to known results, taken together they show interrelations between various types of conditions, and as such they organize knowledge in this area of inertia theory.

Let A and B be square complex matrices and suppose there is a real t such that the Lyapunov matrix $L(A + tB)$ associated with $A + tB$ is nonsingular. We show that $A + tB$ changes inertia at no more than n^2 values of t. Let T be an interval, i.e., a connected subset of the real numbers. Under the assumption that $L(A)$ is nonsingular, we state our principal condition,

(CI) $\quad A + tB$ has constant inertia of type $(\pi, \nu, 0)$ for every t in T,

and we compare several other conditions (some obviously equivalent) to (CI). Some of these conditions involve the real eigenvalues of $A^{-1}B$ and of $L(A)^{-1}L(B)$. Each of the conditions either implies or is implied by (CI), but not all are equivalent in general. By adding additional requirements on a single matrix or on the interval, such as stability, the reality of all eigenvalues, or a condition we call Property X (which Z-matrices satisfy), some implications in one direction become equivalences.

Section 2 of our paper contains notation, definitions, and some well-known results stated for easy reference. Section 3 contains preliminary results on eigenvalues and results on changes of inertia. Our main results on intervals with constant inertia, summarized above, may be found in § 4. In § 5 we give some applications to the convex hull of two matrices. We derive results from [1]–[4] and [7].

* Received by the editors December 19, 1988; accepted for publication (in revised form) September 8, 1989. This research was supported by grant 85-00153 from the United States–Israel Binational Science Foundation (BSF), Jerusalem, Israel.

† Mathematics Department, Technion–Israel Institute of Technology, Haifa 32000, Israel (MAR23AA@TECHNION.BITNET). This research was completed while this author was a visiting professor at the University of Wisconsin, Madison, Wisconsin 53706.

‡ Mathematics Department, University of Wisconsin, Madison, Wisconsin 53706 (hans@math.wisc.edu). The research of this author was supported in part by National Science Foundation grants DMS-8521521, DMS-8901445, and EMS-8718971.

Our principal theorems are proved for the case of general complex matrices, and we then apply the results to Hermitian matrices and Z-matrices.

Properties of the Lyapunov operator $A \to L(A)$ that are crucial to our results are the following:

(1.1) If λ is an eigenvalue of A, then $2 \operatorname{Re}(\lambda)$ is an eigenvalue of $L(A)$.

(1.2) If t is the maximal (minimal) eigenvalue of $L(A)$, then there is an eigenvalue λ of A with $2 \operatorname{Re}(\lambda) = t$.

Similar results may be proved for any real linear operator, from the space of complex $n \times n$ matrices into a space of matrices, which satisfies (1.1) and (1.2). For spaces of real matrices, another operator that satisfies these conditions is found in [1] and [3]. The results in [1] are proved for that operator, whereas in [3] results are proved for all operators satisfying (1.1) and (1.2). The results in [2] are proved for the Lyapunov operator, as in the present paper. Only real matrices are considered in [1]–[3]. In referring to the results of these papers in the sequel, we do not distinguish between the various operators involved. We also observe that the results in [7] deal with real symmetric matrices (where there is no need to employ the Lyapunov operator), but some results in [7] hold under weaker nonsingularity assumptions.

2. Notation and preliminaries. As usual, $\mathbb{R}$ and $\mathbb{C}$ denote the real and complex fields, respectively, and $\mathbb{C}^{nn}$ denotes the complex space of all complex matrices. By $\mathscr{H}_n$ we denote the *real* space of all $n \times n$ Hermitian matrices. In this paper, A and B will always be $n \times n$ complex matrices that may be considered fixed throughout. The convex hull of A and B is denoted by $\operatorname{conv}(A, B)$. The spectrum of a matrix A is denoted by $\operatorname{spec}(A)$. The spectrum is considered to be a multiset, that is, every eigenvalue is counted as many times as its multiplicity.

Notation 2.1. We denote the following:

$\pi(A)$—the number of eigenvalues of A in the open right halfplane,

$\nu(A)$—the number of eigenvalues of A in the open left halfplane,

$\delta(A)$—the number of eigenvalues of A on the imaginary axis.

DEFINITION 2.2. The inertia $\operatorname{In}(A)$ of A is defined to be the triple $(\pi(A), \nu(A), \delta(A))$.

DEFINITION 2.3. (i) The matrix A is said to be *positive* [*negative*] *stable* if all its eigenvalues are in the open right [left] halfplane.

(ii) The matrix A is said to be *positive* [*negative*] *semistable* if all its eigenvalues are in the closed right [left] halfplane.

(iii) The matrix A is said to be *positive* [*negative*] *near-stable* if A is positive [negative] semistable but not positive [negative] stable.

In this paper "stable," "semistable," and "near-stable" may be interpreted consistently to mean either "positive stable," "positive semistable," and "positive near-stable" or "negative stable," "negative semistable," and "negative near-stable."

DEFINITION 2.4. The *Lyapunov operator* (*or Lyapunov matrix*) $L(A)$ of A is defined to be the linear operator of $\mathscr{H}_n$ into itself given by

$$L(A)H = AH + HA^*.$$

For reference, we collect some properties of the operator $L(A)$. We follow the notation of [5] for the Kronecker (or tensor) product of matrices.

PROPOSITION 2.5. *We have*

(i) $L(A) = I \otimes A + \bar{A} \otimes I$.

(ii) *The spectrum of* $L(A)$ *is the multiset* $\{\lambda + \bar{\mu} : \lambda, \mu \in \text{spec}(A)\}$.

(iii) $L(A)$ *is nonsingular if and only if* $\lambda + \bar{\mu} \neq 0$ *for* $\lambda, \mu \in \text{spec}(A)$.

(iv) A *is stable* [*semistable*] (*near-stable*) *if and only if* $L(A)$ *is*.

(v) *The mapping* $A \to L(A)$ *is real linear, i.e.,* $L(sA + tB) = sL(A) + tL(B)$, *for all real numbers* s *and* t.

Proof. Parts (i) and (ii) are standard (e.g., see [5, Chap. 12]). Parts (iii) and (iv) follow immediately from (ii). Part (v) follows from the definition of $L(A)$. □

In our proofs (as in the proof of almost any inertia theorem) we use properties often called "continuity of eigenvalues." The basic result is stated as Lemma 3 in [6]. Here we state consequences of this lemma in the forms needed for our applications.

LEMMA 2.6. (i) *Let* $A(t)$ *be a continuous matrix function of the real variable* t. *Let* λ *be an eigenvalue of* $A(0)$. *Let* S *be a disc in the complex plane with center at* λ *such that* S *does not contain any other eigenvalue of* $A(0)$. *If there exists a positive* δ *such that for all* t, $0 < t < \delta$, $A(t)$ *has an even number of eigenvalues in* S, *then the multiplicity of* λ *as an eigenvalue of* $A(0)$ *is even*.

(ii) *If* A *has no imaginary eigenvalues, then for all sufficiently small* ε, *we have* $\text{In}(A + \varepsilon B) = \text{In}(A)$.

(iii) *If* $\text{In}(A) \neq \text{In}(B)$, *then there is a matrix* $C \in \text{conv}(A, B)$ *that has an imaginary eigenvalue*.

(iv) *If* A *is stable but* B *is not stable, then there is a matrix* $C \in \text{conv}(A, B)$ *that is near-stable*.

Proof. Parts (i) and (ii) follow from Lemma 3 of [6]. Parts (iii) and (iv) follow from (ii) using the completeness of the real numbers and the connectedness of the interval $[0, 1]$. □

Convention 2.7. By the term "*interval*" we mean a connected subset of the real line. That is, open intervals, closed intervals, half-open intervals, halflines and the whole real line are intervals.

DEFINITION 2.8. Let T be an interval. The *matrix interval* $S(A, B; T)$ of matrices is defined to be the set $\{A + tB : t \in T\}$.

DEFINITION 2.9. Let $t_0 \in \mathbb{R}$. We say that t_0 is an *inertia change point* for $S(A, B; \mathbb{R})$ if for every $\varepsilon > 0$ there exists $t \in \mathbb{R}$ such that $|t - t_0| < \varepsilon$ and $\text{Inertia}(A + tB) \neq \text{Inertia}(A + t_0 B)$.

DEFINITION 2.10. Let T be an interval.

(i) The interval T is called an *interval of constant inertia* (π, ν, δ) for $S(A, B; \mathbb{R})$ if every matrix in $S(A, B; T)$ has inertia (π, ν, δ).

(ii) The interval T is called an *interval of semiconstant inertia* $(\pi, \nu, 0)$ for $S(A, B; \mathbb{R})$ if every matrix $C \in S(A, B; T)$ such that $\delta(C) = 0$ has $\text{In}(C) = (\pi, \nu, 0)$.

If T is an interval of constant inertia for $S(A, B; \mathbb{R})$ we may also say that $S(A, B; T)$ has constant inertia or, when every matrix in $S(A, B; T)$ is stable [semistable], that $S(A, B; T)$ is stable [semistable].

DEFINITION 2.11. We call (A, B) a *regular pair of matrices* if there exists a $t \in \mathbb{R}$ such that $L(A + tB)$ is nonsingular.

PROPOSITION 2.12. *If* (A, B) *is a regular pair of matrices, then the number of complex numbers* t *for which* $L(A + tB)$ *is singular is at most* n^2.

Proof. Since $L(A + tB)$ is singular if and only if $\det(L(A + tB)) = 0$, and since $p(t) = \det(L(A + tB))$ is a polynomial of degree at most n^2, it follows that either (A, B) is a regular pair, in which case $p(t)$ has at most n^2 roots, or (A, B) is not a regular pair, in which case $p(t) \equiv 0$. □

COROLLARY 2.13. *If (A, B) is a regular pair of matrices, then the number of complex numbers t for which $A + tB$ has an imaginary eigenvalue is at most n^2.*

Proof. The claim follows from Proposition 2.5(ii) and Proposition 2.12. □

Another corollary of Proposition 2.12 is the following.

COROLLARY 2.14. *(A, B) is a regular pair of matrices if and only if (B, A) is a regular pair of matrices.*

Proof. If (A, B) is a regular pair of matrices then, by Proposition 2.12, there exists a nonzero number t such that $L(A + tB)$ is nonsingular. Therefore, $L(A/t + B)$ is nonsingular, and so (B, A) is a regular pair of matrices. □

Since for all complex $n \times n$ matrices A, (I, A) is a regular pair, it follows from Proposition 2.12 and Corollary 2.14 that $L(A + tI)$ is nonsingular for all but at most n^2 complex numbers t.

3. Observations on eigenvalues and inertia. We start with an immediate observation.

OBSERVATION 3.1. Let A be nonsingular, and let t be a nonzero real number. Then the following are equivalent:

(a_1) $-1/t$ is an eigenvalue of $A^{-1}B$.

(a_2) $I + tA^{-1}B$ is singular.

(a_3) $A + tB$ is singular.

Accordingly, we label the three equivalent conditions in Observation 3.1 (under the assumption that A is nonsingular) as condition (a).

If $L(A)$ is nonsingular then, applying Observation 3.1 to $L(A)$ and $L(B)$, we obtain the following equivalent conditions:

(la_1) $L(A) + tL(B)$ is singular.

(la_2) $I + tL(A)^{-1}L(B)$ is singular.

(la_3) $-1/t$ is an eigenvalue of $L(A)^{-1}L(B)$.

By Proposition 2.5(v), condition (la_1) is equivalent to

(la_4) $L(A + tB)$ is singular.

We now label the four equivalent conditions (la_1)–(la_4) (under the assumption that $L(A)$ is nonsingular) as condition (la).

A third condition we will discuss is

(ie) $A + tB$ has an imaginary eigenvalue.

THEOREM 3.2. *Let A and B be $n \times n$ complex matrices, let t be a nonzero real number, and assume that $L(A)$ is nonsingular. Then we have*

$$(a) \rightarrow (ie) \rightarrow (la).$$

Proof. First observe that if $L(A)$ is nonsingular then A is nonsingular, so both conditions (a) and (la) are well defined. The implication (a) → (ie) follows from the trivial implication (a_3) → (ie). The implication (ie) → (la) follows from (ie) → (la_4), which follows from Proposition 2.5(ii). □

Clearly, the converses of the implications (a) → (ie) and (ie) → (la) do not hold under the stated hypotheses.

We now add three more conditions that relate to the previous eight.

(ic) t is an inertia change point for $S(A, B; \mathbb{R})$.

(ns) $A + tB$ is near stable.

(us) $A + tB$ is not stable.

THEOREM 3.3. *Let A and B be $n \times n$ matrices, let t be a nonzero real number, and assume that $L(A)$ is nonsingular. Then we have*

$$(a) \rightarrow (ie) \Leftrightarrow (ic) \rightarrow (la) \rightarrow (us).$$

$$(ns) \nearrow$$

Proof. The implication (ns) $\rightarrow$ (ie) is clear by Definition 2.3(iii). The implication (ic) $\rightarrow$ (ie) follows from Lemma 2.6(ii). The implication (ie) $\rightarrow$ (ic) follows from Corollary 2.13. The implication (la) $\rightarrow$ (us) follows from (la_4) $\rightarrow$ (us), which follows from Proposition 2.5(iv). □

The converses of the implications (ns) $\rightarrow$ (ie) and (la) $\rightarrow$ (us) do not hold. Also, neither (a) $\rightarrow$ (ns) nor (ns) $\rightarrow$ (a) holds.

Theorem 3.3 yields the following corollary.

COROLLARY 3.4. *Suppose that (A, B) is a regular pair of matrices. Then the number of inertia change points for $S(A, B; \mathbb{R})$ is at most n^2.*

Proof. If (A, B) is a regular pair of matrices, then (ic) $\rightarrow$ (ie) holds even if $L(A)$ is singular. To see that, let $A' = A + t'B$, where $t' \in \mathbb{R}$ is chosen so that $L(A')$ is nonsingular. Let t be an inertia change point for $S(A, B; \mathbb{R})$. Obviously, $t - t'$ is an inertia change point for $S(A', B; \mathbb{R})$. By Theorem 3.3 (applied to A' and B), $A' + (t - t')B = A + tB$ has an imaginary eigenvalue. Our claim now follows from Corollary 2.13. □

THEOREM 3.5. *Let (A, B) be a regular pair of matrices and let $t_1, \cdots, t_m$, where $t_1 < \cdots < t_m$, be the inertia change points of $S(A, B; \mathbb{R})$. Let $T_0 = (-\infty, t_1)$, $T_i = (t_i, t_{i-1})$, $i = 1, \cdots, m$, and $T_m = (t_m, \infty)$. Then the intervals T_i, $i = 0, \cdots, m$ are maximal intervals of constant inertia for $S(A, B; \mathbb{R})$ and the inertia of each matrix in $S(A, B; T_i)$ is of the form $(\pi_i, \nu_i, 0)$, $i = 0, \cdots, m$.*

Proof. By standard results in analysis, T is an interval of constant inertia for $S(A, B; \mathbb{R})$ if and only if T contains no inertia change point for $S(A, B; \mathbb{R})$ and so the first part of the theorem follows. The second part of the theorem follows from the equivalence of (ie) and (ic) in Theorem 3.3. □

4. Inertia of intervals. In this section we apply the observations made in the previous section in order to study the relation between global conditions. The global conditions correspond to the negations of the local conditions in the previous section. In these global conditions as well as in the rest of the paper T denotes an interval.

The equivalent conditions

(A_1) $A^{-1}B$ has no eigenvalue with negative reciprocal in T.

(A_2) $I + tA^{-1}B$ is nonsingular for every t in T.

(A_3) $A + tB$ is nonsingular for every t in T.

will be labeled condition (A).

The equivalence of the following four conditions follows from the equivalence of (la_1)-(la_4):

(LA_1) $L(A) + tL(B)$ is nonsingular for every t in T.

(LA_2) $I + tL(A)^{-1}L(B)$ is nonsingular for every t in T.

(LA_3) $L(A)^{-1}L(B)$ has no eigenvalue with negative reciprocal in T.

(LA_4) $L(A + tB)$ is nonsingular for every t in T.

These conditions will be labeled condition (LA).

We also consider the conditions

(IE) $A + tB$ has no imaginary eigenvalue for any t in T.

(CI) T is an interval of constant inertia $(\pi, \nu, 0)$ for $S(A, B; \mathbb{R})$.

THEOREM 4.1. *Let A and B be $n \times n$ complex matrices, let T be an interval, and assume that $L(A)$ is nonsingular. Then we have*

$$(\mathrm{LA}) \rightarrow (\mathrm{IE}) \Leftrightarrow (\mathrm{CI}) \rightarrow (\mathrm{A}).$$

Proof. In view of Theorem 3.3 it is enough to prove the equivalence (IE) $\Leftrightarrow$ (CI). From Theorem 3.3 and the proof of Theorem 3.5 it follows that (IE) implies that $S(A, B; T)$ has constant inertia. By (IE) it follows that the inertia is of type $(\pi, \nu, 0)$. The implication (CI) $\rightarrow$ (IE) is trivial. □

By adding additional requirements, some of the implications in Theorem 4.1 become equivalences, as we will show presently.

THEOREM 4.2. *Let A and B be $n \times n$ complex matrices, let T be an interval, assume that $L(A)$ is nonsingular, and assume that $A + tB$ is stable for some t in T. Then we have*

$$(\mathrm{LA}) \Leftrightarrow (\mathrm{IE}) \Leftrightarrow (\mathrm{CI}) \rightarrow (\mathrm{A}).$$

Proof. In view of Theorem 4.1 it is enough to prove the implication (CI) $\rightarrow$ (LA). Observe that under our additional assumption, (CI) implies that $A + tB$ is stable for every t in T. By the implication (la) $\rightarrow$ (us) in Theorem 3.3 we now obtain (LA). □

The following theorem is found in [2] and [3] for real matrices.

THEOREM 4.3. *Let A and B be $n \times n$ complex matrices, and assume that A is positive stable.*

(i) *If $L(A)^{-1}L(B)$ has no real eigenvalue, then $A + tB$ is stable for every real number t.*

(ii) *If $L(A)^{-1}L(B)$ has real eigenvalues, then let r_1 and r_2 be the greatest and the least real eigenvalues of $L(A)^{-1}L(B)$. Define*

$$t_1 = \begin{cases} -\dfrac{1}{r_1}, & r_1 > 0, \\ -\infty, & r_1 \leqq 0, \end{cases}$$

$$t_2 = \begin{cases} -\dfrac{1}{r_2}, & r_2 < 0, \\ \infty, & r_2 \geqq 0. \end{cases}$$

Then the interval $T = (t_1, t_2)$ is the maximal interval of constant inertia $(n, 0, 0)$ that contains the point $t = 0$.

Proof. Part (i) follows immediately from the equivalence (LA) $\Leftrightarrow$ (CI) in Theorem 4.2.

(ii) Observe that $L(A)^{-1}L(B)$ has no real eigenvalue in $T_1 = (-\infty, -1/t_2)$, nor in $T_2 = (-1/t_1, \infty)$. Therefore, $L(A)^{-1}L(B)$ has no real eigenvalue with negative reciprocal in $(0, t_2)$ or in $(t_1, 0)$. By Theorem 4.2 it follows that $(t_1, 0)$ and $(0, t_2)$ are intervals of constant inertia $(\pi, \nu, 0)$ for $S(A, B; \mathbb{R})$. Since A is stable, it follows from Theorem 3.3 that zero is not an inertia change point for $S(A, B; \mathbb{R})$. Hence, it follows that $T = (t_1, t_2)$ is an interval of constant inertia $(n, 0, 0)$ that contains the point $t = 0$. If $t_1 \neq -\infty$, then it follows that $-1/t_1$ is an eigenvalue of $L(A)^{-1}L(B)$, and by Theorem 4.2 $[t_1, t_2)$ is not an interval of constant inertia $(n, 0, 0)$. Similarly, if $t_2 \neq \infty$, then it follows that $-1/t_2$ is an eigenvalue of $L(A)^{-1}L(B)$, and by Theorem 4.2, $(t_1, t_2]$ is not an interval of constant inertia $(n, 0, 0)$. The maximality of T follows. □

DEFINITION 4.4. A square matrix A is said to have *Property X* if the minimal real part of an eigenvalue of A is an eigenvalue of A.

For example, Hermitian matrices and Z-matrices have Property X.

THEOREM 4.5. *Let A and B be $n \times n$ complex matrices, let T be an interval, assume that $L(A)$ is nonsingular, assume that $A + tB$ is positive stable for some t in T, and assume that $A + tB$ has Property X for every t in T. Then we have*

$$(\mathrm{CI}) \Leftrightarrow (\mathrm{LA}) \Leftrightarrow (\mathrm{IE}) \Leftrightarrow (\mathrm{A}).$$

Proof. Since $A + tB$ is positive stable for some t in T, and since $A + tB$ has Property X for every t in T, it follows, using continuity arguments (see Lemma 2.6(iv)) that $(\mathrm{A}_3) \rightarrow (\mathrm{IE})$. So $(\mathrm{A}) \rightarrow (\mathrm{IE})$, and our claim follows from Theorem 4.2. □

THEOREM 4.6. *Let A and B be $n \times n$ complex matrices, let T be an interval, assume that $L(A)$ is nonsingular, and assume that all eigenvalues of $A + tB$ are real for every t in T. Then we have*

$$(\mathrm{LA}) \rightarrow (\mathrm{IE}) \Leftrightarrow (\mathrm{CI}) \Leftrightarrow (\mathrm{A}).$$

Proof. The implication $(\mathrm{A}_3) \rightarrow (\mathrm{CI})$ follows immediately by continuity (see Lemma 2.6(iii)). So $(\mathrm{A}) \rightarrow (\mathrm{CI})$, and the claim follows from Theorem 4.1. □

We now consider matrix intervals with the same inertia except for a finite number of points.

First, we restate the implications (ic) $\rightarrow$ (la) and (a) $\rightarrow$ (ic) of Theorem 3.3 in a somewhat different form together with a partial converse.

PROPOSITION 4.7. *Let A and B be $n \times n$ complex matrices and assume that $L(A)$ is nonsingular. Let G be the set of inertia change points for $S(A, B; \mathbb{R})$, and let t be a nonzero number.*

(i) *If $t \in G$, then $-1/t$ is an eigenvalue of $L(A)^{-1}L(B)$.*

(ii) *If $-1/t$ is an eigenvalue of $A^{-1}B$, then $t \in G$.*

The converses of Proposition 4.7(i) and (ii) do not hold in general. We give a counterexample to the converse of Proposition 4.7(i).

Example 4.8. Consider the matrices

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

Observe that $L(A)$ is nonsingular, and that $A + tB$ has the inertia $(1, 1, 0)$ for all t in $[-0.5, \infty)$ except $t = -0.5$. However, it is easy to verify that $L(A + B)$ is singular and hence, by the equivalence of conditions (la_3) and (la_4), $-1/t$ is an eigenvalue of $L(A)^{-1}L(B)$ also for $t = 1$.

Proposition 4.7 does not give necessary and sufficient conditions for a point t to belong to the exceptional set G of inertia change points for $S(A, B; \mathbb{R})$. However it does lead to a finite algorithm for finding these points.

ALGORITHM 4.9. For the sake of simplicity, we assume that $L(A)$ is nonsingular.

Step 1. Find the real nonzero eigenvalues of $L(A)^{-1}L(B)$.

Step 2. Take the negative reciprocals $t_1, \cdots, t_m$ of the numbers found in Step 1. The inertia change points for $S(A, B; \mathbb{R})$ are those t_i, $i \in \{1, \cdots, m\}$, for which $A + t_i B$ has an imaginary eigenvalue.

Necessary and sufficient conditions for a point t to be an inertia change points for $S(A, B; \mathbb{R})$ may be obtained under additional assumptions, as will be demonstrated in the sequel.

First we consider intervals of semistability.

THEOREM 4.10. *Let A and B be $n \times n$ complex matrices, and assume that $L(A)$ is nonsingular. Let T be an interval of semistability for $S(A, B; \mathbb{R})$. Then*

(i) *For $t \in T$, $A + tB$ is near-stable if and only if $t \neq 0$ and $-1/t$ is an eigenvalue of $L(A)^{-1}L(B)$.*

(ii) *All the eigenvalues of $L(A)^{-1}L(B)$ whose negative reciprocals lie in the interior of T have even multiplicity.*

(iii) *If A and B are real, then all the eigenvalues of $A^{-1}B$ whose negative reciprocals lie in the interior of T have even multiplicity.*

Proof. (i) If T consists of one point t_0 then, since $A + t_0 B$ is semistable, it follows by Proposition 2.5(ii) that $A + t_0 B$ is near stable if and only if $L(A + t_0 B)$ is singular (so $t_0 \neq 0$ since $L(A)$ is nonsingular), which is true if and only if $-1/t_0$ is an eigenvalue of $L(A)^{-1}L(B)$. If T consists of more than one point, then it consists of infinitely many points. By Theorem 3.3, every t for which $A + tB$ is near stable is an inertia change point for $S(A, B; \mathbb{R})$. In view of Corollary 3.4, $A + tB$ is stable for all $t \in T$ except for a finite number of t's. Part (i) now follows immediately from the equivalence (LA) $\Leftrightarrow$ (CI) in Theorem 4.2.

(ii) Let λ be an eigenvalue of $L(A)^{-1}L(B)$ whose negative reciprocal lies in the interior of T, and let m be its multiplicity. Let Γ be a disc with center at λ that contains no other eigenvalue of $L(A)^{-1}L(B)$, and such that the negative reciprocals of real numbers in Γ lie in T. Without loss of generality assume that $A + tB$ is positive semistable for every t in T. Since $L(A)$ is nonsingular, it follows that for all sufficiently small positive δ, $L(A + \delta I)$ is nonsingular. For such δ, $(A + \delta I) + tB$ is positive stable for all t in T. By Theorem 4.2, the operator $F(\delta) = L(A + \delta I)^{-1}L(B)$ has no eigenvalue with negative reciprocal in T. Since $F(\delta)$ is an operator on the *real* space $\mathscr{H}_n$, its complex eigenvalues appear in conjugate pairs. Consequently, $F(\delta)$ has an even number of eigenvalues in Γ. By Lemma 2.6(i) it now follows that the multiplicity of λ as an eigenvalue of $L(A)^{-1}L(B)$ is even.

(iii) Let δ be a positive number. If A and B are real then $C(\delta) = (A + \delta I)^{-1}B$ is real, and hence the complex eigenvalues of $C(\delta)$ appear in conjugate pairs. By Theorem 3.2, if $-1/t$ is an eigenvalue of $(A + \delta I)^{-1}B$, then $-1/t$ is an eigenvalue of $L(A + \delta I)^{-1}L(B)$. As in the proof of part (ii), for δ sufficiently small, $L(A + \delta I)^{-1}L(B)$ has no eigenvalue with negative reciprocal in T. Therefore, $(A + \delta I)^{-1}B$ has no eigenvalue with negative reciprocal in T. Since the complex eigenvalues of $C(\delta)$ appear in conjugate pairs, it follows that $C(\delta)$ has an even number of eigenvalues in Γ. By Lemma 2.6(i) it now follows that the multiplicity of λ as an eigenvalue of $A^{-1}B$ is even. □

Remark 4.11. In general, if $A + tB$ is near stable then the multiplicity of $-1/t$ as an eigenvalue of $L(A)^{-1}L(B)$ is not necessarily even. For example, take A to be an identity matrix of odd order, and let $B = A$. Then $L(A)^{-1}L(B)$ is an identity matrix of odd order and hence its only eigenvalue, 1, has odd multiplicity. Yet, $A - B$ is near stable.

Next we assume that all eigenvalues of $A + tB$ are real for t in an interval T. The following result is essentially due to Väliaho [7], where it is stated for Hermitian matrices. It is stated here for the sake of completeness.

THEOREM 4.12. *Let A and B be complex $n \times n$ matrices and assume that A is nonsingular. Assume that all eigenvalues of $A + tB$ are real for all $t \in \mathbb{R}$. Let $-1/t_i$, $i = 1, \cdots, m$, where $t_1 < \cdots < t_m$, be the distinct nonzero eigenvalues of $A^{-1}B$. Let $T_0 = (-\infty, t_1)$, $T_i = (t_i, t_{i-1})$, $i = 1, \cdots, m$, and $T_m = (t_m, \infty)$. Then the intervals T_i, $i = 0, \cdots, m$ are maximal intervals of constant inertia for $S(A, B; \mathbb{R})$ and the inertia of each matrix in $S(A, B; T_i)$ is of the form $(\pi_i, \nu_i, 0)$, $i = 0, \cdots, m$.*

Proof. As in Theorem 3.5, by standard results in analysis, T is an interval of constant inertia for $S(A, B; \mathbb{R})$ if and only if T contains no inertia change point for $S(A, B; \mathbb{R})$. By Theorem 3.3, if $-1/t$ is an eigenvalue of $A^{-1}B$, then t is an inertia change point for $S(A, B; \mathbb{R})$. Our claim now follows from Theorem 4.6. □

5. Stable convex hull of matrices. The results of the previous section can be applied in several directions. We conclude the paper by demonstrating a sample of such applications.

The following result was proved for real matrices in [1] and [2].

THEOREM 5.1. *Let A and B be $n \times n$ complex matrices. Then the convex hull* conv (A, B) *is stable if and only if A is stable and $L(A)^{-1}L(B)$ has no nonpositive real eigenvalue.*

Proof. If A is stable, it follows from the equivalence (LA) $\Leftrightarrow$ (CI) in Theorem 4.2, applied to the matrices A and $B - A$ and the interval $T = [0, 1]$, that conv (A, B) is stable if and only if $L(A)^{-1}L(B - A)$ has no real eigenvalue less than -1, which is equivalent to saying that $L(A)^{-1}L(B)$ has no nonpositive real eigenvalue. Since the stability of conv (A, B) of course implies that A is stable, the result now follows. □

THEOREM 5.2. *Let A and B be $n \times n$ complex matrices, and assume that all the matrices in* conv (A, B) *have Property X. Then the following are equivalent.*

(i) *The convex hull* conv (A, B) *is stable.*

(ii) *A is stable and $L(A)^{-1}L(B)$ has no nonpositive real eigenvalue.*

(iii) *A is stable and $A^{-1}B$ has no nonpositive real eigenvalue.*

Proof. Our claim follows from the equivalences (A) $\Leftrightarrow$ (LA) $\Leftrightarrow$ (CI) in Theorem 4.5, applied to the matrices A and $B - A$ and the interval $[0, 1]$. □

The following theorem is found in [4], where it is stated for Hermitian matrices.

THEOREM 5.3. *Let A and B be $n \times n$ complex matrices, and assume that all the matrices in* conv (A, B) *have all eigenvalues real. Then the following are equivalent.*

(i) *A is nonsingular and $A^{-1}B$ has no nonpositive real eigenvalue.*

(ii) *All matrices in* conv (A, B) *are nonsingular.*

(iii) conv (A, B) *has constant inertia of type* $(\pi, \nu, 0)$.

Proof. (i) $\Leftrightarrow$ (ii) follows from the equivalence of conditions (A_1) and (A_2) applied to the matrices A and $B - A$ and the interval $T = [0, 1]$.

(ii) $\Rightarrow$ (iii) by Lemma 2.6 (iii), since all matrices in conv (A, B) have all eigenvalues real.

(iii) $\Rightarrow$ (ii) is trivial. □

We end with an example that illustrates Theorem 5.2 and the analogue for the convex hulls of Theorem 4.10(iii).

Example 5.4. Let

$$A = \begin{pmatrix} 2 & -3 \\ -1 & 2 \end{pmatrix} = B^T.$$

Then it is easy to show that all matrices in conv (A, B) are M-matrices and hence are semistable. Furthermore, each matrix in conv (A, B) is stable, except for $(A + B)/2$. Note that -1 is an eigenvalue of $A^{-1}B$ of multiplicity two. For every positive ε, conv $(A + \varepsilon I, B)$ is stable, and hence, by Theorem 5.2, $(A + \varepsilon I)^{-1}B$ has no nonpositive real eigenvalue. Indeed, the eigenvalues of $(A + .1I)^{-1}B$ are approximately $-.8 \pm .8775i$. Note also that every matrix in conv $(A + .1I, B)$ has Property X, but $(A + .1I)^{-1}B$ does not.

Acknowledgment. The authors are grateful to Professor David Carlson for his helpful comments, which helped to improve the presentation of the paper.

REFERENCES

[1] S. BIALAS, *A necessary and sufficient condition for the stability of convex combinations of stable polynomials or matrices*, Bull. Polish Acad. Sci. Tech. Sci., 33 (1985), pp. 473–480.

[2] M. FU AND B. R. BARMISH, *A generalization of Kharitonov's polynomial framework to handle linearly independent uncertainty*, Tech. Report ECE-87-9, Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI, 1987.

[3] ———, *Maximal undirectional perturbation bounds for stability of polynomials and matrices*, Systems Control Lett., 11 (1988), pp. 173–178.

[4] C. R. JOHNSON AND L. RODMAN, *Convex sets of Hermitian matrices with constant inertia*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 351–359.

[5] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, Second Edition, Academic Press, New York, 1985.

[6] H. SCHNEIDER, *Topological aspects of Sylvester's theorem on the inertia of Hermitian matrices*, Amer. Math. Monthly, 73 (1966), pp. 817–821; Selected Papers Algebra, The Mathematical Association of America, Washington, DC, 1977, pp. 339–343.

[7] H. VÄLIAHO, *Determining the inertia of a matrix pencil as a function of the parameter*, Linear Algebra Appl., 106 (1988), pp. 245–258.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 575–588, October 1990

008

AN IMPROVED METHOD FOR ONE-WAY DISSECTION WITH SINGULAR DIAGONAL BLOCKS*

JESSE L. BARLOW† AND UDAYA B. VEMULAPATI‡

Abstract. Matrices arising out of the one-way dissection method for solving large sparse systems of linear equations are considered. The systems that are considered are those that may have singular diagonal blocks. Such systems arise in certain fluid flow problems.

Gunzberger and Nicholaides [*Linear Algebra Appl.*, 64 (1985), pp. 183–189] proposed a method for resolving the singularity in the diagonal blocks. This method uses the Moore–Penrose pseudoinverse. Two improvements to the Gunzberger–Nicholaides procedure are proposed: (1) The substitution of a weighted pseudoinverse for the Moore–Penrose pseudoinverse; (2) A more elegant implementation of the back substitution procedure. A stability analysis of both the Barlow–Vemulapati and the Gunzberger–Nicholaides procedures is given. Both analysis and empirical tests show that the former method has better numerical stability properties than the Gunzberger–Nicholaides procedure. The Barlow–Vemulapati procedure is also implemented on the Intel iPSC/1 Hypercube. The improvement to the back substitution method makes the natural parallelism in the problem easier to exploit.

Key words. weighted pseudoinverse, parallel processing, error analysis

AMS(MOS) subject classifications. 65F05, 65F20, 65F25

1. Introduction. One-way dissection is a common technique for the solution of large sparse systems of linear equations (cf. [8], [11]). In the literature on the numerical solution of partial differential equations, it has also been called substructuring [4] and domain decomposition [6]. However, such systems also arise in other applications, for example, in economic models [7].

The basic problem is to solve the $n \times n$ system of linear equations

$$Ax = s, \tag{1.1}$$

where A and s have the form

$$A = \begin{bmatrix} B_1 & 0 & 0 & \cdot & S_1 \\ 0 & B_2 & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & B_k & S_k \\ G_1^T & G_2^T & \cdot & G_k^T & F \end{bmatrix}; \qquad s = (s_1, s_2, \cdots, s_k, s_{k+1})^T. \tag{1.2}$$

Here B_i, $i = 1, 2, \cdots, k$ are $m_i \times m_i$ matrices, F is a $p \times p$ matrix, and G_i and S_i are $m_i \times p$ matrices, where $p + \sum_{i=1}^{k} m_i = n$. Each s_i, $i = 1, 2, \cdots, k$ is an m_i-vector and s_{k+1} is a p-vector. Much of the discussion of one-way dissection in the literature has

* Received by the editors April 4, 1988; accepted for publication (in revised form) September 6, 1989.

† Department of Computer Science, Pennsylvania State University, University Park, Pennsylvania 16802 (barlow@shire.cs.psu.edu). The research of this author was supported by the National Science Foundation under grant CCR-8700172, the Air Force Office of Scientific Research under grant AFOSR-88-0161, the Office of Naval Research under grant N0014-80-0517, and the Applied Mathematical Sciences Research Program of the Office of Energy Research, U.S. Department of Energy, under contract DE-AC05-040R214000 with Martin Marietta Energy Systems, Inc. This research was done in part while the author was visiting Oak Ridge National Laboratory.

‡ Department of Computer Science, Pennsylvania State University, University Park, Pennsylvania 16802 (vemula@na-net.stanford.edu). The research of this author was supported by the Office of Naval Research under grant N00024-85-C-6041.

concerned symmetric, positive definite systems. This implies that B_i, $i = 1, 2, \cdots, k$ and F are symmetric, positive definite, and $G_i = S_i$, $i = 1, 2, \cdots, k$. Instead we make the much weaker assumption that rank $(A) = n$, i.e., that A is nonsingular. Thus we have that rank $(B_i) = l_i \leqq m_i$, $i = 1, 2, \cdots, k$. Applications of such systems are given in [14].

Gunzberger and Nicholaides [13] suggested an algorithm based on Gaussian elimination with singular pivots. It uses the Moore–Penrose inverses of the diagonal blocks B_i, $i = 1, 2, \cdots, k$. The Moore–Penrose pseudoinverse of a matrix B, denoted by B^+, is the unique matrix satisfying the four Penrose conditions:

$$\begin{aligned} &(1)\ BB^+B = B, \\ &(2)\ B^+BB^+ = B^+, \\ &(3)\ (BB^+)^T = BB^+, \\ &(4)\ (B^+B)^T = B^+B. \end{aligned} \tag{1.3}$$

We will use the notation $B^{(i)}$, $B^{(i,j)}$, or $B^{(i,j,k)}$ to denote matrices satisfying conditions i, j, or k among those in (1.3). The procedure in [13] has a simple elimination procedure, but a complicated back substitution procedure.

In this paper, we suggest an alternative method for resolving the singularity in the diagonal blocks B_i, $i = 1, 2, \cdots, k$. This method is based on the weighted pseudoinverse discussed in a fundamental paper by Elden [9]. We give evidence that this method is more stable. We also give a more elegant back substitution procedure, which makes the algorithm easier to implement on a message-passing architecture. These algorithms are outlined in § 2. An error analysis of our proposed algorithm is given in § 3. Empirical tests verifying the stability properties of our algorithm are given in § 4. We also give an implementation on Intel Hypercube(iPSC/1) in § 4. The implementation is very straightforward.

2. Description of algorithms. We first describe the elimination procedure of Gunzberger and Nicholaides [13] for solving (1.1). It makes use of the Moore–Penrose pseudoinverses of the diagonal blocks B_i, $i = 1, 2, \cdots, k$. The other elimination procedures in this section will take a similar form.

ALGORITHM 1. *Block elimination using the Moore–Penrose pseudoinverse* [13].
(1) *Compute*

$$\tilde{F} = F - \sum_{i=1}^{k} G_i^T B_i^+ S_i; \qquad \tilde{s}_{k+1} = s_{k+1} - \sum_{i=1}^{k} G_i^T B_i^+ s_i;$$

$$\tilde{G}_i^T = G_i^T (I - B_i^+ B_i) \quad \textit{projection of } G_i^T \textit{ onto } \text{Range}\,(B_i).$$

(2) *For $i = 1, 2, \cdots, k$ find an $m_i \times (m_i - l_i)$ matrix X_i such that $B_i X_i = 0$. Note that $l_i = \text{rank}\,(B_i)$. Thus X_i is a basis for the null-space of B_i. Algorithms for finding such a basis are given by Heath* [15] *and Pothen* [17].

We note that the terms $G_i^T B_i^+ S_i$, $G_i^T B_i^+ s_i$, $i = 1, 2, \cdots, k$ can be computed independently, as can the null-space bases X_i, $i = 1, 2, \cdots, k$. The same is true for the $\tilde{G}_i^T$, $i = 1, 2, \cdots, k$, but we will see later that it is not necessary to compute these matrices at all.

The back substitution phase of the Gunzberger–Nicholaides procedure is somewhat complicated. Let $x = (x_1, x_2, \cdots, x_k, x_{k+1})^T$, where the first k block components x_i, $i = 1, 2, \cdots, k$ are of the form

$$x_i = y_i + z_i \tag{2.1}$$

where

$$y_i^T z_i = 0, \qquad i = 1, 2, \cdots, k,$$

and the vectors z_i satisfy

$$B_i z_i = 0, \qquad i = 1, 2, \cdots, k, \tag{2.2a}$$

$$\tilde{F} z_{k+1} = 0. \tag{2.2b}$$

Since A is nonsingular, $\tilde{F}$ is also nonsingular (cf. [13]). Thus

$$z_{k+1} = 0, \tag{2.3a}$$

$$x_{k+1} = y_{k+1}. \tag{2.3b}$$

Then Algorithm 1 reduces (1.1) to the system

$$B_i y_i + S_i x_{k+1} = s_i, \tag{2.4a}$$

$$\sum_{i=1}^{k} \tilde{G}_i^T z_i + \tilde{F} x_{k+1} = \tilde{s}_{k+1}. \tag{2.4b}$$

From (2.2a), $\tilde{G}_i^T z_i = G_i^T z_i$, $i = 1, 2, \cdots, k$; thus we can replace (2.4b) with

$$\sum_{i=1}^{k} G_i^T z_i + \tilde{F} x_{k+1} = \tilde{s}_{k+1}. \tag{2.5}$$

Thus $\tilde{G}_i^T$ need never be explicitly computed. The system (2.4) can be written

$$My = \tilde{s} - Nz, \tag{2.6}$$

where

$$M = \begin{bmatrix} B_1 & 0 & \cdot & 0 & S_1 \\ 0 & B_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & B_k & S_k \\ 0 & \cdot & \cdot & 0 & \tilde{F} \end{bmatrix} \tag{2.7a}$$

$$N = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & S_1 \\ 0 & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & S_k \\ G_1^T & G_2^T & \cdot & G_k^T & 0 \end{bmatrix} \tag{2.7b}$$

$$\tilde{s} = (s_1, s_2, \cdots, s_k, \tilde{s}_{k+1})^T;$$

$$y = (y_1, \cdots, y_k, y_{k+1})^T; \tag{2.7c}$$

$$z = (z_1, \cdots, z_k, z_{k+1})^T.$$

The consistency of (2.6) and the nonsingularity of $\tilde{F}$ follow from the nonsingularity of A. If we assume that z is known, and let

$$f=(f_1,f_2,\cdots,f_k,f_{k+1})^T=\tilde{s}-Nz, \tag{2.8}$$

then a basic (nonunique) solution y is given by

$$y_{k+1}=x_{k+1}=\tilde{F}^{-1}f_{k+1}, \tag{2.9a}$$

$$y_i=B_i^+(f_i-S_iy_{k+1}). \tag{2.9b}$$

From [13], we have that y solves (2.6). Thus if we define the matrix Φ such that

$$y=\Phi f \tag{2.10}$$

where Φ has the form

$$\Phi=\begin{bmatrix} B_1^+ & 0 & \cdot & \cdot & -B_1^+S_1\tilde{F}^{-1} \\ \cdot & B_2^+ & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & B_k^+ & -B_k^+S_k\tilde{F}^{-1} \\ 0 & \cdot & \cdot & 0 & \tilde{F}^{-1} \end{bmatrix}, \tag{2.11}$$

we note the following fact about Φ. Its proof is obvious.

LEMMA 2.1. $\Phi = M^{(1,2,4)}$. *That is,* Φ *is a* (1, 2, 4)*-pseudoinverse of* M.

If we combine (2.6) and (2.10) we have

$$(I-M\Phi)Nz=(I-M\Phi)\tilde{s}. \tag{2.12}$$

Let

$$X=\operatorname{diag}(X_1,X_2,\cdots,X_k,0).$$

Thus (2.12) becomes

$$Tw=g, \tag{2.13}$$

where $T = (I - M\Phi)NX$; $z = Xw$; $g = (I - M\Phi)\tilde{s}$. Equation (2.13) is consistent but overdetermined (cf. [13]). It can be solved by orthogonal factorization of T (in [13], the use of normal equations is advocated). We assume that the dimensions of the null-spaces of B_i, $i = 1, 2, \cdots, k$ are much smaller than the dimensions of the blocks themselves. That is, that $m_i - l_i \ll m_i$. Thus the solution of (2.13) should be very fast compared with the rest of the algorithm. We state the procedure as Algorithm 2.

ALGORITHM 2. *Back substitution procedure* [13].

(1) *Explicitly form* $T = (I - M\Phi)NX$; $g = (I - M\Phi)\tilde{s}$.

(2) *Solve* $Tw = g$ *by orthogonal factorization* (*or normal equations*).

(3) *Let* $z = Xw$ *and solve*

$$y=\Phi(\tilde{s}-Nz).$$

(4) *The solution* $x = y + z$.

We propose two changes in Algorithms 1–2. The first is a simplification of the back substitution procedure. This simplification uses computations arising directly out of the elimination procedure. To describe that, we give a more specific version of Algorithm 1 that includes the method for computing B_i^+, $i = 1, 2, \cdots, k$. The method is slightly

different from that given in [13], but uses the method for computing B_i^+ given in Golub and Van Loan [12, pp. 162–167].

ALGORITHM 3. *Implementation of block elimination using the Moore–Penrose pseudoinverse.*

(1) *For* $i = 1, 2, \cdots, k$ *perform steps* 2–7.

(2) *Factor* B_i *into*

$$B_i = Q_i \begin{bmatrix} U_i^{[1]} & U_i^{[2]} \\ 0 & 0 \end{bmatrix} P_i^T,$$

where Q_i *is orthogonal,* $U_i^{[1]}$ *is an* $l_i \times l_i$ *upper triangular,* $U_i^{[2]}$ *is an* $l_i \times (m_i - l_i)$ *matrix, and* P_i *is a permutation matrix. This factorization and the determination of* rank l_i *can be done by orthogonal decomposition with column pivoting (cf.* [16, *Chap.* 10]) *or some other method (cf.* [2], [5], [10]).

(3) *Compute* $(\hat{S}_i, \hat{s}_i)$ *satisfying*

$$\begin{bmatrix} S_i^{[1]} & s_i^{[1]} \\ S_i^{[2]} & s_i^{[2]} \end{bmatrix} = Q_i^T (S_i, s_i),$$

where $S_i^{[1]}$ *is* $l_i \times p$ *and* $S_i^{[2]}$ *is* $(m_i - l_i) \times p$.

(4) *Compute*

$$U_i^{[1]} (\hat{S}_i, \hat{s}_i) = (S_i^{[1]}, s^{[1]}).$$

(5) *Compute*

$$X_i = \begin{bmatrix} -[U_i^{[1]}]^{-1} U_i^{[2]} \\ I_{m_i - l_i} \end{bmatrix}. \tag{2.14}$$

X_i is a common choice for the null basis matrix of B_i (cf. [15], [17]).

(6) *Factor*

$$X_i = Z_i \begin{bmatrix} W_i \\ 0 \end{bmatrix},$$

where Z_i is orthogonal and W_i is upper triangular and compute

$$(V_i, v_i) = Z_i \begin{bmatrix} 0 & 0 \\ 0 & I_{p - m_i + l_i} \end{bmatrix} Z_i^T (\hat{S}_i, \hat{s}_i).$$

(7) *Compute*

$$(R_i, r_i) = -G_i^T (V_i, v_i).$$

(8) *Compute*

$$\tilde{F} = F + \sum_{i=1}^{k} R_i; \tilde{s}_{k+1} = s_{k+1} + \sum_{i=1}^{k} r_i.$$

Algorithm 3 requires

$$2m_il_i(m_i-l_i)+\frac{2}{3}l_i^3+2m_il_i(p+1)+l_i^2(m_i-l_i)+(m_i-l_i)^2\left(m_i-\frac{1}{3}(m_i+l_i)\right)$$
$$+4(m_i-l_i)m_i(p+1)+p(p+1)l_i+O(m^2)$$

flops for each $i = 1, 2, \cdots, k$. Here $m = \max_{1 \leq i \leq k} m_i$. If $|m_i - l_i| \leqq c = O(1)$, this simplifies to

$$2m_il_i(p+1)+\frac{2}{3}l_i^3+p(p+1)l_i+O(cm^3)$$

for each $i = 1, 2, \cdots, k$. We assume here that all of the blocks in (1.2) are dense.

If we consider (2.4a) and apply the reduction from Algorithm 3, we have

$$U_iy_i+S_i^{[1]}x_{k+1}=s_i^{[1]}, \tag{2.15a}$$

$$S_i^{[2]}x_{k+1}=s_i^{[2]}, \tag{2.15b}$$

where $U_i = (U_i^{[1]}, U_i^{[2]})$. Since (2.15) is just an orthogonal reduction of some rows from $Ax = s$, it follows that it is underdetermined but consistent. Using the null-basis (2.14) for B_i and by letting

$$\hat{G}_i = G_i^T X_i,$$

(2.5) becomes

$$\sum_{i=1}^{k} \hat{G}_i w_i + \tilde{F} x_{k+1} = \tilde{s}_{k+1}, \tag{2.16}$$

where $z_i = X_i w_i$. Thus if we let $S^{[2]} = (S_1^{[2]}, \cdots, S_k^{[2]})^T$, $s^{[2]} = (s_1^{[2]}, \cdots, s_k^{[2]})^T$, and $\hat{G} = (\hat{G}_1, \cdots, \hat{G}_k)$, then x_{k+1} and $w = (w_1, w_2, \cdots, w_k)^T$ solve the linear system

$$\begin{bmatrix} \hat{G} & \tilde{F} \\ 0 & S^{[2]} \end{bmatrix} \begin{bmatrix} w \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} \tilde{s}_{k+1} \\ s^{[2]} \end{bmatrix}. \tag{2.17}$$

The nonsingularity of A guarantees that (2.15) is a nonsingular system of linear equations. For problems arising in practice, its dimension will be small compared to the dimension of A. It can be solved by Gaussian elimination with partial pivoting or orthogonal decomposition. Such a reduction is much simpler than the back substitution procedure in Algorithm 2. The values of y_i and x_i, $i = 1, 2, \cdots, k$ can be recovered from (2.9a) and the step

$$x_i = y_i + X_i w_i. \tag{2.18}$$

The computation (2.9a) can be simplified into

$$y_i = [U]_i^+ (s_i^{[1]} - S_i^{[1]} x_{k+1}), \tag{2.19}$$

thereby avoiding the reuse of the orthogonal factor Q_i. We now formally state this procedure as Algorithm 4. This algorithm is a method for solving (2.5) and is simply a particular implementation of Algorithm 2.

ALGORITHM 4. *Improved back substitution procedure.*

(1) *Solve the linear system in* (2.17) *for* x_{k+1} *and* $w = (w_1, w_2, \cdots, w_k)^T$ *using orthogonal factorization by Householder transformations.*

(2) *For* $i = 1, 2, \cdots, k$ *do steps* 3–6.

(3) *Compute*

$$f_i^{[1]} = s_i^{[1]} - S_i^{[1]} x_{k+1}. \tag{2.20}$$

(4) *Let*

$$g_i = -\begin{bmatrix} I_{l_i} \\ 0 \end{bmatrix} [U_i^{[1]}]^{-1} f_i^{[1]} \tag{2.21}$$

$$\hat{g}_i = (I_{m_i - l_i}, 0) Z_i^T g_i.$$

(5) *Solve*

$$\begin{bmatrix} U_i^{[1]} & U_i^{[2]} \\ 0 & W_i \end{bmatrix} y_i = \begin{bmatrix} f_i^{[1]} \\ \hat{g}_i \end{bmatrix}.$$

Here $S_i^{[1]}$, $s_i^{[1]}$, $G_i^{[1]}$, $U_i^{[1]}$, W_i, *and* Z_i *are from Algorithm* 3.

(6) *Compute*

$$x_i = y_i + X_i w_i$$

where X_i *is in Algorithm* 3.

The back substitution procedure requires

$$\frac{2}{3}\left[p + \sum_{i=1}^{k} (m_i - l_i)\right]^3 + \sum_{i=1}^{k} \left[3l_i(m_i - l_i) + \frac{1}{2}m_i^2 + \frac{1}{2}l_i^2\right] + O(m)$$

flops. If $\max_{1 \le i \le k} |m_i - l_i| = c = O(1)$ then this reduces to

$$\frac{2}{3}\left[p + kc\right]^3 + \frac{1}{2}\sum_{i=1}^{k} [l_i^2 + m_i^2] + O(cm)$$

flops.

The second modification to Algorithms 1–2 is to replace B_i^+ with $B_i^{(1,3)}$, $i = 1, 2, \cdots, k$, i.e., any matrix $B_i^{(1,3)}$ satisfying Penrose conditions 1 and 3. For the elimination algorithm, this is equivalent to solving (cf. [9])

$$\min_{(V_i, v_i)} \| B_i(V_i, v_i) - (S_i, s_i) \|_F$$

and then computing

$$\tilde{F} = F - \sum_{i=1}^{k} G_i^T V_i, \tag{2.22a}$$

$$\tilde{s}_{k+1} = s_{k+1} - \sum_{i=1}^{k} G_i^T v_i, \tag{2.22b}$$

$$\tilde{G}_i^T = G_i^T (I - B_i^{(1,3)} B_i). \tag{2.22c}$$

It is essential that all of the columns of

$$(H_i, h_i) = (S_i, s_i) - B_i(V_i, v_i) \tag{2.23}$$

be vectors in the space orthogonal to the columns of B_i. It is guaranteed by the use of $B_i^{(1,3)}$. This allows us to set up (2.17) by orthogonal factorization of B_i by column pivot-

ing or some other method to detect rank (e.g., [2], [5], [10]). When we substitute $B_i^{(1,3)}$ for B_i^+, we lose the property that $y_i^T z_i = 0$, but this property is not necessary for the algorithm to work. Again since $\tilde{G}_i^T z_i = G_i^T z_i$, it is not necessary to do the computation (2.22c).

The matrix $B_i^{(1,3)}$ is not unique unless B_i has full rank. In our modified algorithm, we can choose $B_i^{(1,3)}$ so as to minimize $\|G_i^T B_i^{(1,3)} S_i\|_F$ and $\|G_i^T B_i^{(1,3)} s_i\|_2$. As will be shown in the next section, this leads to a new algorithm with better numerical stability properties. Elden [9] showed that the (1, 3) pseudoinverse with this property is the weighted pseudoinverse defined below.

DEFINITION 2.1. The G-weighted pseudoinverse of B is defined by

$$B_G^+ = (I - (G^T P)^+ G^T) B^+,$$

where

$$P = I - B^+ B.$$

In [9], it is shown that the matrix B_G^+ is the (1, 3)-inverse such that

$$\|G^T B_G^+ S\|_F \leqq \|G^T B^{(1,3)} S\|_F \tag{2.24}$$

for all (1, 3)-inverses of B and matrices S. The G-weighted pseudoinverses $[B_G]_i^+$ need not and should not be explicitly computed. Instead we compute the quantities

$$R_i = -G_i^T [B_G]_i^+ S_i \qquad i = 1, 2, \cdots, k \tag{2.25a}$$

$$r_i = -G_i^T [B_G]_i^+ s_i \qquad i = 1, 2, \cdots, k \tag{2.25b}$$

and then compute

$$\tilde{F} = F + \sum_{i=1}^{k} R_i; \qquad \tilde{s}_{k+1} = s_{k+1} + \sum_{i=1}^{k} r_i. \tag{2.26}$$

The quantities (R_i, r_i) are simply the residuals of the least squares problem

$$\min_{(V_i, v_i) \in T_{B_i}} \|G_i^T (V_i, v_i)\|_F, \tag{2.27}$$

where T_{B_i} is the set of minimizers of

$$\min_{(V_i, v_i) \in \mathbb{R}^{m_i \times (p+1)}} \|B_i (V_i, v_i) - (S_i, s_i)\|_F.$$

The computation of (V_i, v_i) is not necessary. The residuals (R_i, r_i) can be computed directly. Problem (2.27) has a unique solution if

$$\operatorname{rank} \begin{bmatrix} B_i \\ G_i^T \end{bmatrix} = m_i, i = 1, 2, \cdots, k.$$

This is a direct consequence of nonsingularity of A. We now give a more detailed description of this procedure. Steps 1–4 are the Björck–Golub (cf. [3]) direct elimination procedure for solving (2.27).

ALGORITHM 5. *Block elimination scheme using the weighted pseudoinverse.*

(1) *For* $i = 1, 2, \cdots, k$ *do steps* 2–5.

(2) *Same as steps* 2–3 *of Algorithm* 3.

(3) *Let* $G_i^T = (G_i^{[1]}, G_i^{[2]})$, *where* $G_i^{[1]}$ *is a* $p \times l_i$ *matrix and* $G_i^{[2]}$ *is a* $p \times (m_i - l_i)$ *matrix. Compute* $\hat{G}_i = G_i^{[2]} - G_i^{[1]}[U_i^{[1]}]^{-1}U_i^{[2]}$ *and* $(\hat{S}_i, \hat{s}_i) = -G_i^{[1]}[U_i^{[1]}]^{-1}(S_i, s_i)$. (*Note that* $\hat{G}_i = G_i^T X_i$.)

(4) *Factor*

$$\hat{G}_i = Z_i \begin{bmatrix} W_i \\ 0 \end{bmatrix},$$

where Z_i *is orthogonal and* W_i *is upper triangular. Then compute*

$$(R_i, r_i) = Z_i \begin{bmatrix} 0 & 0 \\ 0 & I_{p-n_i+l_i} \end{bmatrix} Z_i^T (\hat{S}_i, \hat{s}_i).$$

(5) *Compute*

$$\tilde{F} = F + \sum_{i=1}^{k} R_i; \qquad \tilde{s}_{k+1} = s_{k+1} + \sum_{i=1}^{k} r_i.$$

With the change that

$$g_i = -G_i^{[1]}[U_i^{[1]}]^{-1} f_i^{[1]} \tag{2.28}$$

in (2.21), the back substitution procedure in Algorithm 4 can be used directly after Algorithm 5. This adds an additional $l_i p$ flops for each $i = 1, 2, \cdots, k$. Except for differences in terms of $O(m^2)$, the operation count for Algorithm 5 is identical to that of Algorithm 3. We note, however, one difference that the matrix

$$\tilde{\Phi} = \begin{bmatrix} B_1^{(1,3)} & 0 & \cdot & \cdot & -B_1^{(1,3)} S_1 \tilde{F}^{-1} \\ \cdot & B_2^{(1,3)} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & B_k^{(1,3)} & -B_k^{(1,3)} S_k \tilde{F}^{-1} \\ 0 & \cdot & \cdot & 0 & \tilde{F}^{-1} \end{bmatrix}$$

is only a (1)-pseudoinverse of M. This can be verified easily. However, this is enough to assure that $y = \tilde{\Phi} f$ satisfies (2.10). Hence we can use the back substitution procedure in Algorithm 4.

The stability properties of these direct elimination procedures can be shown using well known properties of methods for solving constrained least squares problems and systems of linear equations (cf. [12], [18]). These properties are given in the next section.

3. Error analysis of the revised algorithms. We now use backward error analysis (cf. [18]) to bound the errors in the computational versions of the algorithms in § 1. The general form of the reductions are

$$\tilde{F} = F - \sum_{i=1}^{k} G_i^T B_i^{(1,3)} S_i; \qquad \tilde{s}_{k+1} = s_{k+1} - \sum_{i=1}^{k} G_i^T B_i^{(1,3)} s_i, \tag{3.1}$$

where $B_i^{(1,3)}$ is a (1, 3) pseudoinverse of B_i. First we need the following lemma from [1] on the Björck–Golub direct elimination procedure, as applied in steps 1–4 of Algorithm 3.

LEMMA 3.1. *Let steps* 2–5 *of Algorithm* 5 *be implemented using Householder transformations in floating-point arithmetic with machine unit* μ. *Then the factorization of*

each of the blocks $\begin{bmatrix} B_i \\ G_i^T \end{bmatrix}$, $i = 1, 2, \cdots, k$ *satisfy*

$$\begin{bmatrix} B \\ G^T \end{bmatrix} = Y \begin{bmatrix} U_1 & U_2 \\ 0 & 0 \\ 0 & W \\ 0 & 0 \end{bmatrix} P^T + \begin{bmatrix} \delta B \\ \delta G^T \end{bmatrix}, \tag{3.2a}$$

$$(S, s) = Y \begin{bmatrix} \hat{S}^{[1]} & \hat{s}^{[1]} \\ \hat{S}^{[2]} & \hat{s}^{[2]} \end{bmatrix} + (\delta S, \delta s), \tag{3.2b}$$

where $Y = Q_1 L Q_2$, Q_1 *and* Q_2 *are orthogonal, and* L *is unit lower triangular. The backward errors* δB, δG, δS, *and* δs *satisfy*

$$\|\delta B\|_F \leqq \phi_B \|B\|_F \mu + O(\mu^2), \tag{3.3a}$$

$$\|\delta G\|_F \leqq \phi_G \tau_G \left\| \begin{pmatrix} B \\ G^T \end{pmatrix} \right\|_F \mu + O(\mu^2), \tag{3.3b}$$

$$\|(\delta S, \delta s)\|_F \leqq \phi_S \tau_S \|(S, s)\|_F \mu + O(\mu^2), \tag{3.3c}$$

where ϕ_B, ϕ_G, *and* ϕ_S *are modestly sized polynomials in the dimension of* B, G, *and* S *and*

$$\tau_G = \max \{ \|[U^{[1]}]^{-1} U^{[2]}\|_F, \max_{1 \leq q \leq l} 1 + \|[U_{(q)}^{[1]}]^{-1} U_{(q)}^{[2]}\|_F \} \tag{3.4}$$

$$\tau_S = \|G^{[1]}[U^{[1]}]^{-1}\|_2, \tag{3.5}$$

where $U_{(q)} = (U_{(q)}^{[1]}, U_{(q)}^{[2]})$ *is the first* q *rows of* U, $U_{(q)}^{[1]}$ *is a* $q \times q$ *nonsingular matrix, and* $U_{(q)}^{[2]}$ *is a* $q \times (n - q)$ *matrix,* $q = 1, 2, \cdots, l$.

A backward error analysis of the reduction stage of Algorithm 5 can be obtained from this theorem by substituting I for G^T.

Let $(\hat{R}_i, \hat{r}_i)$, $i = 1, 2, \cdots, k$, $\hat{F}$, and $\hat{s}_{k+1}$ be the computed values of (R_i, r_i), $\tilde{F}$, and $\tilde{s}_{k+1}$ from Algorithm 3 or 5. Then we have the computational equations

$$(\hat{R}_i, \hat{r}_i) = (G_i + \delta G_i)^T (B_i + \delta B)^{(1,3)} (S_i + \delta S_i, s_i + \delta s_i),$$

$$\tilde{F} = F + \delta F + \sum_{i=1}^{k} \hat{R}_i = fl\left(F + \sum_{i=1}^{k} \hat{R}_i \right),$$

$$\tilde{s}_{k+1} = s_{k+1} + \delta s_{k+1} + \sum_{i=1}^{k} \hat{r}_i = fl\left(s_{k+1} + \sum_{i=1}^{k} \hat{r}_i \right),$$

where δG_i, δB_i, $(\delta S_i, \delta s_i)$ are errors that can be bounded by Lemma 3.1 and $fl(\cdot)$ denotes the floating-point computation of the contents. The errors δF and δs_{k+1} are just the errors in the floating-point sums. Thus from standard bounds on errors in sums we have

$$\|\delta F\|_F \leqq \phi_F \mu \max \{ \|F\|_F, \max_{1 \leq i \leq k} \|R_i\|_F \} + O(\mu^2),$$

$$\|\delta s_{k+1}\|_2 \leqq \phi_S \mu \max \{ \|s_{k+1}\|_2, \max_{1 \leq i \leq k} \|r_i\|_2 \} + O(\mu^2),$$

where ϕ_F and ϕ_S are modestly sized polynomials in the dimensions of A. From the definition of R_i and r_i we have

$$R_i = G_i^T B_i^{(1,3)} S_i; \qquad r_i = G_i^T B_i^{(1,3)} s_i \qquad i = 1, 2, \cdots, k,$$

thus

$$\|R_i\|_F \leqq \|G_i^T B_i^{(1,3)}\|_2 \|S_i\|_F, \tag{3.6}$$

$$\|r_i\|_2 \leqq \|G_i^T B_i^{(1,3)}\|_2 \|s_i\|_2. \tag{3.7}$$

Using the fact that

$$\begin{bmatrix} [U_i^{[1]}]^{-1} \\ 0 \end{bmatrix} = U_i^{(1,2,3)}$$

is a (1, 2, 3) pseudoinverse of U_i and (2.24), we have that

$$\|G_i^T [B_G^+]_i\|_2 \leqq \|[G_i^{[1]}][U_i^{[1]}]^{-1}\|_2. \tag{3.8}$$

This gives us the following bound for Algorithm 5. Note that if the Moore–Penrose pseudoinverse B_i^+ is substituted for $[B_G^+]_i$, the inequality (3.8) may be false since the Moore–Penrose pseudoinverse does not satisfy (2.24). The following theorem summarizes our results.

THEOREM 3.1. *Let Algorithm* 3 *or* 5 *be implemented using Householder transformations in floating-point arithmetic with machine unit* μ. *Let the backward substitution phase be done using Algorithm* 4. *Then the computed solution* $\bar{x}$ *satisfies*

$$(A + \delta A)\bar{x} = s + \delta s,$$

where

$$\|\delta A\|_F \leqq \phi_A \tau_A \|A\|_F \mu + O(\mu^2),$$

$$\|\delta s\|_2 \leqq \phi_s \tau_A \|s\|_2 \mu + O(\mu^2),$$

$$\tau_A = \max\{\max_{1 \leqq i \leqq k} \|G_i^{[1]}[U_i^{[1]}]^{-1}\|_2, \max_{1 \leqq i \leqq k} \|G_i B_i^{(1,3)}\|_2\},$$

and ϕ_A *and* ϕ_s *are modestly sized polynomials in the dimension of* A.

We now give a corollary that gives stronger stability results for Algorithm 5. It is a straightforward consequence of Theorem 3.1 and equation (3.8).

COROLLARY 3.1. *Let Algorithm* 5 *be implemented using Householder transformations in floating-point arithmetic with machine unit* μ. *Then* δA *and* δs *in Theorem* 3.1 *satisfy*

$$\|\delta A\|_F \leqq \phi_A \tau_G \|A\|_F \mu + O(\mu^2),$$

$$\|\delta s\|_2 \leqq \phi_s \tau_G \|s\|_2 \mu + O(\mu^2),$$

where

$$\tau_G = \max_{1 \leqq i \leqq k} \|G_i^{[1]}[U_i^{[1]}]^{-1}\|_2$$

and ϕ_A *and* ϕ_S *are modestly sized polynomials in the dimension of* A.

The bound τ_G arise out of the Björck–Golub procedure. The factors $\|G_i^T[B_G^+]_i\|_2$, $i = 1, 2, \cdots, k$ arise out of the condition of each of the problems of the form (2.27). We note that the bound in Corollary 3.1 is smaller than that in Theorem 3.1. The Moore–Penrose inverse does not satisfy the inequality (3.8) and we know of no error bound as good as that in Corollary 3.1 for Algorithm 3.

Thus the error bounds obtained by this analysis are better for Algorithm 5 than for Algorithm 3. In the next section, we give numerical tests that seem to indicate that Algorithm 5 will give more reliable answers.

4. Tests and conclusions.

4.1. Stability tests. We implemented Algorithms 3 and 5 in FORTRAN single precision on the SUN3 with the back substitution procedure in Algorithm 4. The two algorithms differ only in their computation of $B_i^{(1,3)}$, $i = 1, 2, \cdots, k$.

The matrix A is generated randomly. Rank one singularities are introduced into each diagonal block by replacing the last row of each such block by the sum of its other rows. Then the right-hand side is formed by making the known solution vector $(1, 1, \cdots, 1)^T$. We then calculated the relative error in the solution. The results are shown in Table 1. Here the experiments clearly suggest that Algorithm 5 has better numerical stability properties than Algorithm 3. Thus we see that the use of the weighted pseudoinverse rather than the Moore–Penrose pseudoinverse gives us a better method of resolving the singularity in the diagonal blocks.

4.2. Hypercube implementation. To simplify the implementation on a Hypercube, it is assumed that each diagonal block B_i and F are of equal size, i.e., $m_i = p$, $i = 1, 2, \cdots, k$, and that $p = k + 1$, i.e., the size of each diagonal block is also equal to the number of diagonal blocks. It then follows that $p^2 = n$. The number of processors in the hypercube is denoted by P (numbered from 1 to P). It is further assumed that the number of diagonal blocks $k + 1$ is at least as large as the number of processors (P).

The blocks B_i, $i = 1, 2, \cdots, k$ are equally distributed among the first $P - 1$ processors, along with the corresponding S_i and G_i matrices. And the matrix F is processed by the node P. A brief description of the algorithm emphasizing the flow of data between the processors follows.

4.2.1. Host program.

generate matrix A and the vector s
compute the number of the blocks that each node numbered from 1 to $P - 1$ gets
for $i := 1$ *to* $P - 1$
    send appropriate blocks of B, S, G, *and* s to node i
send F to node P
wait for the solution parts to arrive from all the nodes

4.2.2. Node program.

if it is not the last node (P) then
    receive the matrix blocks B, G, S, and s
    diagonalize each B_i and solve the LSE problem as described in Algorithm 3
    send the matrices $\hat{G}_i$ and $S_i^{[2]}$ along with $s_i^{[2]}$, R_i, and r_i to node P (cf. Algorithm 5)
    wait for x_{k+1} and w_i vectors to arrive from node P
    complete the solution process to get x_i
    send x_i's to the host
else
    receive F from the host
    receive the matrices $\hat{G}_i$ and $S_i^{[2]}$, $s_i^{[2]}$, R_i, and r_i sent by all other nodes
    solve the system (2.17)
    broadcast x_{k+1} and appropriate blocks of w_i to all the other $P - 1$ nodes
    send x_{k+1} to host

The above Algorithm was implemented in FORTRAN on an Intel hypercube (iPSC/1) at the ACRF facility at Argonne National Laboratory, and Table 2 shows the timings results from these experiments. The matrix in each case was a $p^2 \times p^2$ matrix. For a fixed value of p, the problem was run on cubes of different dimensions to determine

TABLE 1
Error in Algorithms 3 and 5 for random matrices.

n	$k+1$	Estimated condition no.	Error: Alg. 3	Error: Alg. 5
2	2	2.0E02	0	0
4	2	1.0E01	9.0E−6	6.0E−7
10	2	4.0E01	3.0E−6	2.0E−6
10	3	1.0E02	4.0E−6	2.0E−6
20	4	3.0E02	9.0E−6	3.0E−6
40	5	8.0E02	2.0E−4	4.0E−5
60	6	9.0E02	8.0E−5	7.0E−6
80	8	2.0E03	1.0E−4	2.0E−5
100	10	2.0E04	2.0E−3	5.0E−5

the speedup. *The time shown is elapsed time in seconds from the moment the host starts sending the data to the nodes till the final solution is returned to the host.*

It appears from the results that by increasing the number of processors by a factor of j, one would get a speedup by a factor of $j/2$. The main reason is that the back substitution process has a bottleneck—the other nodes must remain idle while node P determines x_{k+1} and w.

4.2.3. Complexity of the parallel algorithm. It is assumed that the time required to transmit a message of N words from one node to another is $(\alpha + \beta N)d$, where α is the start-up time for the message and β is the time required to send one word after the initial message is set up and d is the distance between the nodes.

The only communication required in the parallel algorithm described above is the transmission of $\hat{G}_i$, $S_i^{[2]}$, $s_i^{[2]}$, R_i, and r_i to node P and vectors x, w from node P to nodes 1 to $P-1$. Since the size of R_i is much larger than other matrices and since the maximum distance between any two nodes on the hypercube is $\log P$, it is easily seen that the upper bound on the communication complexity of the algorithm is $O([P\alpha + \beta(p^2 + Pn)] \log P)$.

The computational complexity is easier to bound because all the computational work except the solution of (2.17) is done in parallel, and hence it is divided equally among $P-1$ processors. However, the matrix in system (2.17) is of the order $p + \sum_{i=1}^{n} (m_i - l_i)$ and hence only $\frac{2}{3}(p + \sum_{i=1}^{n} (m_i - l_i))^3$ are not done in parallel.

TABLE 2
Timings results on Intel *hypercube.*

Size of each block (p)	No. of processors (P)	Time in seconds
8	8	1.22
8	4	1.46
8	2	2.64
16	16	5.36
16	8	7.86
16	4	11.46
32	32	34.12
32	16	48.62
32	8	71.94

Acknowledgments. The authors thank the Argonne National Laboratory for the usage of the Intel Hypercube at the ACRF facility. We also acknowledge the help of John Gilbert and some anonymous referees.

REFERENCES

[1] J. L. BARLOW AND S. L. HANDY, *The direct solution of weighted and equality constrained least squares problems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 704–716.

[2] C. H. BISCHOF, *Incremental condition estimation*, Tech. Report ANL/MCS-P15-1088, Mathematics and Computer Sciences Division, Argonne National Laboratory, Argonne, IL, 1988; SIAM J. Matrix Anal. Appl., 11 (1990), pp. 312–322.

[3] Å. BJÖRCK AND G. H. GOLUB, *Iterative refinement of linear least squares solutions by Householder transformations*, BIT, 7 (1967), pp. 322–337.

[4] P. E. BJØRSTAD AND O. B. WIDLUND, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal., 23 (1986), pp. 1097–1120.

[5] T. F. CHAN, *Rank revealing* QR *factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.

[6] T. F. CHAN AND D. C. RESASCO, *A survey of preconditioners for domain decomposition*, Tech. Report YALEU/Department of Computer Science/RR-414, Yale University, New Haven, CT, 1985.

[7] F. DUCHIN AND D. B. SZYLD, *Application of sparse matrix techniques to inter-regional input-output analysis*, Econom. Planning, 15 (1979), pp. 147–167.

[8] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 1987.

[9] L. ELDEN, *Perturbation theory for the least squares problem with equality constraints*, SIAM J. Numer. Anal., 17 (1980), pp. 338–350.

[10] L. V. FOSTER, *Rank and null space calculations using matrix decomposition without column pivoting*, Linear Algebra Appl., 74 (1986), pp. 47–72.

[11] J. A. GEORGE AND J. W. H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[13] M. GUNZBERGER AND R. NICHOLAIDES, *Elimination with noninvertible pivots*, Linear Algebra Appl., 64 (1985), pp. 183–189.

[14] ———, *On substructuring algorithms and solution techniques for the numerical approximation of partial differential equations*, Appl. Numer. Math., 64 (1986), pp. 243–256.

[15] M. T. HEATH, *Some extensions of an algorithm for sparse linear least squares problems*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 223–237.

[16] C. L. LAWSON AND R. J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[17] ALEX POTHEN, *Sparse null bases and marriage theorems*, Ph.D. thesis, Cornell University, Ithaca, NY, 1984.

[18] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 589–600, October 1990

009

AN ALTERNATING PROJECTION ALGORITHM FOR COMPUTING THE NEAREST EUCLIDEAN DISTANCE MATRIX*

W. GLUNT†, T. L. HAYDEN†, S. HONG†, AND J. WELLS†

Abstract. Recent extensions of von Neumann's alternating projection algorithm permit an effective numerical approach to certain least squares problems subject to side conditions. This paper treats the problem of minimizing the distance from a given symmetric matrix to the class of Euclidean distance matrices; in dimension $n = 3$ we obtain the solution in closed form.

Key words. alternating projections, distance matrices, matrix cones, normal cones

AMS(MOS) subject classifications. 49D99, 51K05, 65F30, 92A40

1. Introduction.

The purpose of this paper is to propose an efficient computational algorithm for solving the following problem:

(I) Given a real symmetric matrix $F \in \mathbb{R}^{n\times n}$, find the Euclidean distance matrix $\bar{D} \in \mathbb{R}^{n\times n}$ that minimizes

$$\|F-D\|.$$

Here, matrix norm means the Frobenius norm and $D = \{d_{ij}\} \in \mathbb{R}^{n\times n}$ is a *Euclidean distance matrix* if

(1.1) (i) $d_{ij} = d_{ji}$,
(ii) $d_{ii} = 0$,
(iii) there exist points $P_1, \cdots, P_n$ in $\mathbb{R}^r$ $(r \leqq n-1)$ such that

$$d_{ij} = \|P_i - P_j\|^2 (1 \leqq i,j \leqq n).$$

Our goal is to place Problem (I) in the setting of minimizing a quadratic functional over the intersection of a finite collection of convex sets (in the ambient space construct the respective projection maps onto the convex sets) and apply the alternating projection method of Dykstra [7], which guarantees convergence to the solution of (I).

This problem is a special case, indeed the easiest, of the more general ones of finding the minimum in (I) over the class of Euclidean distance matrices for which the embedding points $P_1, \cdots, P_n$ lie in a Euclidean space of dimension r, the case $r = 3$ being of prime concern. The major difficulty in studying such lower rank problems is that one loses convexity of the constraint set, so there looms the computationally difficult issue of distinguishing local and absolute minima.

Such problems arise in the conformation of molecular structures from nuclear magnetic resonance data. One wishes to determine a molecular model in $\mathbb{R}^3$ whose generated Euclidean distance matrix minimizes the distance to the given data matrix [13]. Other applications arise under the general title of multidimensional scaling. A discussion of four types of multidimensional scaling, with references to specific applications from the social and behavioral sciences, geography, and genetics, may be found in de Leeuw and Heiser [5]. A broader review of scaling, with applications and algorithms, is given in Young [20]. The book by Meulman [15] gives additional related applications in multivariate analysis.

* Received by the editors January 9, 1989; accepted for publication (in revised form) July 7, 1989. This work was supported in part by National Science Foundation grant CHE-8802341.

† Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506 (glunt@ms.uky.edu; hayden@ms.uky.edu; hong@ms.uky.edu; and wells@ms.uky.edu).

Although this paper makes no direct contribution to solving the lower rank version of Problem (I), we do develop basic geometry, optimality conditions, and possible initial starting points that should aid the further development of algorithms.

2. Notation and background. We consider real $n \times n$ matrices A, B and define their inner product by

$$\langle A, B \rangle = \sum_{i,j=1}^{n} a_{ij} b_{ij};$$

$\|A\| = \langle A, A \rangle^{1/2}$ is the Frobenius norm and the distance between matrices A and B is $\|A - B\|$. In addition to the notion of a Euclidean distance matrix defined in (1.1), we shall refer to $D \in \mathbb{R}^{n \times n}$ as a *predistance matrix if* (i) and (ii) of (1.1) are satisfied.

Schoenberg gave the first modern characterization of Euclidean distance matrices [17]. He showed that the predistance matrix $D \in \mathbb{R}^{(n+1)\times(n+1)}$ is a Euclidean distance matrix if and only if the $n \times n$ symmetric matrix A defined by

$$a_{ij} = \tfrac{1}{2}[d_{oi} + d_{oj} - d_{ij}] \qquad (1 \leqq i, j \leqq n) \tag{2.1}$$

is positive semidefinite ($A \geqq 0$). Furthermore, $r = rk(A)$ is the minimum imbedding dimension, that is, the lowest dimensional Euclidean space in which there exist points that satisfy (iii) of (1.1). And, finally, if one considers the spectral decomposition

$$A = U \Lambda U^T$$

and defines C by $C = U\Lambda^{1/2}$, then $A = CC^T$ and the columns of C^T furnish coordinate choices for $P_0 = 0$, P_1, P_2, $\cdots$, P_n. Independently, and three years later, Young and Householder [21] published the same results.

Evidently, whether a predistance matrix is a Euclidean distance matrix is unaffected by shifting the origin to the centroid of an embedding configuration. This is accomplished by bordering D with $d_{00} = 0$ and

$$d_{ko} = d_{ok} = \frac{1}{n} \sum_{j=1}^{n} d_{kj} - \frac{1}{2n^2} \sum_{i,j=1}^{n} d_{ij} \qquad (1 \leqq k \leqq n). \tag{2.2}$$

Gower [9], [10], (see also [14]) in his work on multidimensional scaling, gave a simple matrix formulation of this process: start with a predistance matrix D, border it according to (2.2), and apply the Schoenberg transformation (2.1). The resulting matrix A in (2.1) is given by

$$2A = P(-D)P$$

where

$$P = I - \frac{1}{n} ee^T, \qquad e = [1, 1, \cdots, 1]^T \tag{2.3}$$

is the orthogonal projection onto the subspace

$$M = \{x \in \mathbb{R}^n : x^T e = 0\}. \tag{2.4}$$

Thus the predistance matrix D is a Euclidean distance matrix if and only if

$$P(-D)P \geqq 0. \tag{2.5}$$

Obviously (2.5) is equivalent to requiring that $-D$ be positive semidefinite on M.

For our purposes it will be convenient to replace the projection P in (2.5) by the Householder matrix

$$Q = I - \frac{2}{v^T v} v v^T, \qquad v = [1, 1, \cdots, 1, 1 + \sqrt{n}]^T. \tag{2.6}$$

Given any matrix $F = F^T \in \mathbb{R}^{n \times n}$ there is a unique matrix $\hat{F} = \hat{F}^T \in \mathbb{R}^{(n-1)\times(n-1)}$ such that

$$QFQ = \begin{bmatrix} \hat{F} & f \\ f^T & \zeta \end{bmatrix}. \tag{2.7}$$

In [12], two of the present authors show that

$$F \geqq 0 \quad \text{on } M \text{ if and only if} \quad \hat{F} \geqq 0. \tag{2.8}$$

In particular, a predistance matrix $D \in \mathbb{R}^{n \times n}$ is a Euclidean distance matrix if and only if

$$Q(-D)Q = \begin{bmatrix} -\hat{D} & d \\ d^T & \delta \end{bmatrix}, \qquad -\hat{D} \geqq 0, \tag{2.9}$$

and that the minimal embedding dimension is $r = rk(\hat{D})$. The precise relation between (2.5) and (2.9) appears once it is observed that

$$P = Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q$$

and hence that

$$P(-D)P = Q \begin{bmatrix} -\hat{D} & 0 \\ 0 & 0 \end{bmatrix} Q.$$

The advantage of formulation in (2.9) over that given in (2.5) is that it provides the basis for the construction of a projection map essential to the implementation of Dykstra's algorithm.

3. Matrix cones. Recalling that $M = \{x \in \mathbb{R}^n : e^T x = 0\}$, we define

$$K_1 = \{A : A \in \mathbb{R}^{n \times n}, A = A^T, \quad \text{and} \quad x^T A x \geqq 0 \quad \text{for all} \quad x \in M\} \tag{3.1}$$

and

$$K_2 = \{A : A \in \mathbb{R}^{n \times n}, A = A^T, \quad \text{and} \quad a_{ii} = 0, i = 1, 2, \cdots, n\}. \tag{3.2}$$

In the inner product space of real symmetric $n \times n$ matrices, K_1 is a closed convex cone and K_2 is a subspace. Clearly $D \in K_1 \cap K_2$ if and only if $-D$ is a Euclidean distance matrix. Moreover, the matrices in K_1 are characterized by (2.7) and (2.8). Thus, the approximation problem (I) is a special case of the following problem:

(II) Given $F = F^T \in \mathbb{R}^{n \times n}$,

$$\min_{A \in K_1 \cap K_2} \| F - A \|.$$

The minimizing matrix $\bar{A}$ for (II) is uniquely characterized by the condition

$$\langle Z - \bar{A}, F - \bar{A} \rangle \leqq 0 \qquad (Z \in K_1 \cap K_2), \tag{3.3}$$

which says, in effect, that $F - \bar{A}$ must belong to the normal cone to $K_1 \cap K_2$ at $\bar{A}$ (see [8] or [16]). In general, if K is any convex set in $\mathbb{R}^n$ and $a \in K$, the *normal cone* to K at a is the closed convex cone

$$\partial K(a) = \{y \in \mathbb{R}^n : \langle z - a, y \rangle \leqq 0 \quad \text{for all} \quad z \in K\}. \tag{3.4}$$

Let $A \in K_2$. Since K_2 is a subspace it is clear that

$$\partial K_2(A) = \partial K_2(0) = \{B : B = \operatorname{diag}[b_1, b_2, \cdots, b_n]\}. \tag{3.5}$$

To find the normal cone $\partial K_1(A)$ at $A \in K_1$, we first use (2.7) to write A in the form

$$A = Q\begin{bmatrix} \hat{A} & a \\ a^T & \alpha \end{bmatrix} Q, \qquad \hat{A} \geqq 0 \tag{3.6}$$

along with the spectral decomposition

$$\hat{A} = U\Lambda U^T = U\begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} U^T \tag{3.7}$$

where $\Lambda > 0$.

THEOREM 3.1. *If* $A \in K_1$,

$$\partial K_1(A) = \left\{ B : B = Q\begin{bmatrix} U\begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} U^T & 0 \\ 0 & 0 \end{bmatrix} Q, \qquad M \in \mathbb{R}^{(n-r-1)\times(n-r-1)},\ M \leqq 0 \right\}. \tag{3.8}$$

Proof. Let $Z \in K_1$ and $B \in \partial K_1(A)$ be written in the form

$$Z = Q\begin{bmatrix} \hat{Z} & z \\ z^T & \omega \end{bmatrix} Q;\ \hat{Z} \geqq 0, \qquad B = Q\begin{bmatrix} \hat{B} & b \\ b^T & \beta \end{bmatrix} Q. \tag{3.9}$$

Then $\langle Z - A, B \rangle \leqq 0$ by (3.4) and, since Q is orthogonal,

$$\langle QZQ, QBQ \rangle \leqq \langle A, B \rangle.$$

Using the representations in (3.9), we arrive at the inequality

$$\langle \hat{Z}, \hat{B} \rangle + 2z^T b + \omega\beta \leqq \langle A, B \rangle. \tag{3.10}$$

For fixed $\hat{Z}$, $\hat{Z} \geqq 0$, the matrix Z in (3.9) remains in K_1 as z varies over $\mathbb{R}^{n-1}$ and ω varies over $\mathbb{R}^1$; hence, in view of (3.10), $b = 0$ and $\beta = 0$. Following Fletcher [8, p. 496], let $\hat{B} = V\Omega V^T$ be the spectral decomposition of $\hat{B}$ with V orthogonal and let $\Omega = \operatorname{diag}[\omega_1, \omega_2, \cdots, \omega_{n-1}]$ be the diagonal matrix of eigenvalues. From (3.10),

$$\langle A, B \rangle \geqq \langle \hat{Z}, \hat{B} \rangle = \langle V^T \hat{Z} V, \Omega \rangle = \langle C, \Omega \rangle = \sum_{j=1}^{n-1} c_{jj}\omega_j$$

where $C = V^T \hat{Z} V$ may be any positive semidefinite matrix, that is, $c_{11}, \cdots, c_{n-1,n-1}$ may be any sequence of nonnegative numbers. Thus $\omega_j \leqq 0$ $(1 \leqq j \leqq n-1)$ and $\hat{B} \leqq 0$. Therefore, every $B \in \partial K_1(A)$ has the form

$$B = Q\begin{bmatrix} \hat{B} & 0 \\ 0 & 0 \end{bmatrix} Q, \qquad \hat{B} \leqq 0. \tag{3.11}$$

From (3.11) and (3.9), $\langle Z, B \rangle \leqq 0$ for all $Z \in K_1$ and, since $\langle Z - A, B \rangle \leqq 0$,

$$\langle A, B \rangle \leqq \sup_{Z \in K_1} \langle Z, B \rangle \leqq 0 \leqq \langle A, B \rangle;$$

whence

$$\langle A, B \rangle = 0.$$

Using (3.6) and (3.11), this implies that

$$\langle \hat{A}, \hat{B} \rangle = 0$$

and, aided by the spectral decomposition of $\hat{A}$ from (3.7), that

$$\langle \Lambda, U^T \hat{B} U \rangle = \langle \Lambda, F \rangle = \sum_{j=1}^{r} \lambda_j f_{jj} = 0$$

where $F = U^T \hat{B} U \leqq 0$. Inasmuch as $\lambda_j > 0$ and $f_{jj} \leqq 0$ for $1 \leqq j \leqq r$, we conclude that $f_{jj} = 0$ for $1 \leqq j \leqq r$. Being negative semidefinite, F has the form

$$F = \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} \begin{matrix} r \\ n-r-1, \end{matrix} \qquad M \leqq 0.$$
$$\begin{matrix} r & n-r-1 \end{matrix}$$

Thus

$$\hat{B} = U \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} U^T, \qquad M \leqq 0$$

and B has the structure of (3.8). Conversely, it is easy to see that every such B does, in fact, belong to $\partial K_1(A)$. □

THEOREM 3.2. *If* $\bar{A} \in K_1 \cap K_2$,

$$\partial(K_1 \cap K_2)(\bar{A}) = \partial K_1(\bar{A}) + \partial K_2(\bar{A}), \tag{3.12}$$

that is, if $\bar{A} \in K_1 \cap K_2$ *is written in the form* (3.6), *and* $F = F^T \in \mathbb{R}^{n \times n}$, *then* $\bar{A}$ *solves problem* (II) *if and only if*

$$F - \bar{A} = Q \begin{bmatrix} U \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} U^T & 0 \\ 0 & 0 \end{bmatrix} Q + B \tag{3.13}$$

or

$$F = Q \begin{bmatrix} U \begin{bmatrix} \Lambda & 0 \\ 0 & M \end{bmatrix} U^T & a \\ a^T & \alpha \end{bmatrix} Q + B \tag{3.14}$$

where, $\Lambda > 0$, $M \leqq 0$, *and* $B = diag\,[b_1, \cdots, b_n]$.

Proof. According to [16, p. 223], (3.12) holds for any two convex sets whose relative interiors have a point in common, a hypothesis clearly satisfied in the present setting: any $\bar{A} \in K_1 \cap K_2$ in the form (3.6) that has $\hat{A} > 0$ belongs to the relative interior of both K_1 and K_2.

In the context of problem (II), equations (3.13) and (3.14) simply state that a given $\bar{A} \in K_1 \cap K_2$ solves (II) if and only if $F - \bar{A}$ satisfies (3.13), that is, $F - \bar{A} \in \partial(K_1 \cap K_2)(\bar{A})$. □

Whatever the application, the computational success of Dykstra's algorithm depends crucially upon the computational complexity of the relevant projections. In our setting we need the projections P_1 onto K_1 and P_2 onto K_2. Since K_2 is the subspace consisting

of all real symmetric $n \times n$ matrices with zero diagonal,

$$P_2(F) = F - \operatorname{diag}(F), \tag{3.15}$$

that is, P_2 maps F to the matrix obtained by replacing each diagonal element by zero.

Given $F = F^T \in \mathbb{R}^{n\times n}$, $P_1(F)$ is the unique solution to the problem

$$\min_{A \in K_1} \|F - A\|.$$

To compute $P_1(F)$, we use the representation

$$F = Q\begin{bmatrix} U\Lambda U^T & f \\ f^T & \zeta \end{bmatrix} Q$$

from (2.6) and decompose the diagonal matrix Λ into its positive and negative parts: $\Lambda = \Lambda^+ - \Lambda^-$, $\Lambda^+ \geqq 0$, $\Lambda^- \geqq 0$. According to [12, Thm. 2.1],

$$P_1(F) = Q\begin{bmatrix} U\Lambda^+ U & f \\ f^T & \zeta \end{bmatrix} Q. \tag{3.16}$$

4. The method of alternating projections. The minimization problem treated here is one of a broad class of problems, conveniently posed in a real Hilbert space H, which ask for the point $\bar{f}$ in the intersection of a finite number of convex sets $C_1, C_2, \cdots, C_m$ that minimizes the distance from a given point f; thus

$$\|f - \bar{f}\| = \min_{g \in \cap_{i=1}^m C_i} \|f - g\|. \tag{4.1}$$

The fundamental idea of the algorithm presented here traces its origin to von Neumann [19] who, in 1933, showed that if S_1 and S_2 are closed subspaces of H and P_1, P_2 are, respectively, the orthogonal projections onto S_1 and S_2, then the sequence of alternating projections

$$P_1 f, P_2 P_1 f, P_1 P_2 P_1 f, P_2 P_1 P_2 P_1 f, \cdots \tag{4.2}$$

converges to $P_{S_1 \cap S_2} f$, the orthogonal projection onto the intersection of S_1 and S_2. Deutsch [6] showed that the rate of convergence on (4.2) depends on the "angle" between the two subspaces, decreasing with the angle.

Cheney and Goldstein [3] analyzed (4.2) in the case that the subspaces are replaced by closed convex sets C_1 and C_2, and P_1 and P_2 represent, respectively, the projections (proximity maps) onto C_1 and C_2. They showed that if one of the sets is compact or one is finite dimensional and the distance between them is attained, then

$$\lim_{k \to \infty} (P_2 P_1)^k f = \bar{f},$$

a point in the fixed-point set of $P_2 P_1$ which, it turns out, is the set of points in C_2 nearest C_1. In general $\bar{f}$ need not be the near point to f in $C_1 \cap C_2$. For example, consider the convex sets $C_1 = \{(x_1, x_2)\colon x_1^2 + x_2^2 \leqq 1\}$ and $C_2 = \{(x_1, x_2)\colon x_1 = 0, x_2 \leqq 2\}$ in $\mathbb{R}^2$ and let $f = (3, 4)$. As Han points out in [11], algorithm (4.2) stops at $P_2 P_1 f = (0, \frac{4}{5})$ while $\bar{f} = (0, 1)$. As one would expect, von Neumann's alternating method cannot be applied successfully to problem (II) for general n. However, as we shall see in § 6, it does converge to the near point solution of problem (II) in dimension $n = 3$.

Dykstra's algorithm [7] (which, of course, solves (II) in all dimensions) is based on an ingeniously simple modification of (4.2). Given a point f, closed convex sets C_1,

$C_2, \cdots, C_m$, and the corresponding projection maps $P_1, P_2, \cdots, P_m$, the algorithm sets

$$y_1^{(0)} = y_2^{(0)} = \cdots = y_m^{(0)} = 0, \qquad x_m^{(0)} = f$$

and in each iteration computes $2m$ vectors

$$x_1^{(k)}, y_1^{(k)}, \cdots, x_m^{(k)}, y_m^{(k)}$$

as follows: set $x_0^{(k)} = x_m^{(k-1)}$ for $i = 1, 2, \cdots, m$, $k = 1, 2, \cdots$

$$\begin{aligned} z &= x_{i-1}^{(k)} + y_i^{(k-1)} \\ x_i^{(k)} &= P_i(z) \\ y_i^{(k)} &= z - P_i(z). \end{aligned} \tag{4.3}$$

Then, for $i = 1, 2, \cdots, m$, $x_i^{(k)} \to \bar{f}$ as $k \to \infty$, $\bar{f}$ being the solution of (4.1).

This notation is adopted from Han [11] who discovered the algorithm independently. Algorithm (4.3) differs from (4.2) in that at each step the preceding outward-pointing normal $y_i^{(k-1)}$ is added to $x_{i-1}^{(k)}$ before applying the projection P_i. This operation has the effect of forcing the z iterates toward the normal cone of $\cap_{i=1}^m C_i$ at the solution $\bar{f}$.

As Dykstra and Boyle point out in [1], the term $y_i^{(k)}$ may be suppressed at any step that precedes projection onto a subspace. In our setting, in which we have a closed convex set K_1, a subspace K_2, and projections P_1 and P_2 given by (3.15) and (3.16), algorithm (4.3) takes the following form:

$$\begin{aligned} &\text{Let } F_0 = F \\ &\text{For } k = 0, 1, 2, \cdots \\ &F_{k+1} = F_k + [P_2 P_1(F_k) - P_1(F_k)]. \end{aligned} \tag{4.4}$$

Here F_k plays the role of z in (4.3). We shall refer to (4.4) as the modified alternating projection algorithm (MAP). Using the Boyle–Dykstra convergence result [1, Thm. 2], we have the following theorem.

THEOREM 4.1. *Given $F = F^T \in \mathbb{R}^{n \times n}$ and the sequence $\{F_k\}$ generated by (4.4), both $\{P_1 F_k\}$ and $\{P_2 P_1 F_k\}$ converge in the Frobenius norm to the unique solution $\bar{A}$ of*

$$\min_{A \in K_1 \cap K_2} \|F - A\|.$$

5. Numerical results and comparisons. Other algorithms that address problem (II) and its lower rank versions consist of fixed-point iteration based essentially on the relation in (3.13).

Since $Qe = -\sqrt{n}e_n$ (see (2.3) and (2.6)) it follows in (3.13) that

$$(F - \bar{A})e = Be = [b_1, b_2, \cdots, b_n]^T$$

so that the diagonal matrix B has for its elements the difference of the row sums of F and $\bar{A}$. For any matrix $A \in \mathbb{R}^{n \times n}$, let

$$D(A) = \operatorname{diag}[\textstyle\sum a_{1j}, \cdots, \sum a_{nj}],$$

the diagonal matrix whose entries are the row sums of A, in row order. Then we may write (3.13) in the form

$$F - \bar{A} = Q \begin{bmatrix} U \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} U^T & 0 \\ 0 & 0 \end{bmatrix} Q + D(F) - D(\bar{A}) \tag{5.1}$$

or

$$F - D(F) + D(\bar{A}) = Q\begin{bmatrix} U\begin{bmatrix}\Lambda & 0\\ 0 & M\end{bmatrix}U^T & a\\ a^T & \alpha\end{bmatrix}Q, \tag{5.2}$$

where

$$\bar{A} = Q\begin{bmatrix} U\begin{bmatrix}\Lambda & 0\\ 0 & 0\end{bmatrix}U^T & a\\ a^T & \alpha\end{bmatrix}Q, \tag{5.3}$$

is as in (3.6) and (3.7). An application of the projection map P_1 from (3.16) yields

$$P_1(F - D(F) + D(\bar{A})) = \bar{A}.$$

Hence the solution $\bar{A}$ of (II) is a fixed point of the map G defined by

$$G(A) = P_1(F - D(F) + D(A)). \tag{5.4}$$

THEOREM 5.1. *For each $F = F^T \in \mathbb{R}^{n\times n}$, the map G has a unique fixed point $\bar{A} \in K_1 \cap K_2$ that is the solution of* (II).

Proof. As we have just shown, for each $F = F^T \in \mathbb{R}^{n\times n}$ the solution $\bar{A}$ of (II) is a fixed point of the map G.

Conversely, suppose $\bar{A} \in K_1 \cap K_2$ is a fixed point of the map G. Because P_1 applied to $F - D(F) + D(\bar{A})$ must yield $\bar{A}$ and since $\bar{A}$ has the form (5.3), it follows that $F - D(F) + D(\bar{A})$ is given by (5.2); hence $F - \bar{A}$ has the form (5.1) so that $\bar{A}$ solves (II). □

A similar argument shows that the map G_α defined by

$$G_\alpha(A) = P_1(\alpha A + (1-\alpha)(F - D(F) + D(A)))(0 \leqq \alpha < 1) \tag{5.5}$$

has a unique fixed point $\bar{A} \in K_1 \cap K_2$, and $\bar{A}$ is the solution of (II).

One of the key algorithms in multidimensional scaling can be based on (5.5). However, researchers such as de Leeuw, Takane, and Browne [5], [18], [2] approach the solution of (II) by computing the gradient of $\|F - A\|$, and finding that

$$(F - \bar{A} - D(F) + D(\bar{A}))P\bar{A}P = 0$$

is a necessary condition for minimality. A description of the algorithm in our setting requires a new operation called Fix.

If $F = F^T \in \mathbb{R}^{n\times n}$ has the form

$$F = Q\begin{bmatrix}\hat{F} & f\\ f^T & \zeta\end{bmatrix}Q,$$

then, keeping $\hat{F}$ fixed, there are unique replacements $\bar{f}$ for f and $\bar{\zeta}$ for ζ that yield a matrix with zero diagonal. We denote this matrix by Fix (F). Thus

$$\text{Fix}\,(F) = Q\begin{bmatrix}\hat{F} & \bar{f}\\ \bar{f}^T & \bar{\zeta}\end{bmatrix}Q \tag{5.6}$$

where $\bar{f}$ and $\bar{\zeta}$ can be calculated from

$$cQ\begin{bmatrix}2\bar{f}\\ \bar{\zeta}\end{bmatrix} + \left(\text{diag}\left(Q\begin{bmatrix}\hat{F} & 0\\ 0 & 0\end{bmatrix}Q\right)\right)e = 0, \qquad c = -1/\sqrt{n}, \tag{5.7}$$

where $(\text{diag}\,(A))e$ is the column vector whose entries are the diagonal elements of A in row order. We observe that if $F \in K_1$, then Fix $(F) \in K_1 \cap K_2$.

Elegant Algorithm (E)

Given $F = F^T \in \mathbb{R}^{n\times n}$, choose α, $0 < \alpha < 1$ and let $F_1 = \text{Fix } P_1(F)$. For $k = 1, 2, 3, \cdots$

$$F_{k+1} = \text{Fix } P_1((1-\alpha)F_k) + \alpha(F - D(F) + D(F_k)). \tag{5.8}$$

At each stage $F_k \in K_1 \cap K_2$ so if $F_k \to \bar{A}$ as $k \to \infty$, then, clearly, $\bar{A} \in K_1 \cap K_2$ and

$$\bar{A} = \text{Fix } P_1((1-\alpha)\bar{A}) + \alpha(F - D(F) + D(\bar{A})). \tag{5.9}$$

Furthermore, $\bar{A}$ must also be a fixed point of (5.5) and, accordingly, the solution of (II). To see this note that the row sums of $(1-\alpha)\bar{A} + \alpha(F - D(F) + D(\bar{A}))$ and $\bar{A}$ are identical. Thus, because $Qe_n = -e/\sqrt{n}$, the nth row and column of

$$S = Q((1-\alpha)\bar{A}) + \alpha(F - D(F) + D(\bar{A}))Q \quad \text{and} \quad Q\bar{A}Q$$

are identical. But since Fix acts only on the last row and column of QSQ, we conclude that $P_1 S = \bar{A}$. Thus Fix is irrelevant and (5.9) implies that $G_\alpha(\bar{A}) = \bar{A}$.

For $\alpha = 2/(n-1)^2$ $(n > 1)$, this algorithm is due to de Leeuw [4] and its convergence is discussed in Takane [18]. Because convergence is slow for small α, Browne [2] proposed using $\alpha = \frac{1}{2}$ as long as $\|F_{k+1} - F\| < \|F_k - F\|$ and then reducing α by a factor of $\frac{1}{2}$ at failure and continuing with this criterion as required. He termed this modification ELEGANT STAR (ES) and reported a vastly improved convergence rate. To further improve the rate of convergence, Browne added a penalty function to the iteration in (5.9) and introduced an intermediate Newton–Raphson step (NR). Our numerical computations support his claim that NR converges more rapidly than ES.

Table 1 compares the ES, NR, and MAP algorithms, using $\|F_{k+1} - F_k\| < 10^{-5}$ as a stopping criterion. All three algorithms converge to essentially the same values with maximum distance between final elements of NR and MAP on the order of 10^{-5} and on the order of 10^{-3} for ES and MAP. On two test matrices ES stopped prematurely, requiring a restart to eventually reach an acceptable solution.

The numerical tests were performed on each of four randomly generated predistance matrices of order 4, 8, 16, 32, 64, 100 with values uniformly distributed between 10^{-5} and 10. The average number of iterations, the average CPU time, and the standard deviation (in parentheses) are reported. All data was obtained on an IBM 3090–300E at the University of Kentucky's Center for Computational Science. All computations were done in double precision with a machine epsilon of approximately 10^{-15}.

TABLE 1

	ES		NR		MAP	
n	NI	CPU	NI	CPU	NI	CPU
4	18	.12	21	.17	26	.16
8	31	.25	17	.29	19	.22
16	71	1.94	16	1.62	30	.78
32	175	32.0(.05)	14	17.64(2.3)	36	5.01(.08)
64	367	2189.4(5.23)	14	232.23(17.4)	56	53.46(2.9)
100	708	3095.0(540)	13	948.11(529.8)	68	241.56(16.0)

NI: *Average number of iterations.*
CPU: *Average* CPU *time in seconds.*
( ): *Standard deviation in* CPU *time.*

Although the number of iterations of NR actually decreased with size, it consumes more CPU time than MAP roughly in the ratio 4:1. Moreover, an attempt to fit a power function ($y = ax^b$) to CPU time versus n for MAP relative to a larger data set yielded a best fit of CPU = $.000251n^{3.058}$. This is perhaps better than expected since the eigenvalue decomposition on each upper block constitutes about $16(n-1)^3$ work.

6. The case $n = 3$. Despite its failure in higher dimensions, von Neumann's alternating projection algorithm provides an explicit solution of Problem (I) in the case of a 3×3 predistance matrix. Specifically, we suppose that

$$D = \begin{bmatrix} 0 & \alpha & \beta \\ \alpha & 0 & \gamma \\ \beta & \gamma & 0 \end{bmatrix}, \tag{6.1}$$

put $F = -D$, and write

$$F = Q\begin{bmatrix} U\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}U^T & f \\ f^T & \zeta \end{bmatrix}Q \tag{6.2}$$

where $\lambda_1 \geqq 0 > \lambda_2$ and

$$Q = \begin{bmatrix} a & b & c \\ b & a & c \\ c & c & c \end{bmatrix},$$

with

$$c = -1/\sqrt{3}, b = -1/(3+\sqrt{3}), a = 1+b. \tag{6.3}$$

Notice, Problem (I) is trivial if both of the eigenvalues, λ_1 and λ_2, in (6.2) are nonnegative since, then, D itself would be a Euclidean distance matrix.

THEOREM 6.1. *Let D and F be given by* (6.1) *and* (6.2) *with* $\lambda_1 \geqq 0 > \lambda_2$ *and let* $\bar{\lambda} = \lambda_1 + \lambda_2/3$. *Then the solution* $\bar{A}$ *to Problem* (I), *namely*,

$$\min_{A \in K_1 \cap K_2} \|F - A\|$$

is $\bar{A} = 0$ *if* $\bar{\lambda} \leqq 0$; *otherwise*

$$\bar{A} = Q\begin{bmatrix} U\begin{bmatrix} \bar{\lambda} & 0 \\ 0 & 0 \end{bmatrix}U^T & \bar{f} \\ \bar{f}^T & \bar{\zeta} \end{bmatrix}Q \tag{6.4}$$

where $\bar{f} = [f_1, f_2]^T$ *and* $\bar{\zeta}$ *are computed by the* Fix *operation* (5.6) *and* (5.7).

We shall only outline our rather tedious proof. First, let $\lambda_1^{(1)} = \lambda_1$, $\lambda_2^{(1)} = \lambda_2$ and consider the case $\bar{\lambda} = \lambda_1 + \lambda_2/3 \geqq 0$. With von Neumann's algorithm in the form

$$F_1 = F, F_2 = P_2P_1F_1, \cdots, F_{k+1} = P_2P_1F_k, \cdots,$$

a lengthy but straightforward calculation shows the principal 2×2 submatrix $\hat{F}_{k+1}$ of $QF_{k+1}Q$ has the spectral decomposition

$$U\begin{bmatrix} \lambda_1^{(k+1)} & 0 \\ 0 & \lambda_2^{(k+1)} \end{bmatrix}U \tag{6.5}$$

where

$$\lambda_1^{(k+1)} = \lambda_1^{(k)} + \lambda_2^{(k)}/2, \qquad \lambda_2^{(k+1)} = \lambda_2^{(k)}/2 \qquad (k = 1, 2, \cdots).$$

Because $\bar{\lambda} \geqq 0$, each $\lambda_1^{(k)} \geqq 0$ and

$$\lambda_1^{(k)} \to \lambda_1 + \lambda_2/3, \qquad \lambda_2^{(k)} \to 0 \text{ as } k \to \infty.$$

Using the result of Cheney and Goldstein [3], it follows that the sequence F_k, $k = 1, 2, \cdots$ converges to a matrix $\bar{A}$ of the form

$$\bar{A} = Q \begin{bmatrix} U \begin{bmatrix} \bar{\lambda} & 0 \\ 0 & 0 \end{bmatrix} U^T & \bar{f} \\ \bar{f}^T & \bar{\zeta} \end{bmatrix} Q, \tag{6.6}$$

where the border is computed by the Fix operation. That $\bar{A}$ is actually the unique near point to F in $K_1 \cap K_2$ follows from the inequality

$$\langle Z - \bar{A}, F - \bar{A} \rangle \leqq 0$$

that holds for all $Z \in K_1 \cap K_2$.

In the case remaining, $\bar{\lambda} = \lambda_1 + \lambda_2/3 < 0$, the iteration proceeds as before except that $\lambda_1^{(k)}$ in (6.4) eventually goes negative at some index $k' \geqq 2$, after which the eigenvalues of the principal submatrix of $QF_{k+1}Q$ follow the pattern

$$\lambda_1^{(k+1)} = \lambda_1^{(k)}/2 + \lambda_2^{(k)}/6, \qquad \lambda_2^{(k+1)} = \lambda_1^{(k)}/6 + \lambda_2^{(k)}/2$$

and, accordingly, tend to 0. Consequently, we conclude in this case that the near point is $\bar{A} = 0$. □

To illustrate, we find that the Euclidean distance matrix that best approximates the predistance matrix

$$D = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 9 \\ 1 & 9 & 0 \end{bmatrix}$$

in the Frobenius norm is given by the Euclidean distance matrix

$$\bar{D} = \begin{bmatrix} 0 & \frac{19}{9} & \frac{19}{9} \\ \frac{19}{9} & 0 & \frac{76}{9} \\ \frac{19}{9} & \frac{76}{9} & 0 \end{bmatrix}.$$

Note added in proof. The following reference was recently brought to our attention by N. Gaffke and R. Mather [*A cyclic projection algorithm via duality*, Metrika, 36 (1989), pp. 29–54]. A new proof of the Dykstra–Boyle–Han result is given and several applications, including the Euclidean fit of distance matrices in data analysis, are presented.

REFERENCES

[1] J. P. BOYLE AND R. L. DYKSTRA, *A method for finding projections onto the intersection of convex sets in Hilbert Space*, in Advances in Order Restricted Statistical Inference, R. Dykstra, T. Robertson, and F. T. Wright, eds., Vol. 37, Lecture Notes in Statistics, Springer-Verlag, 1986, pp. 28–47.

[2] M. W. BROWNE, *The Young–Householder algorithm and the least square multidimensional scaling of squared distance*, J. Classification, 4 (1987), pp. 175–190.

[3] W. CHENEY AND A. GOLDSTEIN, *Proximity maps for convex sets*, Proc. Amer. Math. Soc., 10 (1959), pp. 448–450.

[4] J. DE LEEUW, *An alternating least squares approach to squared distance scaling*, unpublished manuscript, Department of Data Theory, University of Leiden, Leiden, the Netherlands, 1975.

[5] J. DE LEEUW AND W. HEISER, *Multidimensional scaling with restrictions on the configuration*, in Multivariate Analysis—V, P. R. Krishnaiah, ed., North Holland, Amsterdam, 1980, pp. 502–522.

[6] F. DEUTSCH, *Von Neumann's alternating method: The rate of convergence*, Approximation Theory IV, C. Chui, L. Schumaker, and J. Ward, eds., Academic Press, New York, London, 1983, pp. 427–434.

[7] R. L. DYKSTRA, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 839–842.

[8] R. FLETCHER, *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–513.

[9] J. C. GOWER, *Euclidean distance geometry*, Math. Sci., 7 (1982), pp. 1–14.

[10] ———, *Properties of Euclidean and non-Euclidean distance matrices*, Linear Algebra Appl., 67 (1985), pp. 81–97.

[11] S. P. HAN, *A successive projection method*, Math. Programming, 40 (1988), pp. 1–14.

[12] T. L. HAYDEN AND J. WELLS, *Approximation by matrices positive semidefinite on a subspace*, Linear Algebra Appl., 109 (1988), pp. 115–130.

[13] T. HAVEL, I. KNUTZ, AND G. CRIPPEN, *The theory and practice of distance geometry*, Bull. Math. Biol., 45 (1983), pp. 665–720.

[14] R. MATHAR, *The best Euclidean fit to a given distance matrix in prescribed dimensions*, Linear Algebra Appl., 67 (1985), pp. 1–6.

[15] J. MEULMAN, *A distance approach to nonlinear multivariate analysis*, DSWO Press, Leiden, the Netherlands, 1986.

[16] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[17] L. J. SCHOENBERG, *Remarks to M. Frechet's article "Sur la définition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de Hilbert,"* Ann. of Math., 36 (1935), pp. 724–732.

[18] Y. TAKANE, *On the relations among four methods of multidimensional scaling*, Behaviormetrika, 4 (1977), pp. 29–43.

[19] J. VON NEUMANN, *The geometry of orthogonal spaces*, Functional Operators Vol. II, Ann. Math. Stud. No. 22, Princeton University Press, Princeton, NJ, 1950.

[20] F. W. YOUNG, *Scaling*, Ann. Rev. Psychol., 35 (1984), pp. 55–81.

[21] G. YOUNG AND A. S. HOUSEHOLDER, *Discussion of a set of points in terms of their mutual distances*, Psychometrika, 3 (1938), pp. 19–22.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 601–611, October 1990

010

SPECTRAL EVOLUTION OF A ONE-PARAMETER EXTENSION OF A REAL SYMMETRIC TOEPLITZ MATRIX*

WILLIAM F. TRENCH†

Abstract. Let $T_n = (t_{i-j})_{i,j=1}^n$ ($n \geqq 3$) be a real symmetric Toeplitz matrix such that T_{n-1} and T_{n-2} have no eigenvalues in common. The evolution of the spectrum of T_n as the parameter $t = t_{n-1}$ varies over $(-\infty, \infty)$ is considered. It is shown that the eigenvalues of T_n associated with symmetric (reciprocal) eigenvectors are strictly increasing functions of t, while those associated with the skew-symmetric (anti-reciprocal) eigenvectors are strictly decreasing. Results are obtained on the asymptotic behavior of the eigenvalues and eigenvectors at $t \to \pm\infty$, and on the possible orderings of eigenvalues associated with symmetric and skew-symmetric eigenvectors.

Key words. Toeplitz, extension, symmetric, eigenvalue, eigenvector

AMS(MOS) subject classifications. 15A18, 15A42

Following Andrew [1], we will say that an m-vector

$$X = [x_1, x_2, \cdots, x_m]$$

is *symmetric* if

$$x_j = x_{m-j+1}, \qquad 1 \leqq j \leqq m,$$

or *skew-symmetric* if

$$x_j = -x_{m-j+1}, \qquad 1 \leqq j \leqq m.$$

(Some authors call such vectors *reciprocal* and *anti-reciprocal*.) Cantoni and Butler [2] have shown that if

$$T_m = (t_{i-j})_{i,j=1}^m$$

is a real symmetric Toeplitz matrix of order m, then R^m has an orthonormal basis consisting of $m - [m/2]$ symmetric and $[m/2]$ skew-symmetric eigenvectors of T_m, where $[x]$ is the integer part of x. A related result of Delsarte and Genin [4, Thm. 8] is that if λ is an eigenvalue of T_m with multiplicity greater than one, then the λ-eigenspace of T_m has an orthonormal basis which splits as evenly as possible between symmetric and skew-symmetric λ-eigenvectors of T_m. For convenience here, we will say that an eigenvalue λ of T_m is *even* (*odd*) if T_m has a symmetric (skew-symmetric) λ-eigenvector. The collection $S^+(T_m)(S^-(T_m))$ of even (odd) eigenvalues will be called the *even* (*odd*) *spectrum of* T_m. From the result of Delsarte and Genin, a multiple eigenvalue is in both the even and odd spectra of T_m.

This paper is motivated by considerations that arose in connection with the inverse eigenvalue problem for real symmetric Toeplitz matrices. Although we do not claim that our results provide much insight into this problem, they may nevertheless be of some interest in their own right.

The inverse eigenvalue problem for real symmetric Toeplitz matrices is usually stated as follows: Find a real symmetric Toeplitz matrix T_m with given spectrum

$$S(T_m) = \{\lambda_1 \leqq \lambda_2 \leqq \cdots \leqq \lambda_m\}.$$

* Received by the editors January 11, 1989; accepted for publication August 29, 1989. This work was partially supported by National Science Foundation grant DMS 8707080.

† Department of Mathematics, Trinity University, 715 Stadium Drive, San Antonio, Texas 78212 (wtrench@trinity.bitnet).

For our purposes it is convenient to impose an additional condition, namely, that T_m have even and odd spectra $S^+(T_m)$ and $S^-(T_m)$, containing, respectively, $m - [m/2]$ and $[m/2]$ given elements (counting repeated eigenvalues according to their multiplicities) of S. We will say that $S^+(T_m)$ and $S^-(T_m)$ are *interlaced* if whenever λ_k and λ_l are in $S^+(T_m)(S^-(T_m))$ and $k < l$, there is an element λ_i in $S^-(T_m)(S^+(T_m))$ such that $\lambda_k \leqq \lambda_i \leqq \lambda_l$. Delsarte and Genin [4] showed that if $m \leqq 4$ then the inverse eigenvalue problem always has a solution (regardless of the numerical values of λ_1, λ_2, λ_3, and λ_4) if $S^+(T_m)$ and $S^-(T_m)$ are interlaced; however, if they are not, then the existence or nonexistence of a solution depends on the specific numerical values of the λ_i's. They also argued that this negative consequence of noninterlacement of $S^+(T_m)$ and $S^-(T_m)$ holds for all $m > 4$; that is, if the two desired spectra are not interlaced, then the inverse eigenvalue problem fails to have a solution for some choices of desired eigenvalues.

Delsarte and Genin [4] formulated the (still open) conjecture that the inverse eigenvalue problem always has a solution (for arbitrary m) provided that the desired even and odd spectra are interlaced. (This was apparently misinterpreted by Laurie [6], who cited a real symmetric Toeplitz matrix for which $S^+(T_m)$ and $S^-(T_m)$ are not interlaced as "a counterexample . . . to the conjecture of Delsarte and Genin that the eigenvectors of a symmetric Toeplitz matrix, corresponding to eigenvalues arranged in decreasing order, alternate between reciprocal and anti-reciprocal vectors.")

In numerical experiments reported in [7] we computed the eigenvalues of hundreds of randomly generated real symmetric Toeplitz matrices with orders up to 1,000. (Since then we have considered matrices of order 2,000.) The even and odd spectra of these matrices are certainly not necessarily interlaced, but they seem to be "almost interlaced," in that we seldom saw more than two or three successive even (or odd) eigenvalues. In unsuccessfully trying to formulate a definition of a measure of interlacement that would be useful in connection with the inverse eigenvalue problem, we were led to study the problem considered here; namely: If

$$T_{n-1} = \begin{bmatrix} t_0 & t_1 & \cdots & t_{n-2} \\ t_1 & t_0 & \cdots & t_{n-3} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-2} & t_{n-3} & \cdots & t_0 \end{bmatrix},$$

is a given real symmetric Toeplitz matrix of order $n - 1$, then how does the spectrum of the nth order matrix

$$T_n(t) = \begin{bmatrix} t_0 & t_1 & \cdots & t_{n-2} & t \\ t_1 & t_0 & \cdots & t_{n-3} & t_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{n-2} & t_{n-3} & \cdots & t_0 & t_1 \\ t & t_{n-2} & \cdots & t_1 & t_0 \end{bmatrix} \tag{1}$$

evolve as t varies over $(-\infty, \infty)$?

We impose the following assumption throughout.

Assumption A. $n \geqq 3$ and T_{n-2} and T_{n-1} have no eigenvalues in common.

Assumption A and the Cauchy interlace theorem imply that T_{n-2} and T_{n-1} have no repeated eigenvalues. Let

$$\alpha_1 < \alpha_2 < \cdots < \alpha_{n-1}$$

be the eigenvalues of T_{n-1}, and let

$$\lambda_1(t) \leqq \lambda_2(t) \leqq \cdots \leqq \lambda_n(t)$$

be the eigenvalues of $T_n(t)$. The Cauchy interlace theorem implies that

$$\lambda_i(t) \leqq \alpha_i \leqq \lambda_{i+1}(t), \quad 1 \leqq i \leqq n-1, \quad -\infty < t < \infty.$$

It is also convenient to introduce distinct names for the even and odd eigenvalues of T_{n-2} and $T_n(t)$. Define

$$r = n - [n/2] \quad \text{and} \quad s = [n/2];$$

thus $r = s$ if n is even and $r = s + 1$ if n is odd. Denote the even and odd eigenvalues of T_{n-2} by

$$\beta_1 < \beta_2 < \cdots < \beta_{r-1}$$

and

$$\gamma_1 < \gamma_2 < \cdots < \gamma_{s-1},$$

respectively, and let

$$\mu_1(t) \leqq \mu_2(t) \leqq \cdots \leqq \mu_r(t) \tag{2}$$

and

$$\nu_1(t) \leqq \nu_2(t) \leqq \cdots \leqq \nu_s(t) \tag{3}$$

be the even and odd eigenvalues, respectively, of $T_n(t)$.

Now define

$$p_j(\lambda) = \det(T_j - \lambda I_j), \qquad 1 \leqq j \leqq n-1,$$

and

$$p_n(\lambda, t) = \det(T_n(t) - \lambda I_n).$$

As observed by Delsarte and Genin [4], a result of Cantoni and Butler [2] implies that $p_n(\lambda, t)$ can be factored in the form

$$p_n(\lambda, t) = p_n^+(\lambda, t) p_n^-(\lambda, t),$$

where p_n^+ and p_n^- are of degrees r and s, respectively, in λ,

$$p_n^+(\mu_i(t), t) = 0, \qquad 1 \leqq i \leqq r, \quad -\infty < t < \infty, \tag{4}$$

and

$$p_n^-(\nu_j(t), t) = 0, \qquad 1 \leqq j \leqq s, \quad -\infty < t < \infty. \tag{5}$$

Moreover, an argument of Delsarte and Genin [4, pp. 203, 208] implies that the even (odd) eigenvalues of T_{n-2} separate the even (odd) eigenvalues of $T_n(t)$, i.e.,

$$\mu_i(t) \leqq \beta_i \leqq \mu_{i+1}(t), \qquad 1 \leqq i \leqq r-1, \tag{6}$$

and

$$\nu_i(t) \leqq \gamma_i \leqq \nu_{i+1}(t), \qquad 1 \leqq i \leqq s-1. \tag{7}$$

It now follows that (2) and (3) can be replaced by the stronger inequalities

$$\mu_1(t) < \mu_2(t) < \cdots < \mu_r(t)$$

and

$$\nu_1(t) < \nu_2(t) < \cdots < \nu_s(t).$$

To see this, suppose, for example, that $\mu_i(\hat{t}) = \mu_{i+1}(\hat{t})$ for some i and $\hat{t}$. Then (6) implies that β_i, an eigenvalue of T_{n-2}, is a repeated eigenvalue of $T_n(\hat{t})$. Cauchy's theorem then implies that β_i is also an eigenvalue of T_{n-1}, which violates Assumption A.

Since $p_n^+(\lambda, t)$ and $p_n^-(\lambda, t)$ have distinct roots for all t, (4)–(7) define $\mu_1(t), \cdots, \mu_r(t)$ and $\nu_1(t), \cdots, \nu_s(t)$ as continuously differentiable functions on $(-\infty, \infty)$. However, (4) and (5) do not provide convenient representations for the derivatives of these functions. The next two lemmas will enable us to find such representations.

LEMMA 1. *Suppose that Assumption* A *holds, and let* α_i $(1 \leqq i \leqq n-1)$ *be an eigenvalue of* T_{n-1}. *Then there is exactly one value* τ_i *of* t *such that* α_i *is an eigenvalue of* $T_n(\tau_i)$. *Moreover,* α_i *is in fact an eigenvalue of* $T_n(\tau_i)$ *with multiplicity two, and* $\tau_1, \cdots, \tau_{n-1}$ *are the only values of* t *for which* $T_n(t)$ *has repeated eigenvalues.*

Proof. By an argument of Iohvidov [5, p. 98], based on Sylvester's identity, it can be shown that

$$(8) \qquad p_n(\lambda,t)p_{n-2}(\lambda)=p_{n-1}^2(\lambda)-\begin{vmatrix} t_1 & t_2 & t_3 & \cdots & t_{n-2} & t \\ t_0-\lambda & t_1 & t_2 & \cdots & t_{n-3} & t_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{n-3} & t_{n-4} & t_{n-5} & \cdots & t_0-\lambda & t_1 \end{vmatrix}^2$$

for all λ and t. Expanding the determinant on the right in cofactors of its first row shows that (8) can be rewritten as

$$(9) \qquad p_n(\lambda,t)p_{n-2}(\lambda)=p_{n-1}^2(\lambda)-[(-1)^{n+1}p_{n-2}(\lambda)t+k_{n-2}(\lambda)]^2,$$

where $k_{n-2}(\lambda)$ is independent of t. Therefore, $p_n(\alpha_i, \tau_i) = 0$ if and only if

$$\tau_i=\frac{(-1)^n k_{n-2}(\alpha_i)}{p_{n-2}(\alpha_i)}.$$

Obviously, α_i is a repeated zero of the polynomial obtained by setting $t = \tau_i$ on the right of (9), and therefore α_i is an eigenvalue of $T_n(\tau_i)$ with multiplicity $m > 1$. To see that $m = 2$, suppose to the contrary that $m > 2$. Then either $\mu_l(\tau_i) = \mu_{l+1}(\tau_i) = \alpha_i$ or $\nu_l(\tau_i) = \nu_{l+1}(\tau_i) = \alpha_i$ for some l. But then (6) and (7) imply that $\alpha_i = \beta_l$ or $\alpha_i = \gamma_l$ for some l, which contradicts Assumption A; hence, $m = 2$. To conclude the proof, we simply observe that a repeated eigenvalue of $T_n(t)$ must be an eigenvalue of T_{n-1}.

This lemma is related to Theorem 3 of Cybenko [3], who also considered questions connected with the eigenstructure of $T_n(t)$ regarded as an extension of T_{n-1}.

Now define

$$q_n(\lambda,t)=\frac{p_n(\lambda,t)}{p_{n-1}(\lambda)}.$$

The next lemma can be proved by partitioning $T_n(t) - \lambda I_n$ in the form

$$T_n(t)-\lambda I_n=\begin{bmatrix} t_0-\lambda & U_{n-1}^T(t) \\ U_{n-1}(t) & T_{n-1}-\lambda I_{n-1} \end{bmatrix},$$

where $U_{n-1}(t)$ is defined below in (11). (For details, see the proof of Theorem 1 of [7].)

LEMMA 2. *If* λ *is not an eigenvalue of* T_{n-1}, *let*

$$X_{n-1}(\lambda,t)=\begin{bmatrix} x_{1,n-1}(\lambda,t) \\ x_{2,n-1}(\lambda,t) \\ \vdots \\ x_{n-1,n-1}(\lambda,t) \end{bmatrix}$$

be the solution of the system

$$(T_{n-1}-\lambda I_{n-1})X_{n-1}(\lambda,t)=U_{n-1}(t), \tag{10}$$

where

$$U_{n-1}(t)=\begin{bmatrix} t_1 \\ \vdots \\ t_{n-2} \\ t \end{bmatrix}. \tag{11}$$

Then

$$q_n(\lambda,t)=t_0-\lambda-U_{n-1}^T(t)X_{n-1}(\lambda,t);$$

moreover, if $q_n(\lambda, t) = 0$, *then the vector*

$$Y_n(\lambda,t)=\begin{bmatrix} -1 \\ X_{n-1}(\lambda,t) \end{bmatrix}$$

is a λ*-eigenvector of* $T_n(t)$; *hence*

$$x_{n-1,n-1}(\lambda,t)=(-1)^{q+1}, \tag{12}$$

where

$$q=\begin{cases} 0 & \text{if } \lambda \text{ is an even eigenvalue of } T_{n-1}(t), \\ 1 & \text{if } \lambda \text{ is an odd eigenvalue of } T_{n-1}(t). \end{cases}$$

We will call q the *parity* of the eigenvalue λ.

Now suppose that $\lambda(t)$ is one of the functions $\mu_1(t), \cdots, \mu_r(t)$ or $\nu_1(t), \cdots, \nu_s(t)$. Lemmas 1 and 2 imply that

$$t_0-\lambda(t)-U_{n-1}^T(t)X_{n-1}(\lambda(t),t)=0, \qquad t\in J,$$

where J is any interval which does not contain any of the exceptional points $\tau_1, \cdots, \tau_{n-1}$ defined in Lemma 1. Differentiating this yields

$$\left(1+U_{n-1}^T(t)\frac{\partial}{\partial\lambda}X_{n-1}(\lambda(t),t)\right)\lambda'(t)+\frac{\partial U_{n-1}^T(t)}{\partial t}X_{n-1}(\lambda(t),t) + U_{n-1}^T(t)\frac{\partial}{\partial t}X_{n-1}(\lambda(t),t)=0. \tag{13}$$

However, if λ is any number which is not an eigenvalue of T_{n-1}, then

$$U_{n-1}^T(t)=X_{n-1}^T(\lambda,t)(T_{n-1}-\lambda I_{n-1}) \tag{14}$$

(see (10)), and differentiating (10) yields

$$(T_{n-1}-\lambda I_{n-1})\frac{\partial}{\partial\lambda}X_{n-1}(\lambda,t)=X_{n-1}(\lambda,t) \tag{15}$$

and

$$(T_{n-1}-\lambda I_{n-1})\frac{\partial}{\partial t}X_{n-1}(\lambda,t)=\frac{\partial}{\partial t}U_{n-1}(t)=\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{16}$$

(see (11)). Setting $\lambda = \lambda(t)$ in (14)–(16) and substituting the results into (13) yields

$$(1+\|X_{n-1}(\lambda(t),t)\|^2)\lambda'(t)+2x_{n-1,n-1}(\lambda(t),t)=0$$

(Euclidean norm); therefore, from (12),

$$\lambda'(t)=\frac{(-1)^q 2}{1+\|X_{n-1}(\lambda(t),t)\|^2}. \tag{17}$$

Because of (12) we can write

$$X_{n-1}(\lambda(t),t)=\begin{bmatrix}\hat{X}_{n-2}(\lambda(t),t)\\ (-1)^{q+1}\end{bmatrix}, \tag{18}$$

where q is the parity of $\lambda(t)$ and $\hat{X}_{n-2}(\lambda(t), t)$ is symmetric if $q = 0$ or skew-symmetric if $q = 1$; then (17) becomes

$$\lambda'(t)=\frac{(-1)^q 2}{2+\|\hat{X}_{n-2}(\lambda(t),t)\|^2}, \tag{19}$$

which is valid for $t \neq \tau_i$, $1 \leqq i \leqq n-1$. This formula does not yet apply at these exceptional points, simply because the vectors $\hat{X}_{n-2}(\lambda(\tau_i), \tau_i) = \hat{X}_{n-2}(\alpha_i, \tau_i)$, $1 \leqq i \leqq n-1$ are as yet undefined. This is easily remedied; by Lemma 2,

$$(T_n(t)-\lambda(t)I_n)\begin{bmatrix}-1\\ \hat{X}_{n-2}(\lambda(t),t)\\ (-1)^{q+1}\end{bmatrix}=0 \tag{20}$$

for all $t \neq \tau_i$, $1 \leqq i \leqq n-1$. This and (1) imply that

$$(T_{n-2}-\lambda(t)I_{n-2})\hat{X}_{n-2}(\lambda(t),t)=\begin{bmatrix}t_1\\ t_2\\ \vdots\\ t_{n-2}\end{bmatrix}+(-1)^q\begin{bmatrix}t_{n-2}\\ t_{n-3}\\ \vdots\\ t_1\end{bmatrix} \tag{21}$$

for all $t \neq \tau_i$, $1 \leqq i \leqq n-1$. However, this system has a unique solution when $t = \tau_i$, since the matrix $T_{n-2} - \lambda(\tau_i)I_{n-2} = T_{n-2} - \alpha_i I_{n-2}$ is nonsingular by Assumption A. Defining this solution to be $\hat{X}_{n-2}(\lambda(\tau_i), \tau_i)$ extends $\hat{X}_{n-2}(\lambda(t), t)$ so as to make it continuous on $(-\infty, \infty)$. Since $\lambda'(t)$ is also continuous for all t, (19) must hold for all t.

For future reference, note from (12) and the continuity of $x_{n-1,n-1}(\lambda_i(t), t)$ that the parity $q_i(t)$ of $\lambda_i(t)$ is constant on any interval J which does not contain any of the exceptional points $\tau_1, \cdots, \tau_{n-1}$.

THEOREM 1. *The even eigenvalues* $\mu_1(t), \cdots, \mu_r(t)$ *are strictly increasing on* $(-\infty, \infty)$ *and the inequalities* (6) *can be replaced by the strict inequalities*

$$\mu_i(t)<\beta_i<\mu_{i+1}(t), \quad -\infty<t<\infty, \quad 1\leqq i\leqq r-1; \tag{22}$$

moreover,

$$\lim_{t\to\infty}\mu_i(t)=\begin{cases}\beta_i, & 1\leqq i\leqq r-1,\\ \infty, & i=r,\end{cases} \tag{23}$$

and

$$\lim_{t\to-\infty}\mu_i(t)=\begin{cases}\beta_{i-1}, & 2\leqq i\leqq r,\\ -\infty, & i=1.\end{cases} \tag{24}$$

Proof. Setting $q = 0$ in (17) shows that $\mu_1(t), \cdots, \mu_r(t)$ are strictly increasing for all t; therefore, (6) implies (22). For convenience, define $\beta_r = \infty$ and suppose that

$$\lim_{t \to \infty} \mu_i(t) = \zeta_i < \beta_i \tag{25}$$

for some i in $\{1, \cdots, r\}$. Since $\beta_{i-1} < \zeta_i < \beta_i$, the system

$$(T_{n-2} - \zeta_i I_{n-2})\tilde{X}_i = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{n-2} \end{bmatrix} + (-1)^q \begin{bmatrix} t_{n-2} \\ t_{n-3} \\ \vdots \\ t_1 \end{bmatrix}$$

has a unique solution, and, from (21) with $\lambda(t) = \mu_i(t)$,

$$\lim_{t \to \infty} \hat{X}_{n-2}(\mu_i(t), t) = \tilde{X}_i.$$

Consequently, (19) implies that

$$\lim_{t \to \infty} \mu_i'(t) = \frac{2}{2 + \|\tilde{X}_i\|^2} > 0,$$

and therefore $\lim_{t \to \infty} \mu_i(t) = \infty$, which contradicts (25). This implies (23). A similar argument implies (24).

The proof of the next theorem is similar to this.

THEOREM 2. *The odd eigenvalues* $\nu_1(t), \cdots, \nu_s(t)$ *are strictly decreasing on* $(-\infty, \infty)$ *and the inequalities* (7) *can be replaced by the strict inequalities*

$$\nu_i(t) < \gamma_i < \nu_{i+1}(t), \quad -\infty < t < \infty, \quad 1 \leqq i \leqq s-1;$$

moreover,

$$\lim_{t \to \infty} \nu_i(t) = \begin{cases} \gamma_{i-1}, & 2 \leqq i \leqq s, \\ -\infty, & i = 1, \end{cases} \tag{26}$$

and

$$\lim_{t \to -\infty} \nu_i(t) = \begin{cases} \gamma_i, & 1 \leqq i \leqq s-1, \\ \infty, & i = s. \end{cases} \tag{27}$$

The remaining theorems deal with the asymptotic behavior of the vectors $\hat{X}_{n-2}(\lambda(t), t)$ (see (18)) and with the orders of convergence in (23), (24), (26), and (27).

THEOREM 3. *Let*

$$A_i = \begin{bmatrix} a_1^{(i)} \\ a_2^{(i)} \\ \vdots \\ a_{n-2}^{(i)} \end{bmatrix}$$

be the β_i*-eigenvector of* T_{n-2} *which is normalized so that*

$$t_{n-2}a_1^{(i)} + t_{n-3}a_2^{(i)} + \cdots + t_1 a_{n-2}^{(i)} = 1.$$

Then

$$\lim_{t \to \infty} \frac{\hat{X}_{n-2}(\mu_i(t), t)}{t} = A_i, \qquad 1 \leqq i \leqq r-1, \tag{28}$$

and

$$\mu_i(t)=\beta_i-\frac{2(1+o(1))}{\|A_i\|^2t},\quad t\to\infty,\quad 1\le i\le r-1. \tag{29}$$

Also,

$$\lim_{t\to-\infty}\frac{\hat{X}_{n-2}(\mu_i(t),t)}{t}=A_{i-1},\qquad 2\le i\le r, \tag{30}$$

and

$$\mu_i(t)=\beta_{i-1}-\frac{2(1+o(1))}{\|A_{i-1}\|^2t},\quad t\to-\infty,\quad 2\le i\le r. \tag{31}$$

Proof. It is easy to verify that the vector

$$\begin{bmatrix}A_i\\0\end{bmatrix}$$

is the last column of $(T_{n-1}-\beta_iI_{n-1})^{-1}$. Setting $\lambda=\mu_i(t)$ in (10) shows that

$$X_{n-1}(\mu_i(t),t)=(T_{n-1}-\mu_i(t)I_{n-1})^{-1}U_{n-1}(t)$$

for $|t|$ sufficiently large. Therefore, (11), (23), and (24) imply (28) and (30). From (19) with $\lambda(t)=\mu_i(t)$ and (28)

$$\mu_i'(t)=\frac{2(1+o(1))}{\|A_i\|^2t^2},\quad t\to\infty,\quad 1\le i\le r-1. \tag{32}$$

Similarly, (19) and (30) imply that

$$\mu_i'(t)=\frac{2(1+o(1))}{\|A_{i-1}\|^2t^2},\quad t\to-\infty,\quad 2\le i\le r. \tag{33}$$

Since (32) and (33) imply (29) and (31), the proof is complete.

A similar argument yields the following theorem.

THEOREM 4. *Let*

$$B_i=\begin{bmatrix}b_1^{(i)}\\b_2^{(i)}\\\vdots\\b_{n-2}^{(i)}\end{bmatrix}$$

be the γ_i-eigenvector of T_{n-2} which is normalized so that

$$t_{n-2}b_1^{(i)}+t_{n-3}b_2^{(i)}+\cdots+t_1b_{n-2}^{(i)}=1.$$

Then

$$\lim_{t\to\infty}\frac{\hat{X}_{n-2}(\nu_i(t),t)}{t}=B_{i-1},\qquad 2\le i\le s,$$

and

$$\nu_i(t)=\gamma_{i-1}+\frac{2(1+o(1))}{\|B_{i-1}\|^2t},\quad t\to\infty,\quad 2\le i\le s.$$

Also,

$$\lim_{t\to-\infty}\frac{\hat{X}_{n-2}(\nu_i(t),t)}{t}=B_i,\qquad 1\leqq i\leqq s-1,$$

and

$$\nu_i(t)=\gamma_i+\frac{2(1+o(1))}{\|B_i\|^2 t},\quad t\to-\infty,\quad 1\leqq i\leqq s-1.$$

Theorems 3 and 4 provide no information on the asymptotic behavior of the eigenvalues which tend to infinite limits as $t\to\pm\infty$. The next theorem fills this gap.

THEOREM 5. *Let*

$$\Gamma_q=\begin{bmatrix}t_1\\t_2\\\vdots\\t_{n-2}\end{bmatrix}+(-1)^q\begin{bmatrix}t_{n-2}\\t_{n-3}\\\vdots\\t_1\end{bmatrix}. \tag{34}$$

Then

$$\mu_r(t)=t+t_0+\frac{\|\Gamma_0\|^2}{2t}+o\left(\frac{1}{t}\right)\quad(t\to\infty), \tag{35}$$

$$\mu_1(t)=t+t_0+\frac{\|\Gamma_0\|^2}{2t}+o\left(\frac{1}{t}\right)\quad(t\to-\infty), \tag{36}$$

$$\lim_{t\to\infty}t\hat{X}_{n-2}(\mu_r(t),t)=\lim_{t\to-\infty}t\hat{X}_{n-2}(\mu_1(t),t)=-\Gamma_0, \tag{37}$$

$$\nu_1(t)=-t+t_0-\frac{\|\Gamma_1\|^2}{2t}+o\left(\frac{1}{t}\right)\quad(t\to\infty), \tag{38}$$

$$\nu_s(t)=-t+t_0-\frac{\|\Gamma_1\|^2}{2t}+o\left(\frac{1}{t}\right)\quad(t\to-\infty), \tag{39}$$

and

$$\lim_{t\to\infty}t\hat{X}_{n-2}(\nu_1(t),t)=\lim_{t\to-\infty}t\hat{X}_{n-2}(\nu_s(t),t)=\Gamma_1. \tag{40}$$

Proof. We will prove (35) and (38) and verify the first limits in (37) and (40); the proof of (36) and (39) and the verification of the second limits in (37) and (40) are similar. Let $\lambda(t)=\mu_r(t)$ and $q=0$ or $\lambda(t)=\nu_1(t)$ and $q=1$. We know from Theorems 1 and 2 that

$$\lim_{t\to\infty}|\lambda(t)|=\infty. \tag{41}$$

From (21) and (34),

$$(|\lambda(t)|-\|T_{n-2}\|)\|\hat{X}_{n-2}(\lambda(t),t)\|\leqq\|\Gamma_q\|;$$

therefore, (41) implies that

$$\lim_{t\to\infty}\|\hat{X}_{n-2}(\lambda(t),t)\|=0.$$

From this and (19),

$$\lim_{t\to\infty}\lambda'(t)=(-1)^q,$$

and therefore

$$\lim_{t\to\infty}\frac{\lambda(t)}{t}=(-1)^q,$$

by l'Hôpital's rule. Now (21) implies that

$$\lim_{t\to\infty}t\hat{X}_{n-2}(\lambda(t),t)=(-1)^{q+1}\Gamma_q, \tag{42}$$

with Γ_q as in (34). This verifies the first limits in (37) and (40). Since the first component of the vector on the left of (20) is identically zero,

$$\lambda(t)-t_0+[t_1,t_2,\cdots,t_{n-2}]\hat{X}_{n-2}(\lambda(t),t)+(-1)^{q+1}t=0. \tag{43}$$

From (35) and (42),

$$\begin{aligned}[t_1,t_2,\cdots,t_{n-2}]\hat{X}_{n-2}(\lambda(t),t)&=(-1)^{q+1}\frac{[t_1,t_2,\cdots,t_{n-2}]\Gamma_q}{t}+o\left(\frac{1}{t}\right)\\&=(-1)^{q+1}\frac{\|\Gamma_q\|^2}{2t}+o\left(\frac{1}{t}\right).\end{aligned}$$

Substituting this into (43) and solving for $\lambda(t)$ yields

$$\lambda(t)=t_0+(-1)^q\left[t+\frac{\|\Gamma_q\|^2}{2t}+o\left(\frac{1}{t}\right)\right],$$

which proves (35) and (38).

We conclude with a comment on the possible orderings of even and odd eigenvalues of $T_n(t)$. Let the eigenvalues of T_{n-2} be

$$\omega_1<\omega_2<\cdots<\omega_{n-2},$$

i.e.,

$$\{\omega_1,\cdots,\omega_{n-2}\}=\{\beta_1,\cdots,\beta_{r-1}\}\cup\{\gamma_1,\cdots,\gamma_{s-1}\}.$$

Define

$$Q_0=[q(\omega_1),q(\omega_2),\cdots,q(\omega_{n-2})],$$

where $q(\omega_i)$ is the parity of ω_i. Suppose that the elements $\tau_1,\cdots,\tau_{n-1}$ of the exceptional set discussed in Lemma 1 are distinct and ordered so that

$$\tau_{i_1}<\tau_{i_2}<\cdots<\tau_{i_{n-1}}. \tag{44}$$

Let

$$J_l=\begin{cases}(-\infty,\tau_{i_1}), & l=1,\\(\tau_{i_{l-1}},\tau_{i_l}), & 2\leqq l\leqq n-1,\\(\tau_{i_l},\infty), & l=n.\end{cases}$$

The eigenvalues of $T_n(t)$ satisfy the strict inequalities

$$\lambda_1(t) < \lambda_2(t) < \cdots < \lambda_n(t)$$

for all t in each interval $J_1, \cdots, J_{n-1}$. Recalling that the parities of $\lambda_1(t), \cdots, \lambda_n(t)$ are constant on each J_l, we can define the n-vectors

$$Q_l = [q_{l1}, q_{l2}, \cdots, q_{ln}], \qquad 1 \leqq l \leqq n,$$

where q_{lj} is the parity of $\lambda_j(t)$ on J_l. Since $\lambda_i(t) \leqq \alpha_i \leqq \lambda_{i+1}(t)$ for all t and α_i is an eigenvalue with multiplicity two of $T_n(\tau_i)$, we must have $\lambda_i(\tau_i) = \lambda_{i+1}(\tau_i) = \alpha_i$. Therefore, α_i is in both the even and odd spectrum of $T_n(\tau_i)$. From the monotonicity properties of the even and odd eigenvalues of $T_n(t)$, it follows that $\lambda_i(t)$ changes from even to odd and $\lambda_{i+1}(t)$ changes from odd to even as t increases through τ_i. This and (23), (24), (26), and (27) imply the following theorem.

THEOREM 6. *If* $\tau_1, \cdots, \tau_{n-1}$ *satisfy* (44), *then*

$$Q_1 = [0, Q_0, 1], \qquad Q_n = [1, Q_0, 0],$$

and, for $1 \leqq l \leqq n-1$,

$$q_{i,l+1} = q_{i,l} \quad \text{if } i \neq i_l \quad \textit{and} \quad i \neq i_{l+1}, \tag{45}$$

and

$$q_{i_l,l} = 0, \quad q_{i_l+1,l} = 1, \quad q_{i_l,l+1} = 1, \quad \textit{and} \quad q_{i_l+1,l+1} = 0. \tag{46}$$

From (45) and (46), Q_{l+1} is obtained by interchanging the zero and one which must be in columns i_l and i_{l+1}, respectively, of Q_l.

The assumption that $\tau_1, \cdots, \tau_{n-1}$ are distinct was imposed for simplicity. Theorem 6 can easily be modified to cover the exceptional case where $\{\tau_1, \cdots, \tau_{n-1}\}$ contains fewer than $n-1$ distinct elements.

REFERENCES

[1] A. L. ANDREW, *Eigenvectors of certain matrices*, Linear Algebra Appl., 7 (1973), pp. 151–162.

[2] A. CANTONI AND F. BUTLER, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl., 13 (1976), pp. 275–288.

[3] G. CYBENKO, *On the eigenstructure of Toeplitz matrices*, IEEE Trans. Acoustics Speech Signal Process., 32 (1984), pp. 275–288.

[4] 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.

[5] I. S. IOHVIDOV, *Hankel and Toeplitz Matrices and Forms*, Birkhauser, Boston, Basel, Stuttgart, 1982.

[6] D. P. LAURIE, *A numerical approach to the inverse Toeplitz eigenproblem*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 431–436.

[7] W. F. TRENCH, *Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 135–146.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 612–619, October 1990

011

POSITIVE SEMIDEFINITE PATTERN DECOMPOSITIONS*

DANIEL HERSHKOWITZ†

Abstract. It is shown that every positive semidefinite matrix in a block tridiagonal form with square diagonal blocks can be written as a sum of positive semidefinite matrices with complementary off-diagonal block patterns. A similar result holds for completely positive matrices and, under a certain condition, for doubly nonnegative matrices.

Key words. positive semidefinite matrix, pattern, doubly nonnegative matrix, completely positive matrix

AMS(MOS) subject classifications. 15, 05

1. Introduction. In this paper we discuss complex square matrices A that have the block pattern

$$\left(\begin{array}{c|c|c} X & X & O \\ \hline X & X & X \\ \hline O & X & X \end{array}\right), \tag{1.1}$$

where the diagonal blocks are square, and where X denotes a possibly nonzero block.

The notation and definitions for this paper are presented in § 2.

In § 3 we show that if A is positive semidefinite, then A can be written as a sum of two positive semidefinite matrices E and F that have the patterns

$$\left(\begin{array}{c|c|c} X & X & O \\ \hline X & X & O \\ \hline O & O & O \end{array}\right), \qquad \left(\begin{array}{c|c|c} O & O & O \\ \hline O & X & X \\ \hline O & X & X \end{array}\right),$$

respectively. If A is real, then E and F can be chosen to be real. These results, as well as the results of the following sections, can be extended to block tridiagonal forms. However, we show that these results cannot be generalized to general block patterns.

The case that A is a doubly nonnegative (nonnegative positive semidefinite) matrix is discussed in § 4. In general, in this case we cannot choose the matrices E and F to be doubly nonnegative, too. However, if the second block in the diagonal of the form (1.1) is a 1×1 block, then E and F must be doubly nonnegative.

Section 5 is devoted to a subclass of the doubly nonnegative matrices (that is, completely positive matrices). It is shown that if A is completely positive then E and F can be chosen to be completely positive. This result is applied in proving a result on completely positive graphs, originally proven in [3].

2. Notation and definitions.

Notation 2.1. For a positive integer n we denote by $\langle n \rangle$ the set $\{1, \cdots, n\}$.

Notation 2.2. For a set α we denote by $|\alpha|$ the cardinality of α.

DEFINITION 2.3. Let $\mathbf{x}$ be an n-dimensional vector. The set $\{i \in \langle n \rangle : x_i \neq 0\}$ is called the *support* of $\mathbf{x}$.

* Received by the editors April 16, 1989; accepted for publication June 23, 1989. This research was supported in part by the Technion VPR grant 100-0754 (M. & M. Bank Mathematics Research Fund).

† Mathematics Department, Technion–Israel Institute of Technology, Haifa 32000, Israel (MAR23AA@TECHNION.BITNET).

Notation 2.4. Let A be an $n \times n$ matrix, and let α and β be subsets of $\langle n \rangle$. We denote

$A[\alpha|\beta]$ – the submatrix of A whose rows are indexed by α and whose columns are indexed by β in their natural order;

$A[\alpha|\beta) = A[\alpha|\langle n \rangle \setminus \beta]$ (granted that $\beta \neq \langle n \rangle$);

$A[\alpha] = A[\alpha|\alpha]$.

DEFINITION 2.5. A positive semidefinite matrix is said to be *doubly nonnegative* if it is nonnegative entrywise.

DEFINITION 2.6. A doubly nonnegative matrix A is said to be *completely positive* if there exists a (not necessarily square) nonnegative matrix B such that $A = BB^T$.

DEFINITION 2.7. Let A be a square matrix in a 3×3 block form $(A_{ij})_1^3$ with square diagonal blocks, and such that $A_{13} = 0$, $A_{31} = 0$. Let $A = E + F$, where E and F are block matrices (partitioned conformably with A), such that the blocks of E and F satisfy

$$E_{13}, E_{23}, E_{33}, E_{31}, E_{32} = 0; \qquad F_{11}, F_{12}, F_{13}, F_{21}, F_{31} = 0. \tag{2.8}$$

Then $A = E + F$ is said to be a *pattern decomposition of* A. If E and F are real, then $A = E + F$ is said to be a *real pattern decomposition of* A.

Remark 2.9. Obviously, the matrices E and F in Definition (2.7) satisfy

$$E_{11} = A_{11}, \qquad E_{12} = A_{12}, \qquad E_{21} = A_{21},$$

$$F_{23} = A_{23}, \qquad F_{32} = A_{32}, \qquad F_{33} = A_{33},$$

$$E_{22} + F_{22} = A_{22}.$$

DEFINITION 2.10. Let A be a positive semidefinite matrix in a 3×3 block form $(A_{ij})_1^3$ with square diagonal blocks, and such that $A_{13} = 0$.

(i) A pattern decomposition $A = E + F$ of A is said to be a *positive semidefinite pattern decomposition* of A if E and F are both positive semidefinite.

(ii) A positive semidefinite pattern decomposition $A = E + F$ of A is said to be *doubly nonnegative pattern decomposition* of A if E and F are both doubly nonnegative.

(iii) A doubly nonnegative pattern decomposition $A = E + F$ of A is said to be a *completely positive pattern decomposition* of A if E and F are both completely positive.

Notation 2.11. For a (directed) graph G we denote by $E(G)$ and $V(G)$ the arc set of G and the vertex set of G, respectively.

DEFINITION 2.12. A graph G_1 is said to be a *subgraph* of a graph G_2 if $E(G_1) \subseteq E(G_2)$ and $V(G_1) \subseteq V(G_2)$.

DEFINITION 2.13. A graph G is said to be the *union* $G_1 \cup G_2$ of graphs G_1 and G_2 if $E(G) = E(G_1) \cup E(G_2)$ and $V(G) = V(G_1) \cup V(G_2)$.

DEFINITION 2.14. Two graphs are said to be *essentially* the same if they have the same arc set. For example, the graphs

$$1 \bullet \rightarrow \bullet 2 \quad \bullet 3 \quad \text{and} \quad 1 \bullet \rightarrow \bullet 2$$

are essentially the same.

DEFINITION 2.15. Let A be an $n \times n$ matrix. The *graph* $G(A)$ of A is defined to be the graph with vertex set $\langle n \rangle$, and such that there is an arc from i to j if $a_{ij} \neq 0$.

DEFINITION 2.16. *A graph* G is said to be *completely positive* if every doubly nonnegative matrix A with $G(A) = G$ is completely positive.

We remark that the completely positive graphs are characterized in [3].

Convention 2.17. By "positive semidefinite matrix" we mean "complex positive semidefinite Hermitian matrix."

3. Positive semidefinite pattern decompositions. In this section we prove that every positive semidefinite matrix that has the block form (1.1) has a positive semidefinite pattern decomposition. This result can be extended to block tridiagonal forms. However, we show that it cannot be generalized to general block patterns.

THEOREM 3.1. *Let A be a positive semidefinite matrix in a 3×3 block form $(A_{ij})_1^3$ with square diagonal blocks, and such that $A_{13} = 0$. Then A has a positive semidefinite pattern decomposition. Furthermore, if A is real then A has a real positive semidefinite pattern decomposition.*

Proof. First we consider the complex case. Let α, β, γ be the subsets of $\langle n \rangle$ such that $A_{11} = A[\alpha]$, $A_{22} = A[\beta]$, and $A_{33} = A[\gamma]$. Since A is positive semidefinite, it follows that there exists a complex $n \times n$ matrix B such that $A = BB^*$. Let $B_1 = B[\alpha|\langle n \rangle]$, $B_2 = B[\beta|\langle n \rangle]$, $B_3 = B[\gamma|\langle n \rangle]$. We have $B_1 B_3^* = A_{13} = 0$. Let V be the row space of B_3, and let $V^\perp$ be the orthogonal complement of V in $\mathbb{C}^n$. Clearly, every row of B_2 can be written as a sum of a vector in V and a vector in $V^\perp$. Accordingly, we write $B_2 = C_1 + C_2$ where the rows of C_1 are elements of V and the rows of C_2 are elements of $V^\perp$. We have $C_1 C_2^* = 0$ and $C_2 B_3^* = 0$. Also, the equality $B_3 B_1^* = 0$ yields that $C_1 B_1^* = 0$. Consequently, we have

$$B_2 B_1^* = C_2 B_1^*, \qquad B_2 B_2^* = C_1 C_1^* + C_2 C_2^*, \qquad B_2 B_3^* = C_1 B_3^*. \tag{3.2}$$

Now, let A_1 be the $n \times n$ matrix defined by $A_1[\alpha|\langle n \rangle] = B_1$, $A_1[\beta|\langle n \rangle] = C_2$, $A_1[\gamma|\langle n \rangle] = 0$, and let A_2 be the $n \times n$ matrix defined by $A_2[\alpha|\langle n \rangle] = 0$, $A_2[\beta|\langle n \rangle] = C_1$, $A_2[\gamma|\langle n \rangle] = B_3$. In view of (3.2) we now have

$$A = BB^* = \begin{pmatrix} B_1 B_1^* & B_1 B_2^* & 0 \\ B_2 B_1^* & B_2 B_2^* & B_2 B_3^* \\ 0 & B_3 B_2^* & B_3 B_3^* \end{pmatrix} = \begin{pmatrix} B_1 B_1^* & B_1 C_2^* & 0 \\ C_2 B_1^* & C_1 C_1^* + C_2 C_2^* & C_1 B_3^* \\ 0 & B_3 C_1^* & B_3 B_3^* \end{pmatrix}$$

$$= \begin{pmatrix} B_1 B_1^* & B_1 C_2^* & 0 \\ C_2 B_1^* & C_2 C_2^* & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & C_1 C_1^* & C_1 B_3^* \\ 0 & B_3 C_1^* & B_3 B_3^* \end{pmatrix} = A_1 A_1^* + A_2 A_2^*.$$

The positive semidefinite matrices $E = A_1 A_1^*$ and $F = A_2 A_2^*$ satisfy conditions (2.8), and hence $A = E + F$ is a positive semidefinite pattern decomposition of A.

If A is real then, as is well known, there exists a real $n \times n$ matrix B such that $A = BB^T$. The proof continues as in the complex case, where we take $\mathbb{R}^n$ instead of $\mathbb{C}^n$. □

We obtain the following result concerning tridiagonal pattern decompositions, as a corollary.

COROLLARY 3.3. *Let A be a positive semidefinite matrix in a $p \times p$ block tridiagonal form with square diagonal blocks, and let $k \in \langle p - 1 \rangle$. Then A can be written as a sum of positive semidefinite block tridiagonal matrices E and F (partitioned conformably with A), where the blocks of E and F satisfy*

$$E_{ij} = 0, \quad \{i,j\} \not\subseteq \langle k \rangle; \qquad F_{ij} = 0, \quad \{i,j\} \not\subseteq \{k, \cdots, n\}.$$

Furthermore, if A is real then the matrices E and F can be chosen to be real.

Proof. Let $\omega_1, \cdots, \omega_p$ be the subsets of $\langle n \rangle$ such that $A_{ii} = A[\omega_i]$, $i \in \langle p \rangle$. Let $\alpha = \cup_{i=1}^{k-1} \omega_i$, $\beta = \omega_k$, and $\gamma = \cup_{i=k+1}^{n} \omega_i$. Partition A in a 3×3 block form $(A_{ij})_1^3$,

where $A_{11} = A[\alpha]$, $A_{22} = A[\beta]$, and $A_{33} = A[\gamma]$. Observe that $A_{13} = 0$. In view of Remark 2.9, our claim follows immediately from Theorem 3.1. □

Theorem 3.1 and Corollary 3.3 raise the following natural question concerning general positive semidefinite pattern decompositions: Let A be a positive semidefinite matrix in an $r \times r$ block form with square diagonal blocks. Can we write the matrix A as a sum of positive semidefinite block matrices E and F (partitioned conformably with A), where E and F are not block diagonal, and where the blocks of E and F satisfy

$$(3.4) \qquad E_{ij} \neq 0 \Rightarrow F_{ij} = 0; \qquad F_{ij} \neq 0 \Rightarrow E_{ij} = 0, \qquad i,j \in \langle r \rangle, i \neq j.$$

In general, the answer to this question is negative, as demonstrated by the following example.

Example 3.5. Let A be the matrix

$$\left(\begin{array}{c|c|c|c|c} 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 2 & 1 & 0 & 0 \\ \hline 0 & 1 & 2 & 0 & 1 \\ \hline 1 & 0 & 0 & 3 & 1 \\ \hline 0 & 0 & 1 & 1 & 2 \end{array}\right).$$

The matrix A is permutationally similar to a matrix discussed in Example 3.12 in [1]. As is shown in [1], A is positive semidefinite. We now refer to A as a 5×5 block matrix, and we assume that A can be written as a sum of positive semidefinite block matrices E and F (partitioned conformably with A), where E and F are not block diagonal, and where the blocks of E and F satisfy (3.4). Then necessarily E and F are nonnegative and hence doubly nonnegative. Furthermore, since E and F are not block diagonal, it follows from the pattern of A that each of E and F is (permutationally similar to) either a tridiagonal matrix or a direct sum of doubly nonnegative matrices of order less than or equal to 4. By [1] and [2], the matrices E and F are completely positive, and therefore A is completely positive. On the other hand, the matrix A is shown in [1] to be not completely positive. This contradiction yields that the assumption that A can be written as a sum of E and F with the above properties is false.

A related question is whether the graphs of the matrices E and F in the pattern decomposition in Theorem (3.1) are subgraphs of the graph of A. In other words, do we have

$$(3.6) \qquad e_{ij} \neq 0 \quad \text{and/or} \quad f_{ij} \neq 0 \Rightarrow a_{ij} \neq 0, \quad \text{for all } i,j \in \langle n \rangle.$$

We conclude the section with an example showing that the answer to this question too is, in general, negative.

Example 3.7. Let A be the same matrix as in Example 3.5, and let us write A in a 3×3 block form, where the diagonal blocks are of size 2, 2, and 1. Let $A = E + F$ be any positive semidefinite pattern decomposition of A. It is easy to verify that, since F is positive semidefinite, we have $f_{33}, f_{44} \geqq 0.5$, and hence $e_{33} \leqq 1.5$, $e_{44} \leqq 2.5$. Also, since E is positive semidefinite, it follows that $e_{33}, e_{44} \geqq 1$. Assume now that $e_{34} = e_{43} = 0$. A calculation shows that, under these conditions, the determinant of $E[\{1, 2, 3, 4\}]$ is less than or equal to -0.75, which contradicts the fact that E is positive semidefinite. Hence, our assumption that $e_{34} = e_{43} = 0$ is false, and we have $e_{34} \neq 0$ although $a_{34} = 0$. Therefore, (3.5) does not hold.

4. Doubly nonnegative pattern decompositions. Unlike the case of positive semidefinite pattern decompositions, not every doubly nonnegative matrix, which has the

form (1.1), has a doubly nonnegative pattern decomposition. To see this we remark the following.

Remark 4.1. In case of a doubly nonnegative pattern decomposition, since both matrices E and F are nonnegative, the implication (3.6) always holds.

Now we take the matrix A used in Example 3.7. Observe that A is doubly nonnegative. It is shown in Example 3.7 that for every positive semidefinite pattern decomposition $A = E + F$ of A, the implication (3.6) does not hold. In view of Remark 4.1, A does not have a doubly nonnegative pattern decomposition. However, we do have the following theorem.

THEOREM 4.2. *Let A be a doubly nonnegative matrix in a 3×3 block form $(A_{ij})_1^3$ with square diagonal blocks, and such that A_{22} is a 1×1 block and $A_{13} = 0$. Then every positive semidefinite pattern decomposition of A is a doubly nonnegative pattern decomposition.*

Proof. Let $A = E + F$ be a positive semidefinite pattern decomposition of A, and let $A_{22} = (a_{kk})$. Observe that for every pair $(i, j) \neq (k, k)$ we have either $e_{ij} = a_{ij}$, $f_{ij} = 0$, or $f_{ij} = a_{ij}$, $e_{ij} = 0$. Also, since E and F are positive semidefinite, it follows that $e_{kk}, f_{kk} \geqq 0$. Therefore, since A is nonnegative, it follows that E and F are nonnegative and thus doubly nonnegative. □

As an immediate corollary of Theorems 3.1 and 4.2 we obtain the following theorem.

THEOREM 4.3. *Let A be a doubly nonnegative matrix in a 3×3 block form $(A_{ij})_1^3$ with square diagonal blocks, and such that A_{22} is a 1×1 block and $A_{13} = 0$. Then A has a doubly nonnegative pattern decomposition.*

The proof of the following corollary is identical to the proof of Corollary 3.3, using Theorem 4.3 instead of Theorem 3.1.

COROLLARY 4.4. *Let A be a doubly nonnegative matrix in a $p \times p$ block tridiagonal form with square diagonal blocks, and let $k \in \langle p-1 \rangle$ be such that A_{kk} is a 1×1 block. Then A can be written as a sum of doubly nonnegative block tridiagonal matrices E and F (partitioned conformably with A), where the blocks of E and F satisfy*

$$E_{ij} = 0, \qquad \{i,j\} \not\subseteq \langle k \rangle; \qquad F_{ij} = 0, \qquad \{i,j\} \not\subseteq \{k, \cdots, n\}.$$

5. Completely positive pattern decompositions. In this section we prove that every completely positive matrix that has the block form (1.1) has a completely positive pattern decomposition. This result, together with a result from the previous section, is applied in proving a characterization of certain completely positive graphs.

THEOREM 5.1. *Let A be a completely positive matrix in a 3×3 block form $(A_{ij})_1^3$ with square diagonal blocks, and such that $A_{13} = 0$. Then A has a completely positive pattern decomposition.*

Proof. Let α, β, γ be the subsets of $\langle n \rangle$ such that $A_{11} = A[\alpha]$, $A_{22} = A[\beta]$, and $A_{33} = A[\gamma]$. Our proof is similar to the proof of Theorem 3.1. Since A is completely positive, it follows that there exists an integer m and a nonnegative $n \times m$ matrix B such that $A = BB^T$. Let $B_1 = B[\alpha | \langle m \rangle]$, $B_2 = B[\beta | \langle m \rangle]$, $B_3 = B[\gamma | \langle m \rangle]$. Since $A_{13} = B_1 B_3^T = 0$, and since B_1 and B_3 are both nonnegative, it follows that the union ω of the supports of the rows of B_1 and the union τ of the supports of the rows of B_3 are disjoint. We now write B_2 as a sum of two nonnegative matrices C_1 and C_2, where

$$C_1[\beta | \tau] = B_2[\beta | \tau], \qquad C_1[\beta | \tau) = 0,$$

$$C_2[\beta | \tau) = B_2[\beta | \tau), \qquad C_2[\beta | \tau] = 0.$$

Observe that $C_1 B_1^T = 0$, $C_1 C_2^T = 0$, and $C_2 B_3^T = 0$. Therefore, (3.2) holds and so we can continue exactly the same way as in the proof of (the real case in) Theorem 3.1. Since the matrices A_1 and A_2, as defined in that proof, are now nonnegative, it follows that the matrices E and F are completely positive. □

Let A be a completely positive matrix in the block form (1.1). We remark that not every positive semidefinite pattern decomposition of A is a completely positive pattern decomposition. Not even every doubly nonnegative pattern decomposition of A is a completely positive pattern decomposition, as demonstrated by the following example.

Example 5.2. Let A be the matrix

$$\left(\begin{array}{cc:cc:cc} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 1 & 3 & 0 & 1 & 0 \\ 1 & 0 & 0 & 4 & 1 & 0 \\ \hdashline 0 & 0 & 1 & 1 & 3 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right).$$

Observe that $A = B + C + D$, where B, C, and D are the matrices

$$\left(\begin{array}{cc:cc:cc} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 3 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \quad \left(\begin{array}{cc:cc:cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ \hdashline 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \quad \left(\begin{array}{cc:cc:cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right),$$

respectively. It is easy to verify that B, C, and D are doubly nonnegative. Furthermore, they are of order essentially less than or equal to 4. By [2], the matrices B, C, and D are completely positive, and so A is completely positive. Now write $A = E + F$, where

$$E = \left(\begin{array}{cc:cc:cc} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 3 & 1 & 0 \\ \hdashline 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \qquad F = \left(\begin{array}{cc:cc:cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right).$$

Observe that E is essentially the doubly nonnegative matrix A discussed in Example 3.5. It is easy to check that F is doubly nonnegative. Thus, $A = E + F$ is a doubly nonnegative pattern decomposition of A. However, it is not a completely positive pattern decomposition since E is not completely positive.

The proof of the following corollary is identical to the proof of Corollary 3.3, using Theorem 5.1 instead of Theorem 3.1.

COROLLARY 5.3. *Let A be a completely positive matrix in a $p \times p$ block tridiagonal form with square diagonal blocks, and let $k \in \langle p-1 \rangle$. Then A can be written as a sum of completely positive block tridiagonal matrices E and F (partitioned conformably with A), where the blocks of E and F satisfy*

$$E_{ij} = 0, \qquad \{i,j\} \not\subseteq \langle k \rangle; \qquad F_{ij} = 0, \qquad \{i,j\} \not\subseteq \{k, \cdots, n\}.$$

Remark 5.4. In a case of completely positive pattern decomposition, since both matrices E and F are nonnegative, we always have the implication (3.6).

The implication (ii) $\Rightarrow$ (i) in the following theorem is originally proven in [3]. It now becomes a nice corollary of Theorem 4.3. The implication (i) $\Rightarrow$ (ii) is a corollary of Theorem 5.1.

THEOREM 5.5. *Let G be a graph, and let G_1 and G_2 be subgraphs of G such that $G = G_1 \cup G_2$ and $|V(G_1) \cap V(G_2)| = 1$. Then the following are equivalent:*

(i) *The graph G is completely positive.*

(ii) *The graphs G_1 and G_2 are completely positive.*

Proof. (i) $\Rightarrow$ (ii). Let $V(G_1) \cap V(G_2) = \{k\}$. Let A_1 and A_2 be doubly nonnegative matrices with graphs G_1 and G_2, respectively. We have to show that A_1 and A_2 are completely positive. Let $|V(G)| = n$. We define a doubly nonnegative $n \times n$ matrix B_1 by

$$(B_1)_{ij} = \begin{cases} (A_1)_{ij}, & i,j \subseteq V(G_1), \quad (i,j) \neq (k,k) \\ x, & (i,j) = (k,k) \\ 0, & \{i,j\} \not\subseteq V(G_1) \end{cases}$$

where x is chosen to be the minimal number such that B_1 is positive semidefinite. Similarly, we define a doubly nonnegative $n \times n$ matrix B_2 by

$$(B_2)_{ij} = \begin{cases} (A_2)_{ij}, & i,j \subseteq V(G_2), \quad (i,j) \neq (k,k) \\ y, & (i,j) = (k,k) \\ 0, & \{i,j\} \not\subseteq V(G_2) \end{cases}$$

where y is chosen to be the minimal number such that B_2 is positive semidefinite. Since $(A_1)_{kk} \geqq x = (B_1)_{kk}$ and $(A_2)_{kk} \geqq y = (B_2)_{kk}$, it is enough to prove that B_1 and B_2 are completely positive. Let A be the doubly nonnegative matrix $B_1 + B_2$. We have $G(A) = G$. Since G is a completely positive graph, it follows by Theorem 5.1 that there exist completely positive matrices E and F with graphs essentially G_1 and G_2 such that $A = E + F$. We now claim that E and F are exactly B_1 and B_2, respectively. Observe that all we have to prove is that $e_{kk} = x$ and $f_{kk} = y$. This follows immediately from the minimality of x and y.

(ii) $\Rightarrow$ (i). Let A be a doubly nonnegative matrix with $G(A) = G$. To prove (i) we have to show that A is a completely positive matrix. By Theorem 4.3, A can be written as a sum of doubly nonnegative matrices E and F, where the graphs of E and F are essentially G_1 and G_2, respectively. Since G_1 and G_2 are completely positive graphs, it follows that E and F are completely positive matrices, and hence A is completely positive. □

We remark that the implication (i) $\Rightarrow$ (ii) in Theorem 5.5 follows immediately from Theorem 2 in [3], also without the condition $|V(G_1) \cap V(G_2)| = 1$. However, we preferred to provide a direct proof in this case. The implication (ii) $\Rightarrow$ (i) in Theorem 5.5 does not hold in general without the condition $|V(G_1) \cap V(G_2)| = 1$, as demonstrated by the following example.

Example 5.6. Let G_1 be the graph with vertex set $\{1, 2, 3\}$ and arc set $\{(1, 1), (2, 2), (3, 3), (1, 2), (2, 3)\}$ and let G_2 be the graph with vertex set $\{1, 3, 4, 5\}$ and arc set $\{(1, 1), (3, 3), (4, 4), (5, 5), (3, 4), (4, 5), (5, 1)\}$. Since each of these graphs has less then five vertices, it follows by [2] that both are completely positive. However, the graph $G = G_1 \cup G_2$, satisfying

$$V(G) = \{1, 2, 3, 4, 5\}$$

and

$$E(G)=\{(1,1),(2,2),(3,3),(4,4),(5,5),(1,2),(2,3),(3,4),(4,5),(5,1)\},$$

is known to be not completely positive, e.g., Example 3.12 in [1] (see Example 3.5).

Acknowledgment. We are grateful to Natalia Kogan and Abraham Berman for their helpful comments.

REFERENCES

[1] A. BERMAN AND D. HERSHKOWITZ, *Combinatorial results on completely positive matrices*, Linear Algebra Appl., 95 (1987), pp. 111–125.

[2] L. J. GRAY AND D. G. WILSON, *Nonnegative factorization of positive semidefinite nonnegative matrices*, Linear Algebra Appl., 31 (1980), pp. 119–127.

[3] N. KOGAN AND A. BERMAN, *Characterization of completely positive graphs*, to appear.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No 4, pp. 620–631, October 1990

012

THE YOUNG–EIDSON ALGORITHM: APPLICATIONS AND EXTENSIONS*

A. HADJIDIMOS† AND D. NOUTSOS†

Abstract. In this paper it is assumed that the point (or block) Jacobi matrix B associated with the matrix A is weakly 2-cyclic consistently ordered with complex, in general, eigenvalue spectrum $\sigma(B)$ lying in the interior of the infinite unit strip. It is then our objective to apply and extend the Young–Eidson algorithm in order to determine the real optimum relaxation factor in the following two cases: (i) In the case of the successive overrelaxation (SOR) matrix associated with A when $\sigma(B)$ lies in a "bow-tie" region, and (ii) in the case of the symmetric SOR (SSOR) matrix associated with A. In the latter case a number of numerical examples are given. It is noted that as a by-product of (ii) both the relaxation factor for the SSOR matrix corresponding to a "bow-tie" spectrum $\sigma(B)$ and the optimum pairs of the relaxation factors for the unsymmetric SOR (USSOR) matrix associated with A are also obtained.

Key words. successive overrelaxation (SOR), symmetric successive overrelaxation (SSOR), unsymmetric successive overrelaxation (USSOR), optimum relaxation factor(s)

AMS(MOS) subject classification. 65F10

C.R. classification. 5.14

1. Introduction and preliminaries.

In 1970 an algorithm for the determination of the real optimum relaxation factor for the successive overrelaxation (SOR) matrix associated with a weakly 2-cyclic consistently ordered Jacobi matrix B (see, e.g., [12], [15], [4], or [9]) whose eigenvalue spectrum $\sigma(B)$ was complex, was developed and proposed by Young and Eidson [17] (see also [15]). To the best of our knowledge, so far, the powerfulness and the simplicity of the Young–Eidson algorithm has been explored by few researchers (see, e.g., [2], [3], [1], etc.). So, problems, which could have been solved by the aforementioned algorithm in a much simpler, clearer and more efficient way, have been attacked with more complicated methods while others have simply remained unsolved. Here we mention (i) the problem of the optimum SOR parameter when $\sigma(B)$ lies in a "bow-tie" region obtained by Chin and Manteuffel [5] (see also [7]) and (ii) the "unsolved" problem of the optimum relaxation factor for the symmetric SOR (SSOR) method.

It is the purpose of this paper to strictly follow the reasoning behind the Young–Eidson algorithm and "extend" it in order to give the solutions to both aforementioned problems (i) and (ii). These problems are solved under the assumption made in the beginning, that is, the Jacobi matrix B is weakly 2-cyclic consistently ordered and that $\sigma(B) \subset S$, where

$$S := \{ z \in \mathbb{C} : |Re z| < 1 \}. \tag{1.1}$$

They are presented in §§ 3 and 4, respectively. As a by-product of our analysis the optimum relaxation factor for the SSOR matrix for a "bow-tie" $\sigma(B)$ and the optimum pairs for the relaxation factors of the unsymmetric SOR (USSOR) method are also obtained in

* Received by the editors December 11, 1988; accepted for publication (in revised form) September 18, 1989. This research was supported in part by National Science Foundation grant CCR-8619817.

† Department of Mathematics, University of Ioannina, GR 451 10 Ioannina, Greece and Visiting Professor and Scholar at Department of Computer Sciences, Purdue University, West Lafayette, Indiana 47907 (hadjidim@cs.purdue.edu).

§ 5. Meanwhile in § 2 we give a synopsis of the background material on which the Young–Eidson algorithm is based, so that the interested reader can follow its extensions in the later sections with not much difficulty.

2. Presentation of background material. Assume that

$$A := D - L - U \tag{2.1}$$

is a 2-cyclic consistently ordered matrix with nonsingular corresponding diagonal (or block diagonal) part D and strictly lower and upper triangular parts L and U. Denote by

$$B := D^{-1}(L+U) \tag{2.2}$$

and

$$\mathscr{L}_\omega := (D-\omega L)^{-1}[(1-\omega)D+\omega U], \tag{2.3}$$

where $\omega \in (0, 2)$ is the relaxation factor, the Jacobi and the SOR matrices associated with A. Let H be the smallest convex polygon symmetric about the axes such that $\sigma(B) \subset H$ and let $P_j(\alpha_j, \beta_j), j = 1(1)s$, be its vertices in the first quadrant, in increasing order of magnitude of their abscissas. Obviously our basic assumption $\sigma(B) \subset S$ implies $H \subset S$.

Now let E_p denote an ellipse passing through the point P, in the first quadrant of S, symmetric about the axes and contained in S. Also let int E_p and $\overline{\text{int}}\ E_p$ denote the interior and the closure of the interior of E_p. Then an analysis based on the Young's famous relationship [14]

$$(\lambda+\omega-1)^2 = \omega^2\mu^2\lambda, \tag{2.4}$$

which connects the two sets of eigenvalues $\mu \in \sigma(B)$ and $\lambda \in \sigma(\mathscr{L}_\omega)$, shows the following (see [15, pp. 191–200]). If a and b are the "real" and the "imaginary" semiaxes of an E_p passing through the vertex P of H and are such that $\sigma(B)$ (and H) $\subset \overline{\text{int}}\ E_p$, then the parameter ω and the spectral radius of $\mathscr{L}_\omega$, $\rho(\mathscr{L}_\omega)$, are given by the expressions

$$\begin{aligned} &\omega = 2/(1+(1-a^2+b^2)^{1/2}), \qquad \rho(\mathscr{L}_\omega) = \rho^2, \\ &\rho = (a+b)/(1+(1-a^2+b^2)^{1/2}). \end{aligned} \tag{2.5}$$

Out of the infinitely many ellipses E_p that satisfy $H \subset \overline{\text{int}}\ E_{p_j}, j = 1(1)s$, there exists a unique "optimum" one $\hat{E}$ for which ρ is a minimum. For $s > 2$ the optimum ellipse is determined by means of the Young–Eidson algorithm. The latter is, in turn, based on the optimum results for $s = 1$ and 2. For $s = 1$, let $P_1(\alpha_1, \beta_1) \equiv P(\alpha, \beta)$. Then we can find out that ρ, in (2.5), as a function of $a \in [\alpha_1, 1]$, strictly decreases in $[\alpha_1, \hat{a}]$ from 1 to $\hat{\rho}$ and strictly increases in $[\hat{a}, 1]$ from $\hat{\rho}$ to 1. The optimum value for ρ, $\hat{\rho}$, is the unique root of

$$((1+\rho^2)/(2\rho))^{2/3}\alpha^{2/3} + ((1-\rho^2)/(2\rho))^{2/3}\beta^{2/3} - 1 = 0 \tag{2.6}$$

in (0, 1), where it is noted that (2.6) is equivalent to a cubic equation (see, e.g., [3]), while the optimum values for a and b, $\hat{a}$ and $\hat{b}$, are given by

$$\hat{a} = (2\hat{\rho}\alpha^2/(1+\hat{\rho}^2))^{1/3}, \qquad \hat{b} = (2\hat{\rho}\beta^2/(1-\hat{\rho}^2))^{1/3}. \tag{2.7}$$

Finally, the optimum values $\hat{\omega}$ and $\rho(\mathscr{L}_{\hat{\omega}})$ are obtained through (2.5) by using (2.7). In the very special cases $\beta_1 = 0$ and $\alpha_1 = 0$ the well-known results

$$\hat{\omega} = 2/(1+(1-\rho^2(B))^{1/2}), \qquad \rho(\mathscr{L}_{\hat{\omega}}) = \hat{\omega} - 1 \tag{2.8a}$$

due to Young [14] and

$$\hat{\omega} = 2/(1+(1+\rho^2(B))^{1/2}), \qquad \rho(\mathscr{L}_{\hat{\omega}}) = 1-\hat{\omega}, \tag{2.8b}$$

a special case of Kredell's result [10], are easily recovered.

For $s = 2$ let $E_{p_1p_2}$ be the ellipse symmetric about the axes that passes through both vertices P_1 and P_2. Its semiaxes $a_{1,2}$ and $b_{1,2}$ are given by

$$a_{1,2} = ((\alpha_2^2\beta_1^2 - \alpha_1^2\beta_2^2)/(\beta_1^2 - \beta_2^2))^{1/2}, \qquad b_{1,2} = ((\alpha_2^2\beta_1^2 - \alpha_1^2\beta_2^2)/(\alpha_2^2 - \alpha_1^2))^{1/2}. \tag{2.9}$$

The optimum ellipse $\hat{E}$ for H (and $\sigma(B)$) is obtained after an analysis based on the previous arguments takes place (see [15]). If $\hat{E}_{p_j}$ is the optimum ellipse corresponding to P_j and $\hat{a}_j$, $\hat{b}_j$ its semiaxes ($j = 1, 2$), then $\hat{E}$ can be determined by the following simple algorithm given in pseudocode:

ALGORITHM 1.
Determine $E_{p_1p_2}(a_{1,2})$;
Determine $\hat{E}_{p_2}(\hat{a}_2)$;
if $\hat{a}_2 \leqq a_{1,2}$ **then** $\hat{E} \equiv \hat{E}_{p_2}$; **stop;**
else Determine $\hat{E}_{p_1}(\hat{a}_1)$;
  if $a_{1,2} \leqq \hat{a}_1$ **then** $\hat{E} \equiv \hat{E}_{p_1}$; **stop;**
  else $\hat{E} \equiv E_{p_1p_2}$; **stop;**
  endif;
endif;
end of Algorithm 1;

The Young–Eidson algorithm is an ingenious systematic extension of Algorithm 1 to $s \geqq 3$ (see [17] and [15]). It is taken into consideration that two distinct ellipses symmetric about the axes cannot have more than one common point in the first quadrant. Thus by virtue of the analysis presented so far the optimum ellipse $\hat{E}$ is the one out of the $\hat{E}_{p_j}$'s, $j = s(-1)1$, for which $H \subset \overline{\text{int}}\ \hat{E}_{p_j}$, provided such an ellipse exists. In case it does not exist, it is the ellipse $E_{p_jp_k}$ out of $E_{p_jp_k}$'s, $j = s(-1)2$, $k = j-1(-1)1$, satisfying $H \subset \overline{\text{int}}\ E_{p_jp_k}$, which corresponds to the smallest ρ. The existence and uniqueness of $\hat{E}$ readily follow. For an H with any finite number of vertices the Young–Eidson algorithm is given below. ($s(\geqq 2)$ denotes the number of vertices of H in the first quadrant and ν is used to denote the real semiaxis of an ellipse passing through two points in the first quadrant.)

ALGORITHM 2.
$\nu_{\text{old}} := \alpha_s$; $\rho_{\text{old}} := 1$;
again: $\nu_{\text{new}} := 1$;
  for $j := s-1(-1)1$ **do**
    Determine $E_{p_sp_j}(a_{s,j})$;
    if $\nu_{\text{new}} > a_{s,j}$ **then**
      $k := j$; $\nu_{\text{new}} := a_{s,k}$;
    endif;
  end do;
  Determine $\hat{E}_{p_s}(\hat{a}_s)$;
  if $\hat{a}_s < \nu_{\text{old}}$ **or** $\hat{a}_s > \nu_{\text{new}}$ **then**
    Determine $\rho_{s,k}$ ($\equiv \rho$ corresponding to $E_{p_sp_k}$);
    if $\rho_{s,k} < \rho_{\text{old}}$ **then**
      $\rho_{\text{old}} := \rho_{s,k}$; $r := s$; $q := k$;
    endif;

if $k = 1$ **then**
Determine $\hat{E}_1(\hat{a}_1)$;
if $\hat{a}_1 \geqq \nu_{\text{new}}$ **then**
$\hat{E} \equiv \hat{E}_{p_1}$; $\hat{\rho} := \hat{\rho}_1$; **stop;**
else
$\hat{E} \equiv E_{p_r p_q}$; $\hat{\rho} := \rho_{\text{old}}$; **stop;**
endif;
else
$s := k$; $\nu_{\text{old}} := \nu_{\text{new}}$; **goto** again;
endif;
else
$\hat{E} \equiv \hat{E}_{p_s}$; $\hat{\rho} := \hat{\rho}_s$; **stop;**
endif;
end of Algorithm 2;

3. Optimum relaxation factor for a bow-tie region. In a recent paper Chin and Manteuffel [5] determined, after a rather complicated analysis, the optimum SOR factor when $\sigma(B)$ lies in a bow-tie region $R \subset S$ (see Figs. 1, 2, and 3). A solution to the same problem was provided by Eiermann, Li, and Varga [7] by applying asymptotically optimal hybrid semi-iterative methods. Here we present a solution based on a strict reasoning of

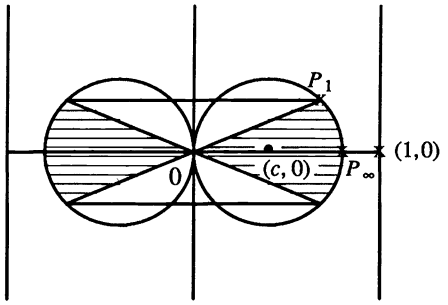


FIG. 1

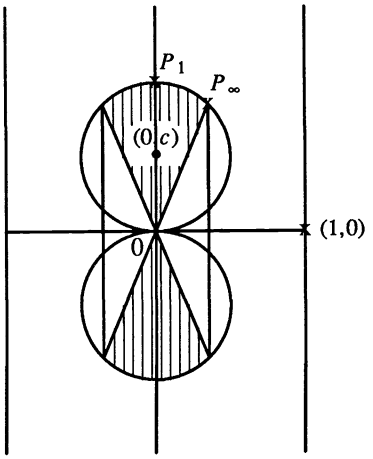


FIG. 2

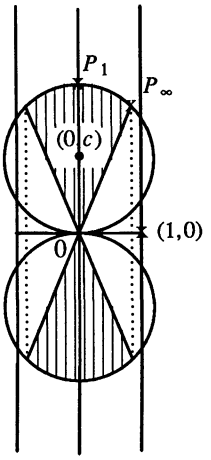


FIG. 3

the Young–Eidson algorithm. As we can find out it is less complicated and easier to understand.

For the algorithm to be applied we must consider the union of all possible convex polygons H, defined in the previous section, whose vertices V in the first quadrant are any arbitrary points of the arc $P_\infty P_1$, with abscissas α strictly decreasing from α_∞ to α_1. The coordinates (α, β) of each vertex V satisfy the equation

$$(x-c)^2+y^2=c^2 \qquad \text{(Fig. 1)} \tag{3.1}$$

or

$$x^2+(y-c)^2=c^2. \qquad \text{(Figs. 2, 3)} \tag{3.1'}$$

First it is noted that the optimum ellipse $\hat{E}$ for the aforementioned infinite set of vertices V cannot be an ellipse passing through two vertices V_2 and V_1 since then the arc $V_2V_1 \not\subset \overline{\text{int}}\, E$. Consequently, $\hat{E}$ must be an optimum ellipse passing through one vertex only. Apparently this leaves us with three possibilities: $\hat{E}$ is either of $\hat{E}_{p_\infty}$, $\hat{E}_{p_1}$, passing through the end point-vertices P_∞ or P_1, or the osculating ellipse that is tangent to the arc $P_\infty P_1$, at a point V of it, and is, at the same time, the $\hat{E}_\nu$.

In the case of Fig. 1, $\hat{E}_{p_\infty}$ is excluded since it is the line segment $[-2c, 2c]$, $c < \frac{1}{2}$, and $H \not\subset \hat{E}_{p_\infty}$. To examine the possibility of the optimum osculating ellipse, if such an ellipse exists, let $\hat{a}$, $\hat{b}$ be its semiaxes and $(\hat{\alpha}, \hat{\beta})$ the coordinates of the point of contact V. Substituting $(\hat{\alpha}, \hat{\beta})$ for (α, β) in (2.6)–(2.7) and recalling that $\hat{E}_\nu$ and (3.1) have a common tangent at $(\hat{\alpha}, \hat{\beta})$ defined by

$$\frac{\hat{\alpha}x}{\hat{a}^2}+\frac{\hat{\beta}y}{\hat{b}^2}=1 \quad \text{and} \quad (\hat{\alpha}-c)(x-c)+\hat{\beta}y=c^2 \tag{3.2}$$

it is obtained that

$$\hat{a}^2=\hat{\alpha}^2c/(\hat{\alpha}-c), \qquad \hat{b}^2=\hat{\alpha}c. \tag{3.3}$$

Hence by using expressions (3.3) and the fact that $(\hat{\alpha}, \hat{\beta})$ lies on (3.1), and then equating the two roots of the two equations (2.7) in the interval (0, 1), namely,

$$\hat{\rho}_1=\left(\frac{(\hat{\alpha}-c)^3}{\hat{\alpha}^2c^3}\right)^{1/2}-\left(\frac{(\hat{\alpha}-c)^3}{\hat{\alpha}^2c^3}-1\right)^{1/2} \tag{3.4a}$$

and

$$\hat{\rho}_2=-\left(\frac{\hat{\beta}^4}{\hat{\alpha}^3c^3}\right)^{1/2}+\left(\frac{\hat{\beta}^4}{\hat{\alpha}^3c^3}+1\right)^{1/2}, \tag{3.4b}$$

we obtain

$$\hat{\alpha}=\frac{c(1+(5-4c^2)^{1/2})}{2(1-c^2)}\left(=\frac{2c}{(5-4c^2)^{1/2}-1}\right). \tag{3.5}$$

Therefore if $\alpha_1 \leqq \hat{\alpha}$, the optimum ellipse for H is $\hat{E}_\nu$. Using (3.5) in (3.3) and also in one of (3.4) and then the resulting expressions in (2.5), we have that the values of $\hat{\rho}$, $\hat{\omega}$, and $\rho(\mathscr{L}_{\hat{\omega}})$, in terms of c, are obtained. It is checked that either of (3.4) coincides with (3.34) in [5] and that (2.6) yields (3.35b) in [5], where in the latter the sign in the constant term should be minus instead of plus. If, on the other hand, $\hat{\alpha} < \alpha_1$, the optimum ellipse for H is $\hat{E}_{p_1}$ and all the optimum parameters associated with it are obtained from (2.6), (2.7), and (2.5), with $(\alpha, \beta) = (\alpha_1, \beta_1)$.

In the case of Figs. 2 and 3, $\hat{E}_{p_1}$ is excluded since it is the line segment $[-2ic, 2ic]$ and $H \not\subset \hat{E}_{p_1}$. The possibility of an osculating ellipse must be examined only in the case of Fig. 2 and this is what is done very briefly in the sequel. The analysis almost duplicates the one made previously, where, instead of (3.1), (3.1′) is used. Thus we can obtain

$$\frac{\hat{\alpha}x}{\hat{a}^2}+\frac{\hat{\beta}y}{\hat{b}^2}=1 \quad \text{and} \quad \hat{\alpha}x+(\hat{\beta}-c)(y-c)=c^2, \tag{3.2′}$$

$$\hat{a}^2=\hat{\beta}c, \qquad \hat{b}^2=\beta^2 c/(\beta-c), \tag{3.3′}$$

$$\hat{\rho}_1=\left(\frac{\alpha^4}{\beta^3c^3}\right)^{1/2}-\left(\frac{\alpha^4}{\beta^3c^3}-1\right)^{1/2}, \tag{3.4a′}$$

$$\hat{\rho}_2=-\left(\frac{(\beta-c)^3}{\beta^2c^3}\right)^{1/2}+\left(\frac{(\beta-c)^3}{\beta^2c^3}+1\right)^{1/2}, \tag{3.4b′}$$

and finally

$$\hat{\beta}=\frac{c(1+(5+4c^2)^{1/2})}{2(1+c^2)}\left(=\frac{2c}{(5+4c^2)^{1/2}-1}\right). \tag{3.5′}$$

Consequently, if $\beta_\infty \leqq \hat{\beta}$, the optimum ellipse for H is $\hat{E}_\nu$. So the values for $\hat{\rho}$, $\hat{\omega}$, and $\rho(\mathscr{L}_{\hat{\omega}})$ are derived in exactly the same way as before, where, however, the corresponding primed expressions are used. It is again checked that either of (3.4′) coincides with (3.49) in [5] and that (2.6) yields (3.50b) in [5]. It should be mentioned that the numerator in the last fraction under the last square root in the denominator should read 4 instead of 1. If $\hat{\beta} < \beta_\infty$ or if we are in the case of Fig. 3, $\hat{E}_{p_\infty}$ is the optimum ellipse for H. The associated optimum parameters are obtained from (2.6), (2.7), and (2.5) with $(\alpha, \beta) = (\alpha_\infty, \beta_\infty)$.

4. Optimum relaxation factor for the SSOR matrix.

4.1. Development of the basic theory. As is stated [15],

$$S_\omega := (D-\omega U)^{-1}((1-\omega)D+\omega L)(D-\omega L)^{-1}((1-\omega)D+\omega U) \tag{4.1}$$

is the SSOR iteration matrix associated with A in (2.1), where $\omega \in (0, 2)$ is the relaxation factor. For A 2-cyclic consistently ordered the sets of eigenvalues $\mu \in \sigma(B)$ and $\lambda \in \sigma(S_\omega)$ are connected through the relationship

$$(\lambda-(1-\omega)^2)^2=\omega^2(2-\omega)^2\mu^2\lambda \tag{4.2}$$

due to D'Sylva and Miles [6] and Lynn [11] (see also [13]). However, as was proved in [8], when we make the substitution $\omega' = \omega(2 - \omega) \in (0, 1]$, there exist values of ω', at least in the neighborhood of zero, for which $\rho(S_\omega) < 1$ if and only if $\sigma(B^2)$ lies in the interior of the parabola $P := y^2 = -4x + 4$, the latter requirement being equivalent to $\sigma(B) \subset S$ (the infinite unit strip). On the other hand, the aforementioned substitution transforms (4.2) into

$$(\lambda+\omega-1)^2=\omega^2\mu^2\lambda, \tag{4.3}$$

where primes have been dropped to simplify the notation. This is nothing but (2.4), and consequently the problem of the determination of the optimum ω is exactly the same as the one solved in § 2 with the only exception being that the new ω is now restricted to values in (0, 1]. This, in turn, implies that for the convex polygon H, defined there, with one vertex $P(\alpha, \beta)$ in the first quadrant out of all the ellipses E_p, such that $H \subset \overline{\text{int}}\ E_p$,

only those with $a \leqq b$, equivalent to $\omega \in (0, 1]$, must be considered. Consequently, having in mind the analysis in § 2 and especially how ρ varies with the semiaxis a varying in $[\alpha, 1]$, we can state the main result in the form of a theorem.

THEOREM (Determination of the optimum SSOR factor). *Given the 2-cyclic consistently ordered matrix A of (2.1) with B of (2.2) being weakly cyclic of index 2. Let H be the smallest convex polygon symmetric about the axes such that $\sigma(B) \subset H$. Assume further that H has one vertex $P(\alpha, \beta)$ in the first quadrant of S defined in (1.1). Then for the determination of the optimum* SSOR *factor follow the steps*:

(a) *Determine the optimum ellipse $\hat{E}_p$, as in § 2, by means of (2.6) and (2.7).*

(b) (i) *If $\hat{a} \leqq \hat{b}$, then find $\hat{\omega}$ ($\leqq 1$) through (2.5). The two zeros $\hat{\omega}_1$, $\hat{\omega}_2$ of $\omega(2 - \omega) = \hat{\omega}$ are the optimum values for the original ω in the* SSOR *matrix (4.1).*

(ii) *If $\hat{a} > \hat{b}$, then $\hat{\omega} = 1$ and the circle $\hat{C}_p$ centered at the origin and passing through P gives the optimum "ellipse" for the* SSOR *problem. In this case $\hat{\omega}_1 = \hat{\omega}_2 = 1$.* □

The above theorem can be directly applied to the cases of (i) $\sigma(B)$ real with $\rho(B) < 1$ and (ii) $\sigma(B)$ purely imaginary to yield well-known results (see, e.g., [8]). So, we have the following corollary.

COROLLARY. *Under the assumptions of the theorem*

(i) *If $\sigma(B)$ is real with $\rho(B) < 1$, then $\hat{a} = \rho(B) > 0 = \hat{b}$* implying that

$$\hat{\omega} = 1, \qquad \rho(S_{\hat{\omega}}) = \rho^2(B). \tag{4.4}$$

(ii) *If $\sigma(B)$ is purely imaginary, then $\hat{a} = 0 < \rho(B) = \hat{b}$ giving that*

$$\hat{\omega}_{1,2} = 1 \pm \frac{\rho(B)}{1 + (1 + \rho^2(B))^{1/2}}, \qquad \rho(S_{\hat{\omega}_1}) = \rho(S_{\hat{\omega}_2}) = \frac{1 - (1 + \rho^2(B))^{1/2}}{1 + (1 + \rho^2(B))^{1/2}}. \tag{4.5}$$

□

Observing the way the Young–Eidson algorithm was developed to determine the optimum ellipse E, based on the analysis of the special cases when the convex polygon H had one or two vertices in the first quadrant, we can see in a quite analogous way that an extension of the Young–Eidson algorithm can be developed to cope with the SSOR case. In the sequel we give first the algorithm in the case where H has two vertices and then the algorithm in the general case, where H has any finite number of s ($\geqq 2$) vertices in the first quadrant. The basic assumptions and the various notations are the ones used so far except that the pair (ν, ξ) is used to denote the semiaxes of an ellipse passing through two points in the first quadrant.

4.2. The two-point algorithm.

ALGORITHM 3.
Determine $E_{p_1p_2}(\nu, \xi)$;
Determine $\hat{E}_{p_2}(\hat{a}_2, \hat{b}_2)$;
if $\nu > \xi$ **then**
    if $\hat{a}_2 > \hat{b}_2$ **then** $\hat{E} \equiv \hat{C}_{p_2}$; **stop;**
    else $\hat{E} \equiv \hat{E}_{p_2}$; **stop;**
    endif;
else
    if $\hat{a}_2 \leqq \nu$ **then** $\hat{E} \equiv \hat{E}_{p_2}$; **stop;**
    else Determine $\hat{E}_{p_1}(\hat{a}_1, \hat{b}_1)$;
        if $\hat{a}_1 \geqq \nu$ **then**
            if $\hat{a}_1 > \hat{b}_1$ **then** $\hat{E} \equiv \hat{C}_{p_1}$; **stop;**
            else $\hat{E} \equiv \hat{E}_{p_1}$; **stop;**

```
            endif;
        else Ê ≡ E_{p1p2}; stop;
        endif;
    endif;
endif;
end of Algorithm 3;
```

4.3. The s-point algorithm ($s \geqq 2$).

ALGORITHM 4.

```
ν_old := α_s; ρ_old := 1;
again: ν_new := 1; ξ_new := 0;
    for j := s − 1(−1)1 do
        Determine E_{p_s p_j}(ν_j, ξ_j);
        if ν_new > ν_j then
            k := j; ν_new := ν_k; ξ_new := ξ_k;
        endif;
    end do;
    Determine Ê_{p_s}(â_s, b̂_s);
    if ν_new > ξ_new then
        if â_s > b̂_s then
            Determine ρ̂_c( ≡ radius of the circle Ĉ_{p_s});
            if ρ̂_c < ρ_old then
                Ê ≡ Ĉ_{p_s}; ρ̂ := ρ̂_c; stop;
            else
                Ê ≡ E_{p_r p_q}; ρ̂ := ρ_old; stop;
            endif;
        else
            if â_s ≧ ν_old then
                Ê ≡ Ê_{p_s}; ρ̂ := ρ̂_s; stop;
            else
                Ê ≡ E_{p_r p_q}; ρ̂ := ρ_old; stop;
            endif;
        endif;
    else
        if â_s < ν_old or â_s > ν_new then
            Determine ρ_{s,k} ( ≡ ρ corresponding to E_{p_s p_k});
            if ρ_{s,k} < ρ_old then
                ρ_old := ρ_{s,k}; r := s; q := k;
            endif;
            if k = 1 then
                Determine Ê_{p_1}(â_1, b̂_1);
                if â_1 ≧ ν_new then
                    if â_1 > b̂_1 then
                        Determine ρ̂_c ( ≡ radius of the circle Ĉ_{p_1});
                        if ρ̂_c < ρ_old then
                            Ê ≡ Ĉ_{p_1}; ρ̂ := ρ̂_c; stop;
                        else
                            Ê ≡ Ê_{p_r p_q}; ρ̂ := ρ_old; stop;
                        endif;
                    else
```

$\hat{E} \equiv \hat{E}_{p_1}$; $\hat{\rho} := \hat{\rho}_1$; **stop;**
endif;
else
$\hat{E} \equiv E_{p_r p_q}$; $\hat{\rho} := \rho_{\text{old}}$; **stop;**
endif;
else
$s := k$; $\nu_{\text{old}} := \nu_{\text{new}}$; **goto** again;
endif;
else
$\hat{E} \equiv \hat{E}_{p_s}$; $\hat{\rho} := \hat{\rho}_s$; **stop;**
endif;
endif;
end of Algorithm 4;

4.4. Differences between Algorithms 3, 4 and Algorithms 1, 2. For the interested reader, who is already familiar with Algorithms 1 and 2 (Young–Eidson), the following two points, in connection with the new Algorithms 3 and 4, must be made: (i) Whenever, in Algorithm 3 (and/or Algorithm 4), a one-point optimum ellipse is found, which is also the optimum ellipse for H, such that $\hat{a} \leqq \hat{b}$, then this ellipse gives the solution to the SSOR problem. If, however, $\hat{a} > \hat{b}$ then the solution to the SSOR problem is given by means of the circle, symmetric about the axes, passing through the point in question. This circle is considered to be the optimum ellipse for H. (ii) Whenever a two-point ellipse is considered and it so happens that for this ellipse $a \leqq b$, then the ellipse in question is treated in exactly the same way as in Algorithms 1 and 2. However, if $a > b$, this ellipse is ignored and the algorithm proceeds on to the next step.

4.5. Numerical examples. In this section a number of numerical examples, covering the two- and the s-point ($s > 2$) cases are presented. These examples have been selected from those worked out in Examples 8–12 of [17]. In each one of them the set of vertices $P_j(\alpha_j, \beta_j), j = 1(1)s$, of H in the first quadrant, in increasing order of magnitude of their abscissas, as was described in §§ 2 and 4.1, is given. For the determination of the optimum SSOR parameter both Algorithms 3 and 4 were used when $s = 2$ and only Algorithm 4 when $s > 2$. The optimal values $\hat{a}$, $\hat{b}$, $\hat{\rho}$ obtained from the corresponding algorithm were subsequently used to determine $\hat{\omega}$ through (2.5) and then $\hat{\omega}_1$ ($\leqq 1 \leqq \hat{\omega}_2$) as the smallest zero of the equation $\omega(2 - \omega) = \hat{\omega}$. The latter value is considered to be the theoretical one for $\hat{\omega}$ of the SSOR method. Next, and for each example, the $s \times s$ Frobenius matrix $\tilde{B}$ (see [12, Ex. 4, p. 48]) with eigenvalues the s numbers $\mu_j = \alpha_j + i\beta_j, j = 1(1)s$, and the weakly 2-cyclic matrix $B = \left[\begin{smallmatrix} 0 & \tilde{B} \\ \tilde{B} & 0 \end{smallmatrix}\right]$ were constructed. Together with B ($:= L + U$) the associated SSOR matrix

$$S_\omega := (I - \omega U)^{-1}((1-\omega)I + \omega L)(I - \omega L)^{-1}((1-\omega)I + \omega U)$$

was considered for all $\omega = 0.001\ (0.001)\ 1.999$. Using a Sequent Symmetry Computer and a FORTRAN program (single precision) with calls from LINPACK and EISPACK the values for $\omega = \hat{\omega}$ for which $\rho(S_\omega)$ is minimized were found within an accuracy of three decimal places. As we can see from the self-explanatory Table 1, the theoretical values for $\hat{\omega}_1$ (and $\hat{\omega}_2$) obtained by using our algorithms and the experimental ones obtained by minimizing $\rho(S_\omega)$ are almost identical, a fact that supports the theory developed in this paper. We simply note that some slight discrepancies between the theoretical values for $\hat{a}$, $\hat{b}$, $\rho(S_{\hat{\omega}_1})$ ($=\rho(S_{\hat{\omega}_2})$) and the ones given in [17] are mainly due to the accuracy used but might be due to a "bug" that was present in one of the programs in the original version of [17] ([16]).

TABLE 1

Example	Set of eigenvalues μ_j	Semiaxes of the optimum ellipse	Theoretical values $\hat{\omega}_1$	Theoretical values $\rho(S_{\hat{\omega}_1})$ $[=\rho(S_{\hat{\omega}_2})]$	Experimental values $\hat{\omega}_1$ $(\hat{\omega}_2)$	Experimental values $\rho(S_{\hat{\omega}_1})$ $[=\rho(S_{\hat{\omega}_2})]$
1	$\mu_1 = .3 + .55i$ $\mu_2 = .75 + .25i$	$\hat{a} = .8171625$ $\hat{b} = .6296909$	1.	.6250000	1.000 (1.000)	.6250000
2	$\mu_1 = .7 + .55i$ $\mu_2 = .75 + .25i$	$\hat{a} = .7847299$ $\hat{b} = 1.216860$	.6068593	.7159068	.607 (1.393)	.7159067 (.7159057)
3	$\mu_1 = .5 + .55i$ $\mu_2 = .75 + .25i$	$\hat{a} = .8171625$ $\hat{b} = .6296909$	1.	.6250000	1.000 (1.000)	.6250000
4	$\mu_1 = .25 + .875i$ $\mu_2 = .5 + .6i$ $\mu_3 = .7 + .3i$	$\hat{a} = .7395588$ $\hat{b} = .9297315$	.7376715	.6040523	.737 (1.263)	.6042088 (.6042085)
5	$\mu_1 = .3 + 3i$ $\mu_2 = .5 + 2i$ $\mu_3 = .78654 + 1.75432i$ $\mu_4 = .8 + .1i$	$\hat{a} = .8519320$ $\hat{b} = 4.565933$	.1983810	.9373946	.198 (1.803)	.9373845 (.9360802)

5. Applications. The analysis and the optimum algorithms presented in the previous section will be applied: (i) to the SSOR matrix corresponding to the "bow-tie" spectrum B of § 2, and (ii) to the unsymmetric (US) SOR matrix associated with a special type block 2-cyclic consistently ordered matrix A in (2.1).

(i) The observation made in § 2, that is the optimum ellipse $\hat{E}$ for the SOR matrix cannot be an ellipse passing through two vertices of the "convex polygon," still holds for the SSOR matrix. In the case of Fig. 1 we note that $\hat{E}$ has $\hat{a}_\infty = \rho(B) = 2c > 0 = \hat{b}_\infty$. This simply implies that $\hat{\omega} = 1$ and the problem is solved. In the case of Figs. 2 and 3 we have for $\hat{E}$ either $\hat{E}_v\,(\hat{\alpha} < c < \hat{\beta})$ or $\hat{E}_{p_\infty}\,(\hat{\alpha}_\infty < c < \hat{\beta}_\infty)$. In view of (2.7) it is implied that either $\hat{a} < \hat{b}$ or $\hat{a} = \hat{a}_\infty < \hat{b}_\infty = \hat{b}$, respectively. Therefore $\hat{\omega}$ obtained from (2.5) with $(a, b) = (\hat{a}, \hat{b})$ provides us with the two values $\hat{\omega}_1$, $\hat{\omega}_2$ of the optimum SSOR factor through the formulas

$$\hat{\omega}_{1,2} = 1 \pm \frac{(\hat{b}^2 - \hat{a}^2)^{1/2}}{1 + (1 - \hat{a}^2 + \hat{b}^2)^{1/2}}. \tag{5.1}$$

(ii) Let A in (2.1) be 2-cyclic consistently ordered matrix of the following block form:

$$A = \begin{bmatrix} D_1 & O \\ O & D_2 \end{bmatrix} - L - U, \tag{5.2}$$

where D_1, D_2 are square nonsingular matrices. If, in (2.1), $D = \operatorname{diag}(D_1, D_2)$, then the USSOR matrix associated with A in (5.2) is defined by

$$C_{\omega_1,\omega_2} := (D - \omega_2 U)^{-1}((1 - \omega_2)D + \omega_2 L)(D - \omega_1 L)^{-1}((1 - \omega_1)D + \omega_1 U) \tag{5.3}$$

and as is proved in [15, pp. 476–478] the eigenvalues of C_{ω_1,ω_2} are the same as those of $\mathcal{L}_\omega$ with

$$\omega := \omega_1 + \omega_2 - \omega_1\omega_2. \tag{5.4}$$

On the other hand, necessary conditions for $\rho(C_{\omega_1,\omega_2}) < 1$ are

$$0 < \omega_1 + \omega_2 - \omega_1\omega_2 < 2. \tag{5.5}$$

So, if there is no further restriction on ω_1, ω_2, then the algorithm of Young and Eidson will provide us with an optimum $\omega = \hat{\omega} \in (0, 2)$ and (5.4) will give us optimum pairs $(\omega_1, \omega_2) = (\hat{\omega}_1, \hat{\omega}_2)$ lying on the hyperbola

$$\hat{\omega}_1 + \hat{\omega}_2 - \hat{\omega}_1\hat{\omega}_2 = \hat{\omega}. \tag{5.6}$$

If, however, we impose a further restriction on ω_1, ω_2, as for example in the case of the SSOR method where $\omega_1 = \omega_2$, this may restrict the interval for ω from $(0, 2)$ to, say, $I \subset (0, 2)$. In such a case new restrictions will be imposed on the semiaxes (a, b) of the ellipse $\hat{E}_p$ passing through the point $P(\alpha, \beta)$, which considered together with the behavior of ρ as a function of a will lead to slight modifications of the basic algorithms of the SSOR case. For example, suppose that ω_1 and ω_2 satisfy

$$\omega_1 = \omega_2 + 1; \tag{5.7}$$

then (5.4) and (5.5) are equivalent to

$$0 < \omega := -\omega_2^2 + \omega_2 + 1 < 2, \tag{5.8}$$

which give $\omega_2 \in ((1 - \sqrt{5})/2, (1 + \sqrt{5})/2)$. This implies that $\omega \in I := (0, \frac{5}{4}] \subset (0, 2)$, and from (2.5) we have

$$a \leqq \left(\frac{16}{25} + b^2\right)^{1/2}. \tag{5.9}$$

In view of the restriction (5.9) the only changes we must make in Algorithms 3 and 4 are the following: In Algorithm 3 replace the two statements

$$\text{“}\nu > \xi\text{”} \quad \text{and} \quad \text{“}\hat{a}_2 > \hat{b}_2\text{”}$$

by

$$\text{“}\nu > \left(\frac{16}{25} + \xi^2\right)^{1/2}\text{”} \quad \text{and} \quad \text{“}\hat{a}_2 > \left(\frac{16}{25} + \hat{b}_2^2\right)^{1/2}\text{”},$$

respectively, and in Algorithm 4 replace the three statements

$$\text{“}\nu_{\text{new}} > \xi_{\text{new}}\text{”} \quad \text{and} \quad \text{“}\hat{a}_j > \hat{b}_j,\text{”} \quad j = s, 1,$$

by

$$\text{“}\nu_{\text{new}} > \left(\frac{16}{25} + \xi_{\text{new}}^2\right)^{1/2}\text{”} \quad \text{and} \quad \text{“}\hat{a}_j > \left(\frac{16}{25} + \hat{b}_j^2\right)^{1/2}\text{”}, \quad j = s, 1,$$

respectively. Since the output of either algorithm will be again $\hat{\rho}$, $\hat{a}$, $\hat{b}$, the optimum $\omega, \hat{\omega} \in (0, \frac{5}{4}]$, will be obtained from (2.5). So, in general, two values for $\hat{\omega}_2 \in ((1 - \sqrt{5})/2, (1 + \sqrt{5})/2)$ will be obtained from (5.8) and two values for $\hat{\omega}_1$ from (5.7). However, if $\hat{\omega} = \frac{5}{4}$, then $\hat{\omega}_2 = \frac{1}{2}$ and $\hat{\omega}_1 = \frac{3}{2}$.

Acknowledgments. The authors express their sincere thanks to the referee for his very constructive criticism. They also thank Dr. Emmanuel Vavalis of Purdue University for his valuable help in connection with § 4.5 of this paper.

REFERENCES

[1] G. AVDELAS, J. DE PILLIS, A. HADJIDIMOS, AND M. NEUMANN, *A guide to the acceleration of iterative methods whose iteration matrix is nonnegative and convergent*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 329–342.

[2] G. AVDELAS, S. GALANIS, AND A. HADJIDIMOS, *On the optimization of a class of second order iterative schemes*, BIT, 23 (1983), pp. 50–64.

[3] G. AVDELAS AND A. HADJIDIMOS, *Optimum second order stationary extrapolated iterative schemes*, Math. Comput. Simulation, 25 (1983), pp. 189–198.

[4] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[5] R. C. Y. CHIN AND T. A. MANTEUFFEL, *An analysis of block successive overrelaxation for a class of matrices with complex spectra*, SIAM J. Numer. Anal., 25 (1988), pp. 564–585.

[6] E. D'SYLVA AND G. A. MILES, *The* SSOR *iteration scheme for equations with* σ_1 *ordering*, Comput. J., 6 (1963), pp. 271–273.

[7] M. EIERMANN, X.-Z. LI, AND R. S. VARGA, *On hybrid semi-iterative methods*, paper presented at the Third International Congress on Computational and Applied Mathematics, Leuven, Belgium, July 1988.

[8] S. GALANIS, A. HADJIDIMOS, AND D. NOUTSOS, *On an* SSOR *matrix relationship and its consequences*, Comput. Methods Appl. Mech. Engrg., 27 (1989), pp. 559–570.

[9] L. A. HAGEMAN AND D. M. YOUNG, *Applied Iterative Methods*, Academic Press, New York, 1981.

[10] B. KREDELL, *On complex successive overrelaxation,* BIT, 2 (1962), pp. 143–152.

[11] M. S. LYNN, *On the equivalence of* SOR, SSOR *and* USSOR *as applied to* σ_1*-ordered systems of linear equations*, Comput. J., 7 (1964), pp. 72–75.

[12] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[13] R. S. VARGA, W. NIETHAMMER, AND D.-Y. CAI, *p-cyclic matrices and the successive overrelaxation method*, Linear Algebra Appl., 58 (1984), pp. 425–439.

[14] D. M. YOUNG, *Iterative methods for solving partial difference equations of elliptic type*, Trans. Amer. Math. Soc., 76 (1954), pp. 92–111.

[15] ———, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

[16] ———, personal communication, 1988.

[17] D. M. YOUNG AND H. E. EIDSON, *On the determination of the optimum relaxation factor for the* SOR *method when the eigenvalues of the Jacobi matrix are complex*, Report CNA-1, Center for Numerical Analysis, University of Texas, Austin, Texas, 1970.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 632–643, October 1990

013

ESTIMATING THE SENSITIVITY OF THE ALGEBRAIC STRUCTURE OF PENCILS WITH SIMPLE EIGENVALUE ESTIMATES*

DANIEL BOLEY†

Abstract. The sensitivity of the algebraic (Kronecker) structure of rectangular matrix pencils to perturbations in the coefficients is examined. Eigenvalue perturbation bounds in the spirit of Bauer–Fike are used to develop computational upper and lower bounds on the distance from a given pencil to one with a qualitatively different Kronecker structure.

Key words. matrix pencil, controllability, sensitivity, distance to uncontrollability, linear dynamical systems, Kronecker canonical form

AMS (MOS) subject classifications. 15A22, 15A21, 93B05, 93B40, 65F30

1. Introduction.

In this paper, the sensitivity of the algebraic (Kronecker) structure of rectangular matrix pencils to perturbations in the coefficients is examined. Eigenvalue perturbation bounds in the spirit of Bauer–Fike are used to develop computational upper and lower bounds on the distance from a given pencil to one with a qualitatively different Kronecker structure. A note on notation: All norms $\|\cdot\|$ used in this paper are the vector or matrix 2-norm, as appropriate.

The main goal of this paper is to present some results regarding matrix pencils, of the form $A - \lambda B$, where λ is a free parameter and A, B are $n \times p$ matrices with $n > p$. In the classical theory of matrix pencils [8], [11], it is well known that any pencil is equivalent to its *Kronecker Canonical Form* (KCF), which is a pseudodiagonal matrix with diagonal blocks of the form L, L^T, and/or J, where

$$L = \begin{bmatrix} I \\ [0, \cdots, 0] \end{bmatrix} + \lambda \begin{bmatrix} [0, \cdots, 0] \\ I \end{bmatrix}$$

is a matrix with one more row than column, and J is a square matrix in Jordan Canonical Form. We call L a "tall-thin" K-block, L^T a "short-fat" K-block, and J the "regular" part.

In this paper, we deal exclusively with tall-thin pencils. Such pencils always have at least $n - p$ tall-thin K-blocks. In [3], we showed that the set of all tall-thin pencils with only tall-thin K-blocks is open and dense in the set of all pencils of the same shape. Hence, given a tall-thin pencil, the question we attempt to address is if it has any other types of K-blocks, and if not, what is the distance to the nearest pencil which does. In [15] and [10], algorithms were proposed that compute the complete KCF for a given pencil guaranteed to be exact for a pencil close to the original given pencil (backward stable). If the KCF computed in this way has only tall-thin K-blocks (the "generic case"), then one is still left with determining how far it is from a pencil with other types of K-blocks. In this paper, we attempt to estimate this distance from both above and below. A detailed algebraic analysis for square pencils was given by Waterhouse [17], but beyond that surprisingly little has been found in the literature on this topic.

* Received by the editors June 27, 1988; accepted for publication (in revised form) November 2, 1989.

† Computer Science Department, University of Minnesota, Minneapolis, Minnesota 55455 (boley@umn-cs.cs.umn.edu). This research was partially supported by National Science Foundation grants DCR-8420935, CCR-8813493, and DCR-8519029.

We use the following characterization of the Kronecker structure of a pencil.

DEFINITION 1. A matrix pencil $A-\lambda B$ is said to be *deficient* if there exists some λ for which it is not full rank, where λ is a complex number or "infinity." If $A-\lambda B$ is always full rank for any value of λ, then it is said to be *nondeficient.*

All tall-thin pencils have at least one tall-thin (L) K-block. The condition that the pencil be deficient is equivalent to the existence of at least one value λ and vector $\mathbf{x}$ such that $(A-\lambda B)\mathbf{x}=0$, and it corresponds to the existence of at least a regular part (J) or a short-fat (L^T) block. We call such a vector $\mathbf{x}$ a right *annihilating vector* of the pencil associated with the *annihilating value* λ. These are also a generalized eigenvector and value, respectively, if they are associated with the regular part, or if there is no short-fat part. If there is a short-fat part, then every complex number (including infinity) is an annihilating value, but only a finite number of these can be generalized eigenvalues as well. The eigenvalues, if any, will be exactly those values of λ at which the matrix $A-\lambda B$ has a rank less than the overall maximum rank.

The work in this paper was motivated by the many roles matrix pencils play in control systems theory. We give one example below. Matrix pencils also play roles in the theory of transmission zeros and in the theory of differential algebraic equations.

Consider a time-invariant linear system

$$\dot{\mathbf{x}} = F\mathbf{x}+G\mathbf{u}; \mathbf{y} = H\mathbf{x}+D\mathbf{u}. \tag{1}$$

A classical result from control theory is the Popov–Belevitch–Hautus (PBH) test (see, e.g., [11]), which states that the system (1) is controllable if and only if the matrix pencil

$$P^T(\lambda) = [\lambda I - F \mid G] = [-F \mid G] - \lambda[-I \mid 0] \tag{2}$$

has full rank for any complex value of λ. From a numerical point of view, one may say that if pencil (2) has a small singular value for some value of λ, then a small perturbation to the coefficients to (1) can yield an uncontrollable system [12].

We mention the main results from the perturbation theory of eigenvalues that we use in this paper. The most important result is the modified Bauer–Fike theorem, which gives bounds on the changes of the eigenvalues under perturbations in the underlying matrix.

PROPOSITION 1 (modified Bauer–Fike theorem [6], [9]). *We are given an $n\times n$ matrix A with a complete set of eigenvalues $\lambda_1,\cdots,\lambda_n$ and corresponding left and right eigenvectors $\mathbf{w}_1,\cdots,\mathbf{w}_n$, $\mathbf{v}_1,\cdots,\mathbf{v}_n$. Let $V := [\mathbf{v}_1,\cdots,\mathbf{v}_n]$ be the matrix of eigenvectors. Let Δ be another arbitrary $n\times n$ matrix, and let $\bar{\lambda}$ be any eigenvalue of $A+\Delta$. Then for at least one λ_i, $1\le i\le n$, the following bound holds:*

$$|\bar{\lambda}-\lambda_i| \le K_i\|\Delta\|, \tag{3}$$

where

$$K_i \equiv \min\left(\|V\|\cdot\|V^{-1}\|, \frac{n}{s_i}\right) \text{ with } s_i \equiv \frac{|\mathbf{w}_i^H\mathbf{v}_i|}{\|\mathbf{w}_i\|\cdot\|\mathbf{v}_i\|}. \tag{4}$$

On (4) we remark that $\max_i s_i^{-1} \le \|V\|\cdot\|V^{-1}\| \le s_1^{-1}+\cdots+s_n^{-1} \le n\cdot\max_i s_i^{-1}$ [18, pp. 88-89], so that these quantities are closely related. We use this definition for K_i instead of just $\|V\|\cdot\|V^{-1}\|$ as in the original Bauer–Fike theorem because for some i this may yield a somewhat tighter bound. The bounds will be noticeably tighter only for those i for which s_i is much larger than some other s_j, $j\neq i$, if there are any.

We also use the following results regarding the changes to the eigenvectors under perturbations to a matrix A. We quote three different, but related, bounds in the following proposition and compare how they fare in the context of our matrix pencil problem.

PROPOSITION 2 (Stewart [14], Boley (Appendix), Demmel [4], [5]). *Let A be some arbitrary $n \times n$ matrix, which we assume for simplicity has distinct eigenvalues. Let $\mathbf{v}_i$, $i = 1, \cdots, n$ be the eigenvectors of A, all of unit length. Let $A + \Delta$ be another arbitrary matrix, and let $\bar{\mathbf{v}}$ be some eigenvector of $A + \Delta$. Let θ_i be the angle between $\bar{\mathbf{v}}$ and $\mathbf{v}_i$, for $i = 1, \cdots, n$. Finally, define* $\text{isep}_A(\lambda_i)$ [14] *as* $\|(R_{22} - \lambda_i I)^{-1}\|$, *where R_{22} is the trailing $(n-1) \times (n-1)$ block in the the Schur decomposition of A:*

$$P^H A P = R = \begin{bmatrix} \lambda_i & R_{12} \\ 0 & R_{22} \end{bmatrix}.$$

Then if $\|\Delta\|$ is small enough to satisfy the condition given below for all i, then the tangent of at least one angle θ_i can be bounded by the corresponding expression. We have three closely related bounds:

(a) (Stewart [14]) *If for all i*

$$\|\Delta\| \leq \frac{1}{4 \cdot \text{isep}_A(\lambda_i) \cdot (1 + \|A\| \cdot \text{isep}_A(\lambda_i))} \tag{5a}$$

then for at least one i

$$\tan\theta_i \leq \gamma_i^{\mathbf{a}} \equiv \frac{\text{isep}_A(\lambda_i)\|\Delta\|}{1 - 2 \cdot \text{isep}_A(\lambda_i)\|\Delta\|}. \tag{6a}$$

(b) (See Appendix) *If for all i*

$$\|\Delta\| \leq \frac{1}{\text{isep}_A(\lambda_i) \cdot (1 + K_i)} \tag{5b}$$

then for at least one i

$$\tan\theta_i \leq \gamma_i^{\mathbf{b}} \equiv \frac{\text{isep}_A(\lambda_i)\|\Delta\|}{1 - \text{isep}_A(\lambda_i)\|\Delta\|(1 + K_i)}. \tag{6b}$$

Note that (5b) *means that* (6b) *applies whenever the denominator is positive.*

(c) (Demmel [4], [5]) *If for all i*

$$\|\Delta\| \leq \frac{1}{4 \cdot \text{isep}_A(\lambda_i) \cdot s_i^{-1}} \tag{5c}$$

then for at least one i

$$\tan\theta_i \leq \gamma_i^{\mathbf{c}} \equiv \frac{4 \cdot \text{isep}_A(\lambda_i) \cdot \|\Delta\|}{1 - 4 \cdot \text{isep}_A(\lambda_i) \cdot \|\Delta\| \cdot \sqrt{s_i^{-2} - 1}}. \tag{6c}$$

In each case above, $\bar{\mathbf{v}}$ can be scaled so that for some i

$$\|\bar{\mathbf{v}} - \mathbf{v}_i\| = \sin\theta_i \leq \frac{\gamma_i^{\mathbf{x}}}{\sqrt{(\gamma_i^{\mathbf{x}})^2 + 1}}$$

where γ_i^{x} *is defined by* (6x), x=a,b,c, *whenever these formulas apply.*

Asymptotically as $\|\Delta\|$ goes to zero, all three bounds are the same, at least qualitatively, but we mention all three because each may yield the tighter bound for different values of $\|\Delta\|$. For example, it is evident that (6a) is tighter than (6b) when they both apply according to (5a) and (5b); but when (6a) does not apply, (6b) may still apply and hence be the tighter bound. Likewise, since the limit (5c) is the largest, the bound (6c) applies over the widest range for $\|\Delta\|$; it can, however, be less tight than (6a) and/or (6b) when they all apply. The numerical examples below will illustrate how one bound is best in some cases and another bound is best in other cases, but qualitatively they are all similar. As the anonymous reviewers pointed out, all these bounds can, and should, be further refined.

The rest of this paper is organized as follows. First we examine a method for computing whether or not a given pencil is deficient. Next we develop an upper bound for the distance to the nearest deficient pencil, and finally we develop a lower bound for this distance, using the eigensystem perturbation theory outlined above. We end with some numerical examples and conclusions. In the Appendix, we briefly sketch the derivation of the eigenvector bound (6b).

2. Find pencil rank deficiency. In this section we address the problem of determining whether a given rectangular pencil is deficient or not. Specifically, given an $n \times p$ pencil $A - \lambda B$, with $n > p$, determine whether or not $A - \lambda B$ loses rank for any λ, including possibly λ infinite. This is equivalent to asking whether or not the pencil has any short-fat K-blocks or regular part. If B has full column rank, and the pencil never loses rank for any finite value of λ, then there are no short-fat blocks and no regular part.

Consider an $n \times p$ pencil $A - \lambda B$ with $n > p$. Choose arbitrary $n \times (n-p)$ matrices C, D. We can then examine the square $n \times n$ generalized eigenvalue problem

$$[A, C]\mathbf{v} = \lambda[B, D]\mathbf{v}. \tag{7}$$

We partition the vector $\mathbf{v}$ as $\mathbf{v}^T \equiv [\mathbf{x}^T, \mathbf{y}^T]$, where $\mathbf{x}$ is a p-vector, and $\mathbf{y}$ is an $(n-p)$-vector. It is then a simple matter to derive the following proposition.

PROPOSITION 3. *Given an* $n \times p$ *pencil* $A - \lambda B$ *with* $n > p$, *and given arbitrary full-rank* $n \times (n-p)$ *matrices* C, D, *the following are equivalent:*

(a) $A - \lambda B$ *is a deficient pencil.*

(b) *Equation* (7) *has an annihilating vector* $\mathbf{v}_0$ *whose last* $n-p$ *components* $\mathbf{y}_0$ *are zero. Call the corresponding annihilating value* λ_0.

Furthermore, we have the following:

(c) *If* B *has full column rank, then all the annihilating vectors* $\mathbf{v}_0$ *and corresponding values* λ_0 *are exactly the generalized eigenpairs for the regular part of the pencil.*

Proof. Both (a) and (b) are equivalent to the following statement:

(d) There is an n-vector $\mathbf{x}$ and scalar λ_0 which satisfies $A\mathbf{x} = \lambda_0 B\mathbf{x}$, or else $B\mathbf{x} = 0$. In the latter case, we say $\lambda_0 = \infty$. If B has full column rank, then there can be no short-fat blocks. Hence it follows that there is a regular part, and that the λ_0's are exactly the eigenvalues of that regular part. Otherwise, there is no regular part. □

Based on this proposition, we have a simple procedure for computing the existence of a regular part or short-fat block in a pencil. Given a tall-thin $n \times p$ pencil, choose the $n \times (n-p)$ augmentation matrices C, D to obtain the square eigenvalue problem

(7). If there are any annihilating vectors of (7) whose last $n-p$ entries are zero, then there is a regular part or short-fat block; otherwise, there is not.

In the special case that $B = [I, 0]^T$, choose $D = [0, I]^T$ to turn (7) into an ordinary eigenproblem, and the annihilating vectors above into ordinary eigenvectors. If the last $n-p$ entries of any of those eigenvectors are zero, then the corresponding eigenvalues are exactly the eigenvalues of the regular part of the original pencil. Otherwise, there is no regular part.

However, this method gives no hint as to the sensitivity of the result to perturbations in the coefficients. Therefore, in the next sections, we develop some bounds that indicate whether a given pencil is "numerically close" to a deficient pencil.

3. Upper bounds. In this section, we examine the problem of computing an upper bound on the distance to a deficient pencil. Specifically, consider a nondeficient $n \times p$ pencil $A - \lambda B$. In this case, we know that B has full rank. We would like to estimate the size of the perturbation E to the matrix A that is needed to obtain a deficient pencil $A + E - \lambda B$. This perturbed pencil will have a regular part, but no "short-fat" blocks. In this section we develop a simple upper bound for $\|E\|$.

In [7] and [12], it was shown that the smallest perturbation E can be obtained by solving the minimization problem

$$\min_s \sigma_{\min}(A - sB), \tag{8}$$

where $\sigma_{\min}(M)$ denotes the smallest singular value of the matrix M, and s varies over the entire complex plane. If we denote by σ^* and s^* the minimum in (8) and the value of s achieving that minimum, respectively, then $\|E\| = \sigma^*$. In [3], we discussed an expensive descent method that would often converge to the minimum (8). In this section, we would like to address a much simpler scheme that can be used to obtain an upper bound, which often not only provides a good estimate for $\|E\|$, but also provides an estimate for that value of s that yields the minimum in (8).

We start with the $n \times p$ pencil $A - \lambda B$. Choose some arbitrary full-rank $n \times (n-p)$ matrices C, D. And, in the case $B = [I, 0]^T$, choose $D = [0, I]^T$. Let

$$\lambda_i, \mathbf{v}_i \equiv \begin{bmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{bmatrix}, \qquad i = 1, \cdots, k \tag{9}$$

be the generalized eigenvalues and vectors for (7), where $\mathbf{x}_i$ denotes the first p components of $\mathbf{v}_i$. For each i, we have the equation $[A, C]\mathbf{v}_i = \lambda_i [B, D]\mathbf{v}_i$. We can rewrite this as $(\lambda_i B - A)\mathbf{x}_i = (C - \lambda_i D)\mathbf{y}_i$. We define the residual for each i by $\mathbf{r}_i := (A - \lambda_i B)\mathbf{x}_i$, and the perturbation E_i to be

$$E_i := -\frac{\mathbf{r}_i \mathbf{x}_i^T}{\|\mathbf{x}_i\|^2} \equiv (\lambda_i B - A)\frac{\mathbf{x}_i \mathbf{x}_i^T}{\|\mathbf{x}_i\|^2} = (C - \lambda_i D)\frac{\mathbf{y}_i}{\|\mathbf{x}_i\|} \cdot \frac{\mathbf{x}_i^T}{\|\mathbf{x}_i\|}. \tag{10}$$

Then $A + E_i - \lambda B$ is a deficient pencil, losing rank exactly at $\lambda = \lambda_i$, for each i. Let σ_i, $\mathbf{u}_i$, $\mathbf{w}_i$ be, respectively, the smallest singular value and the corresponding left and right singular vectors of $A - \lambda_i B$, for each i. Then $E_i' := -\sigma_i \mathbf{u}_i \mathbf{w}_i^T$ is another smaller perturbation yielding a deficient pencil.

By taking norms of (10), we obtain a bound for these perturbations: $\|E_i'\| \leq \|E_i\| \leq \|(C - \lambda_i D)\mathbf{y}_i\|/\|\mathbf{x}_i\|$. If E denotes that perturbation with smallest norm yielding a deficient pencil, then E satisfies

$$\begin{aligned} \|E\| \quad &\equiv \sigma^* \leq \beta_2 \equiv \min_i \|E_i'\| \equiv \min_i \sigma_{\min}(A - \lambda_i B) \\ &\leq \beta_1 \equiv \min_i \|E_i\| \equiv \min_i \frac{\|(C - \lambda_i D)\mathbf{y}_i\|}{\|\mathbf{x}_i\|}. \end{aligned} \tag{11}$$

Regarding the two bounds β_1, β_2, we remark that β_1 can be computed directly from the solution to the eigenproblem (7), whereas β_2 requires computing the singular value decomposition (SVD) at some extra expense. We can summarize the result in the following proposition.

PROPOSITION 4. *Let $A - \lambda B$ be an $n \times p$ pencil, with $n > p$. Let C, D be two arbitrary full-rank $n \times (n-p)$ matrices. Then the smallest perturbation E such that $A + E - \lambda B$ is a deficient pencil satisfies the bound* (11), *where $\lambda_i, \mathbf{v}_i$, $i = 1, \cdots, k$ are the eigenpairs of the generalized eigenproblem* (7), *and $\mathbf{y}_i$ are defined by* (9).

Proof. This follows from the above discussion. All we must note is that from Proposition 3, if the pencil $A - \lambda B$ is already deficient, then $E = 0$ automatically satisfies (11). In fact, if B has full column rank, one of the $\mathbf{y}_i$ should be zero by Proposition 3, so the bound will be hard. □

One question is how to choose C, D. One goal is to make the augmented square eigenproblem as well conditioned as possible. So far, the only requirement we have stated is that C, D have full column rank. To keep the condition number as low as possible, it is best to choose C, D to each have orthonormal columns. Two possible choices are (a) orthonormal basis of a random space, and (b) orthonormal basis of the space orthogonal to the columns of A and B. This last choice has the effect of limiting the increase to the condition numbers of $[A, C]$ and $[B, D]$ with respect to inversion, and hence is a heuristic attempt to obtain a reasonably low condition number with respect to the eigenproblem. In any case, the algorithms are intended to provide a posteriori estimates for a given pencil, and in that context it is easy to check that the condition number of the resulting eigenproblem is reasonably small. Most of the numerical examples below were carried out with choice (b). We note that in the special case $B = [I, 0]^T$, we choose $D = [0, I]^T$ to turn (7) into an ordinary eigenproblem.

4. Lower bounds. In this section, we show how to extend the results of the previous section for the special case of pencils $A - \lambda B$, such that $B = [I, 0]^T$, to obtain some lower bounds and to obtain a disk in the complex plane in which the value s, achieving the minimum in (8), must be located. The first lower bound is based just on the Bauer–Fike theorem whereas the other lower bounds are based on the eigenvector perturbation theorem (Proposition 2). It will be seen that the first lower bound is not as tight as the others, but it is much simpler to derive and much cheaper to compute, since it does not require the "isep" function.

Let s^* be the complex value achieving the minimum in (8), and let σ^* be the smallest singular value of $A - s^* B$. Augment $A - sB$ as before with extra columns C and $D = [0, I]^T$, obtaining the square matrix $[A, C]$, so that (7) becomes the ordinary eigenproblem for the matrix $[A, C]$. Then the smallest singular value τ of $[A, C] - s^* I$ satisfies $\tau \leq \sigma^*$, since augmenting with extra columns can only reduce the smallest singular value [9]. So s^* is an exact eigenvalue of $[A, C] + \tau\Delta$, for some matrix Δ such that $\|\Delta\| = 1$. Denote the eigenvalues of $[A, C]$ by $\lambda_1, \cdots, \lambda_n$. Then, for at least one such eigenvalue, the modified Bauer–Fike theorem implies that $|\lambda_i - s^*| \leq \tau K_i$, where K_i is defined in (4).

Next, let α be the smallest singular value of $A - \lambda_i B$. Then

$$|\alpha - \sigma^*| \leq \|(A - \lambda_i B) - (A - s^* B)\| \leq |\lambda_i - s^*| \leq \tau K_i \leq \sigma^* K_i. \tag{12}$$

From this formula, we can draw two conclusions. One is that $\alpha \leq \sigma^*(K_i + 1)$, yielding

the lower bound on σ^*:

$$\sigma^* \geq \frac{\alpha}{K_i + 1} \geq \frac{\beta_2}{K_i + 1}, \tag{13}$$

where β_2 is the upper bound β_2 defined in (11).

The other conclusion from (12) is

$$|\lambda_i - s^*| \leq \sigma^* K_i \leq \beta_2 K_i. \tag{14}$$

We can summarize this in the following proposition.

PROPOSITION 5. *Given a pencil $A - sB$, where $B = [I, 0]^T$, and an arbitrary (full-rank) augmentation of this pencil to a square matrix $[A, C]$ as in* (7), *then*

(a) *The value of s that achieves the minimum in $\sigma^* \equiv \min_s \sigma_{\min}(A - sB)$ is located within a disk in the complex plane whose center is on an eigenvalue λ_i of Q and whose radius is bounded by* (14), *for some i.*

(b) *A lower bound on σ^* is provided by* (13).

By using the eigenvector bounds in Proposition 2, we can derive some tighter lower bounds on σ^*. We base our development on (5a) and (6a), but by analogy the exact same development goes through with (5b) and (6b) or with (5c) and (6c). Given a pencil $A - sB$ with $B = [I, 0]^T$, the eigenvectors (9) of the augmented matrix $[A, C]$ are defined. If $A + E - sB$ is a deficient pencil, then the matrix $[A + E, C]$ must have at least one eigenvector of the form $\bar{\mathbf{v}} = [\bar{\mathbf{x}}^T, 0]^T$, where $\bar{\mathbf{v}}$ is partitioned as in (9). That is, the square matrix $[A, C]$ must be perturbed to a matrix which has an eigenvector $\bar{\mathbf{v}}$ whose $\mathbf{y}$ part is zero. But then the bounds (5a) and (6a) directly yield a lower bound on the norm of the perturbation to $[A, C]$ so that an eigenvector of the resulting matrix has the indicated form. The resulting lower bound is

$$\sigma^* \geq \delta^{\mathbf{a}} \equiv \min_i \min \left\{ \begin{array}{ll} \text{(i)} & \dfrac{[\text{isep}_A^{-1}(\lambda_i)]^2}{4(\text{isep}_A^{-1}(\lambda_i) + \|A\|)}, \\ \text{(ii)} & \dfrac{\eta_i}{\text{isep}_{[A,C]}(\lambda_i)(1 + 2\eta_i)} \end{array} \right\}, \tag{15a}$$

where

$$\eta_i \equiv \frac{y_i}{\sqrt{1 - y_i^2}}.$$

Bounds (i) and (ii) come from (5a) and (6a), respectively. Alternatively, we can use part (b) of Proposition 2 (this always satisfies the limit (5b)):

$$\sigma^* \geq \delta^{\mathbf{b}} = \min_i \frac{\eta_i}{\text{isep}_{[A,C]}(\lambda_i)(1 + \eta_i(1 + K_i))} \tag{15b}$$

or part (c) of Proposition 2:

$$\sigma^* \geq \delta^{\mathbf{c}} = \min_i \min \left\{ \begin{array}{ll} \text{(i)} & \dfrac{1}{4 \cdot \text{isep}_{[A,C]}(\lambda_i) \cdot s_i^{-1}}, \\ \text{(ii)} & \dfrac{\eta_i}{4 \cdot \text{isep}_{[A,C]}(\lambda_i)(1 + \eta_i \sqrt{s_i^{-2} - 1})} \end{array} \right\}. \tag{15c}$$

We note that the backward stability of these methods depends on the backward stability of the method used to obtain the eigendecompositions. If the eigenvalue

method is backward stable, then these bounds will be exact for a pencil numerically close to the original pencil, and the residual from the eigenvalue method will indicate how far from that original pencil we have strayed. But in the cases we have tried, the size of this residual was never more than 1E-11, much less than the computed bounds themselves.

5. Numerical examples. We illustrate the bounds with the examples taken from [3]. Each example represents a time-invariant linear system of the general form $\dot{\mathbf{x}} = F\mathbf{x} + G\mathbf{u}$, from which we form the pencil (2). Example 1 is defined by

$$F = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \text{ and } G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

In Example 2, we start with a single-input system already in staircase form [13], [15], with $G = [1, 0, \cdots, 0]^T$, and

$$F = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & 7 \\ 1 & -1 & -1 & -1 & -1 & -1 & 6 \\ 0 & 1 & -1 & -1 & -1 & -1 & 5 \\ 0 & 0 & 1 & -1 & -1 & -1 & 4 \\ 0 & 0 & 0 & 1 & -1 & -1 & 3 \\ 0 & 0 & 0 & 0 & 1 & -1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Example 3 is one with a particularly ill-conditioned eigenvalue problem. The system is defined by

$$F = \begin{bmatrix} -149 & 537 & -27 \\ -50 & 180 & -9 \\ -154 & 546 & -25 \end{bmatrix} \text{ and } G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

The poles (eigenvalues) for this system are 1, 2, and 3.

The staircase algorithm [13], [1] applied to a single input system (i.e., G has only one column) transforms F into an upper Hessenberg form, with G a multiple of $\mathbf{e}_1$. The pencil (2) has a regular part if and only if a subdiagonal element of the Hessenberg form is zero, so an obvious upper bound is simply the magnitude of the smallest subdiagonal element. This is the second column of Table 1. In the third column are shown upper bounds obtained by the expensive experimental descent method described in [3]. In the last column of Table 1 are lower bounds from the theory in [2], which was based on using the product of the subdiagonal elements.

We report in Table 2 the upper bounds computed using the formula (11). It is seen that the bound β_2 is always tighter than β_1 and is in fact fairly close to the "optimal" upper bound reported in Table 1. In Table 3 we report the various lower bounds. For reference, we copied the tightest lower bound found for each example to Table 2 to show the spread between the upper and lower bounds. It is seen that the best lower bound is obtained from formula (15a) (ii) when it applies; otherwise, the tightest bound is obtained from (15b). We note that upper bound 1 and lower bound 0 take only $O(n+p)^3$ work to obtain, so the computation is relatively fast. Upper bound 2 requires computing the smallest singular value at some extra expense, but we do not address here possible ways to speed this up. The other lower bounds would also be fast to obtain, except for the computation of the rather expensive "isep"

function. Note that the upper bounds from [3] in Table 1 are tighter, but are much more expensive to obtain.

In Table 4, we give the eigenvalue λ_i of the augmented matrix at which the upper bounds were taken, together with the radius (14) about this value within which the minimum in (8) is located.

TABLE 1

Bounds from older methods.

Example #	Upper bound staircase [1]	Upper bound from [3]	Lower bound from [2]
1	1.0	6.6144E-01	1.2500E-01
2	1.0	6.7690E-04	4.3654E-08
3	1.1610E-02	4.3715E-03	8.5774E-07

TABLE 2

Upper bounds from formula (11).

Example #	Upper bound 1 β_1	Upper bound 2 β_2	Best lower bound from Table 3
1	7.2561E-01	7.0545E-01	3.7272E-01 (15b)
2	8.8790E-04	7.3074E-04	6.5105E-04 (15a)
3	1.1507E-02	4.6607E-03	1.0313E-03 (15b)

TABLE 3

Lower bounds from methods in this paper. Bounds from equation (15) *come from* (ii) *except those marked* "*".

Example #	Lower bound 0 (13)	Lower bound a (15a)	Lower bound b (15b)	Lower bound c (15c)
1	3.1480E-01	*1.9409E-01	3.7272E-01	1.7264E-01
2	7.2095E-05	6.5105E-04	6.4726E-04	1.6279E-04
3	8.6385E-04	*1.7989E-05	1.0313E-03	2.6339E-04

TABLE 4

Values of λ *at which upper bounds in Table* 2 *were obtained.*

Example #	λ_i achieving min in $min_i\, \sigma_{min}(A - \lambda_i B) \equiv \beta_2$ (11)	Radius about given λ_i (14)
1	$-1.6899\text{E}{-01}+1.1509\text{E}{+00}i$	8.7545E−01
2	$+1.9998\text{E}{+00}+6.5937\text{E}{-16}i$	6.6758E−03
3	$+2.4534\text{E}{+00}+0.0000\text{E}{+00}i$	2.0485E−02

When the staircase algorithm is applied to Example 3, we obtain (items in parenthesis are close to the machine epsilon) the following:

Fnew =

2.8300E+02	6.9026E+02	-1.3400E+02	-1.7321E+00
-1.1458E+02	-2.7946E+02	5.4837E+01	(-2.5339E-16)
(-1.6584E-15)	-1.1610E-02	2.4570E+00	(4.2062E-16)

Gnew =

In Table 5, we illustrate the effect of using a different way to augment the matrix rather than an orthonormal basis to the space orthogonal to the column space of A. These numbers were obtained with Example 2. The first two lines were obtained using an orthonormal basis for two different random spaces. The third line was obtained by using random columns, not orthonormal, but with elements uniformly distributed in the interval $[-1, 1]$. The fourth line was obtained by adding a random perturbation to A of norm 1E-5 and then following the original prescription used for Tables 2 and 3. The fifth line was copied from Table 2 for comparison. Generally, the bounds from Table 2 are at least as tight, except for the SVD-based upper bound β_2 (11), for which using a random set of orthonormal columns was better.

TABLE 5
Bounds on Example 2 using different random schemes.

Example #	Upper bound 1 β_1 (11)	Upper bound 2 β_2 (11)	Lower bound 0 (13)	Best lower bound, all from (15a)
Rand 1	7.6641E-04	6.8228E-04	6.5806E-05	6.4203E-04
Rand 2	7.6697E-04	6.8038E-04	6.5224E-05	6.3977E-04
Non-ortho	8.9135E-04	6.8256E-04	6.3614E-05	6.0989E-04
Perturbed	8.8983E-04	7.3232E-04	7.2426E-05	6.5245E-04
Table 2	8.8790E-04	7.3074E-04	7.2095E-05	6.5105E-04

6. Conclusions. We have given a scheme for estimating the distance from a given pencil to the nearest pencil of different Kronecker structure. In the context of dynamical systems, this yields estimates of the distance to the nearest uncontrollable system. Unlike the staircase-type algorithms, the scheme presented here does not depend on the recursive computation of the singular values of small matrix subblocks, so it is less sensitive to the particular choice of zero tolerance. Though there is no a priori guarantee that the bounds obtained using the methods from this paper will be good, the spread between the upper and lower bounds will automatically give a measure of the quality of the bounds themselves. Furthermore, since we also have localized the location of the minimum in (8) to within certain small disks, we also have good values with which to start an iterative procedure to refine the estimate of the location of this minimum (for example, using the experimental descent method proposed in [3], for which a good starting value is critical for successful convergence).

Appendix. We briefly sketch the derivation of the bound (6b). Let A be a matrix with a simple eigenvalue λ and associated eigenvector $\mathbf{v}$, with $\|\mathbf{v}\| = 1$. We can then find a unitary matrix P whose first column is $\mathbf{v}$ such that

$$P^H A P = R = \begin{bmatrix} \lambda & R_{12} \\ 0 & R_{22} \end{bmatrix}, \tag{A1}$$

where R_{22} is an $(n-1) \times (n-1)$ matrix, none of whose eigenvalues equals λ. The eigenvector of R corresponding to λ is $\mathbf{e}_1 \equiv [1, 0, \cdots, 0]^T$. We examine how this eigenvector changes under perturbations E to R. Let E be a (small) perturbation matrix and let $\bar{\lambda}$ be any eigenvalue of $R + E$, with corresponding eigenvector $\mathbf{f} \equiv [1, \mathbf{x}^T]^T$, partitioned conformally with (A1), and which is scaled to have first component equal

to 1. If θ is the angle between $\mathbf{e}_1$ and $\mathbf{f}$, then $\|\mathbf{x}\| \equiv \tan\theta$. We then have the relation

$$0 = (R + E - \bar{\lambda}I)\mathbf{f} = \begin{bmatrix} \lambda + e_{11} - \bar{\lambda} & R_{12} + E_{12} \\ E_{21} & R_{22} + E_{22} - \bar{\lambda}I \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}, \tag{A2}$$

where

$$E \equiv \begin{bmatrix} e_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix}$$

is partitioned as in (A1). Define the $(n-1) \times (n-1)$ matrices $M \equiv R_{22} - \lambda I$, and $-N \equiv E_{22} + (\lambda - \bar{\lambda})I$, and note that M is nonsingular. If $\|M^{-1}N\| < 1$, then $(M-N)^{-1}$ exists and can be bounded by $\|(M-N)^{-1}\| \leq \|M^{-1}\|/(1 - \|M^{-1}N\|)$ [9]. Then the last $n-1$ equations of (A2) can be written as just $(M-N)\mathbf{x} = -E_{21}$, yielding the bound

$$\|\mathbf{x}\| \leq \frac{\|M^{-1}\| \cdot \|E_{21}\|}{\text{pos}(1 - \|M^{-1}N\|)} \leq \frac{\|M^{-1}\| \cdot \|E\|}{\text{pos}(1 - \|M^{-1}\|(\|E\| + |\lambda - \bar{\lambda}|))}, \tag{A3}$$

where "pos" is a function defined by $\text{pos}(r) \equiv r$ for $r > 0$, and $\text{pos}(r) \equiv 0$ for $r \leq 0$. The "pos" function simply expresses the fact that (A3) always holds, but only vacuously if the denominator is not positive. For example, this would occur if $\mathbf{f}$ is orthogonal to $\mathbf{e}_1$. We summarize the above in the following lemma.

LEMMA A1. *Let the upper triangular matrix R be partitioned as in* (A1), *$\lambda, \mathbf{e}_1$ be a simple eigenpair for R, $M \equiv R_{22} - \lambda I$, E be some arbitrary matrix, and $\bar{\lambda}, \mathbf{f}$ be any eigenpair for $R + E$. Then $\mathbf{f}$ can be scaled so that the difference $\mathbf{e}_1 - \mathbf{f}$ satisfies $\|\mathbf{e}_1 - \mathbf{f}\| = \sin\theta$, where $\tan\theta \equiv \|\mathbf{x}\|$ satisfies the bound* (A3).

We note that if one expands (A3) in a power series in $\|E\|$, the first-order term will be identical to the first-order bound in [16]. Since R and A are related by a unitary transformation, this lemma leads directly to a corresponding bound for the change in the eigenvectors for an arbitrary matrix A. We can apply the modified Bauer–Fike theorem directly to the lemma to obtain the following proposition. We define the inverse "separation" function (following [14]) to be $\text{isep}_A(\lambda) \equiv \|(R_{22} - \lambda I)^{-1}\|$.

PROPOSITION A1. *Let A be some arbitrary $n \times n$ matrix with all distinct eigenvalues, and denote by V the matrix of eigenvectors of A. Let $A + \Delta$ be another arbitrary matrix, and let $\bar{\lambda}, \bar{\mathbf{v}}$ be any eigenpair for $A + \Delta$. Let θ_i denote the angle between $\bar{\mathbf{v}}$ and the i-th unit eigenvector $\mathbf{v}_i$ of A, $i = 1, \cdots, n$. Then for some i we have the bound*

$$\tan\theta_i \leq \gamma_i^{\mathbf{b}} \equiv \frac{\text{isep}_A(\lambda_i)\|\Delta\|}{\text{pos}(1 - \text{isep}_A(\lambda_i)\|\Delta\|(1 + K_i))}. \tag{A4}$$

Furthermore, $\bar{\mathbf{v}}$ can be scaled so that we have the bound on the distance to some eigenvector $\mathbf{v}_i$ of A, for some i:

$$\|\mathbf{v}_i - \bar{\mathbf{v}}\| = \sin\theta_i \leq \frac{\gamma_i^{\mathbf{b}}}{\sqrt{1 + (\gamma_i^{\mathbf{b}})^2}}. \tag{A5}$$

Note that this gives a nonvacuous bound as long as $\|\Delta\|$ is small enough to make the denominator in (A4) *positive.*

REFERENCES

[1] D. BOLEY, A. EMAMI-NAEINI, G. F. FRANKLIN, *A New Algorithm for Canonical Decomposition of Linear Systems*, Proc. 19th IEEE Conference on Decision and Control, Albuquerque, NM, 1980, pp. 199-200.

[2] D. BOLEY, *A perturbation result for linear control problems*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 66-72.

[3] ———, *Computing the rank-deficiency of rectangular matrix pencils*, Systems Control Lett., 9 (1987), pp. 207-214.

[4] J. DEMMEL, *Computing stable eigendecompositions of matrices*, Linear Algebra Appl., 79 (1986), pp. 163-193.

[5] ———, *Computing stable eigendecompositions of matrix pencils*, Linear Algebra Appl., 88/89 (1987), pp. 139-186.

[6] ———, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math., 51 (1987), pp. 251-289.

[7] R. EISING, *Between Controllable and Uncontrollable*, Systems Control Lett., 4 (1984), pp. 263-264.

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

[9] G. H. GOLUB AND C. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[10] B. KÅGSTROM, RGSVD — *An algorithm for computing the Kronecker structure and reducing subspaces of singular* $A - \lambda B$ *pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185-211.

[11] T. KAILATH, *Linear Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1980.

[12] G. MIMINIS, *Numerical algorithms for controllability and eigenvalue computation*, Masters thesis, School of Computer Science, McGill University, Montréal, Québec, Canada, May 1981.

[13] C. C. PAIGE, *Properties of numerical algorithms related to controllability*, IEEE Trans. Automat. Control, AC-26, 1 (1981), pp. 130-138.

[14] G. W. STEWART, *Error and perturbation bounds associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727-764.

[15] P. VAN DOOREN, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103-141.

[16] J. VARAH, *Invariant subspace perturbations for a nonnormal matrix*, IFIP 71 Proceedings, North-Holland, Amsterdam, 1971.

[17] W. C. WATERHOUSE, *The codimension of singular matrix pairs*, Linear Algebra Appl., 57 (1984), pp. 227-245.

[18] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 4, pp. 644–659, October 1990

INCREMENTAL CONDITION ESTIMATION FOR SPARSE MATRICES*

CHRISTIAN H. BISCHOF†, JOHN G. LEWIS‡, AND DANIEL J. PIERCE ‡

Abstract. Incremental condition estimation provides an estimate for the smallest singular value of a triangular matrix. In particular, it gives a running estimate of the smallest singular value of a triangular factor matrix as the factor is generated one column or row at a time. An incremental condition estimator for dense matrices was originally suggested by Bischof. In this paper this scheme is generalized to handle sparse triangular matrices, especially those that are factors of sparse matrices. Numerical experiments on a variety of matrices demonstrate the reliability of this scheme in estimating the smallest singular value. A partial description of its implementation in a sparse matrix factorization code further illustrates its practicality.

Key words. incremental condition estimator, restricted pivoting schemes, updating condition estimates

AMS(MOS) subject classifications. 15A18, 65F35

1. Introduction.

Incremental condition estimation is a technique that allows one to compute an estimate for the smallest singular value of a triangular matrix. Its primary application is in providing a running estimate of the smallest singular value of a triangular factor matrix as it is generated one row or column at a time. It was introduced by Bischof [2] and has been used successfully for developing a rank-revealing dense QR factorization algorithm for distributed-memory machines [3] and shared-memory machines with a memory hierarchy [1]. It is also immediately applicable to matrices that are Cholesky factors.

For definiteness and without loss of generality, we will assume, throughout the remainder of this paper, that we are generating a lower triangular matrix L one row at a time. For upper triangular matrices generated one column at a time we would simply transpose the matrix, since a matrix and its transpose have identical singular values.

Given an approximate singular vector x of a lower triangular matrix L and a new row (w^T, γ) by which L is augmented, the incremental condition estimator for dense matrices allows us to obtain an estimate for the smallest singular value of the resulting lower triangular matrix

$$\begin{pmatrix} L & 0 \\ w^T & \gamma \end{pmatrix}$$

without accessing L again. As will be seen, the computational work involved consists only of computing the inner product $w^T x$, scaling x and computing the largest

* Received by the editors November 1, 1989; accepted for publication March 16, 1990. This paper was presented at the Symposium on Sparse Matrices at Salishan Resort, Gleneden Beach, Oregon, May 22-24, 1989, which was sponsored by the SIAM Activity Group on Linear Algebra.

† Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, Illinois 60439 (bischof@mcs.anl.gov). The work of this author was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U. S. Department of Energy under contract W-31-109-Eng-38.

‡ Boeing Computer Services, P.O. Box 24346, MS 7L-21, Seattle, Washington 98214-0346 (jglewis@atc.boeing.com and dpierce@atc.boeing.com).

eigenvector and associated eigenvector of a two-by-two symmetric eigenvalue problem.

In order to compute an incremental condition estimator for *sparse* matrices, it will be necessary and sufficient to show how to compute an estimate for the smallest singular value and vector of

$$\mathcal{L} = \begin{pmatrix} L_1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & L_p & 0 \\ w_1^T & \cdots & \cdots & w_p^T & \gamma \end{pmatrix},$$

where we are given lower triangular matrices $L_1, \ldots, L_p$, estimates for the smallest singular value and corresponding right singular vector of each L_i, and a new row $(w_1^T, \ldots, w_p^T, \gamma)$. This paper generalizes the dense incremental condition estimator to handle such cases, again without accessing any of the L_i.

The outline of the paper is as follows. Section 2 briefly reviews the dense incremental condition estimator and motivates its unsuitability for sparse matrices. The next section generalizes the dense condition estimator technique to handle sparse matrices and shows how it can be computed inexpensively. In the next section we provide a partial understanding of the success of the estimator by relating it to the secular equations for $\mathcal{L}^T\mathcal{L}$. In §5 we report on numerical experiments on a variety of matrices that demonstrate the reliability of the suggested scheme. In §6 we describe how the data required by the estimator can be acquired during the process of a sparse QR or Cholesky factorization. Lastly we summarize our contribution and outline applications of this scheme.

2. An incremental estimator for dense matrices. A common idea underlying condition estimators [6], [7], [14] is to exploit the implication

$$Lx = d \implies \frac{1}{\sigma_{\min}(L)} = \|L^{-1}\|_2 \geq \frac{\|L^{-1}d\|_2}{\|d\|_2} = \frac{\|x\|_2}{\|d\|_2}$$

by generating a solution x having large norm for a right-hand side d having norm of moderate size and then to use

$$\hat{\sigma}_{\min}(L) \equiv \frac{\|d\|_2}{\|x\|_2}$$

as an estimate for $\sigma_{\min}(L)$. The hope is that x will be an approximate singular vector corresponding to the smallest singular value and that, as a consequence, $\hat{\sigma}_{\min}(L)$ will not be too much of an overestimate of $\sigma_{\min}(L)$. An *incremental* condition estimator is one that allows us to easily update our estimate $\hat{\sigma}_{\min}$ as L is augmented with a new row. In particular, the previously generated L should not be accessed in updating our estimate. Accessing L would imply $O(n^2)$ flops at every step of our condition estimation algorithm, which is too great an expense. More precisely, given a good estimate $\hat{\sigma}_{\min}(L)$, defined by a large norm solution x to $Lx = d$, and a new row (w^T, γ), by which L is augmented, the estimator should obtain a large norm solution y to

$$\mathcal{L}y = \begin{pmatrix} L & 0 \\ w^T & \gamma \end{pmatrix} y = d'$$

without accessing L again. Bischof [2] suggested the following approach: Given x such that $Lx = d$ with $\|d\|_2 = 1$, find s and c with $c^2 + s^2 = 1$ that maximize $\|y\|_2$, where

$$y = \begin{pmatrix} \hat{x} \\ \beta \end{pmatrix}$$

solves

$$\begin{pmatrix} L & 0 \\ w^T & \gamma \end{pmatrix} y = \begin{pmatrix} sd \\ c \end{pmatrix}. \tag{1}$$

A formal forward substitution shows that

$$\gamma^2 \|y\|_2{}^2 = (s, c) B \begin{pmatrix} s \\ c \end{pmatrix},$$

where

$$B = \begin{pmatrix} \gamma^2 x^T x + (w^T x)^2 & -w^T x \\ -w^T x & 1 \end{pmatrix}. \tag{2}$$

It follows that B is a positive definite symmetric matrix. Thus, maximizing $\gamma^2 \|y\|_2{}^2$ is equivalent to

$$\begin{aligned} &\max (s, c) B \begin{pmatrix} s \\ c \end{pmatrix} \\ &\text{subject to } c^2 + s^2 = 1. \end{aligned} \tag{3}$$

The pair (s, c) that solves (3) is a normalized eigenvector of B corresponding to the largest eigenvalue of B. The optimal (s, c) can be computed easily (see [2]), and the new approximate singular vector y defined by (1) is then

$$y = \begin{pmatrix} \hat{x} \\ \beta \end{pmatrix} := \begin{pmatrix} sx \\ (c - s(w^T x))/\gamma \end{pmatrix}.$$

The resulting estimate for the smallest singular value $\sigma_{\min}(\mathcal{L})$ of $\mathcal{L}$ is

$$\hat{\sigma}_{\min}(\mathcal{L}) = \frac{1}{\|y\|_2}.$$

From this description it is clear that this condition estimator satisfies our constraints. Given a current L, we only need to save the current solution x and its norm $\|x\|_2$ to arrive at an estimate for $\sigma_{\min}(\mathcal{L})$. Furthermore, the calculation is inexpensive. For a $k \times k$ matrix L we need only $3k$ flops (a dot-product and a scaling of a vector) to arrive at an estimate for $\sigma_{\min}(\mathcal{L})$. Experimental results on the suite of tests suggested by Higham [15] are reported in [2], and they show this condition estimator to be reliable in producing good estimates.

However, this condition estimator breaks down for sparse matrices. To illustrate the issue, consider the matrix

$$\mathcal{L} = \begin{pmatrix} L_{11} & 0 \\ 0 & L_{22} \end{pmatrix} \tag{4}$$

where L_{11} is $n_1 \times n_1$ and L_{22} is $n_2 \times n_2$. During the first n_1 steps, the incremental condition estimator computes an approximate singular vector x_1 of L_{11}. On encountering the first row of L_{22}, we will have $w = 0$ and γ in (1) is the $(1,1)$ entry of L_{22}. So $w^T x$ in (2) will be zero, and we will choose either $(x_1, 0)^T$ or $(0, 1/\gamma)^T$ as an approximate singular vector for

$$\begin{pmatrix} L_{11} & 0 \\ 0 & \gamma \end{pmatrix}.$$

If $(0, 1/\gamma)^T$ is chosen as the approximate singular vector, all information on L_{11} contained in x will be lost. Conversely, if $(x_1, 0)^T$ is chosen, the new row corresponding to L_{22} will have been completely ignored. Instead, the estimate for

$$\sigma_{\min}\left(\begin{pmatrix} L_{11} & 0 \\ 0 & L_{22} \end{pmatrix}\right) \tag{5}$$

should be found by computing two independent estimates, $\hat{\sigma}_{\min}(L_{11})$ and $\hat{\sigma}_{\min}(L_{22})$. We should ignore L_{11} while computing an approximate singular vector x_2 for L_{22}. Then $1/\max(\|x_1\|_2, \|x_2\|_2)$ will be a good estimate for (5).

It may appear that (4) represents an unusual and trivial special case. In fact, matrices of this structure appear frequently *during* the factorization of sparse matrices (see, for example, [9], [10], [11]), where they represent leading principal minors of the matrix being factored. In such cases the indices of the rows in L_{11} correspond to nodes in one subtree of the elimination tree [19], and the indices of the rows in L_{22} correspond to another, independent, subtree.

The more interesting question is how to compute the singular value estimate when the first common ancestor of the nodes corresponding to L_{11} and L_{22} is encountered. This first common ancestor will be a node in the elimination tree that has more than one child. The submatrix corresponding to this first common ancestor and all of its descendations in the elimination tree will be a matrix in block-bordered diagonal form. In the case of two children, the matrix is

$$\mathcal{L} = \begin{pmatrix} L_{11} & 0 & 0 \\ 0 & L_{22} & 0 \\ l_{31} & l_{32} & \gamma \end{pmatrix}, \tag{6}$$

where γ is the parent of the last node of L_{11} and of the last node of L_{22} in the elimination tree. The question then is how to merge x_1 and x_2 upon encountering the border row at the ancestor γ. This is the topic of the next section.

3. An incremental condition estimator for sparse matrices. In this section we generalize the dense incremental condition estimator to integrate singular vectors of submatrices into an approximate singular vector for the whole matrix. The resulting technique makes incremental condition estimation applicable to sparse triangular matrices, in particular, triangular factors of sparse matrices.

Our generalization to the dense algorithm depends on recognizing two special forms. We begin with the augmented lower triangular matrix

$$\mathcal{L} = \begin{pmatrix} L & 0 \\ w^T & \gamma \end{pmatrix},$$

where the vector w may have known zero entries. If $\mathcal{L}$ can be symmetrically permuted

to the block diagonal form

$$\mathcal{L} = \begin{pmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & \tilde{w}^T & \gamma \end{pmatrix},$$

we reduce the problem to the smaller active matrix

$$\begin{pmatrix} L_2 & 0 \\ \tilde{w}^T & \gamma \end{pmatrix},$$

while separately preserving the approximate minimum singular vector from L_1.

We then apply one of two algorithms to the active matrix. If the matrix is symmetrically permutable to block-bordered diagonal form, as in (6), we apply the generalized incremental condition estimator described below. Otherwise, we apply the standard dense algorithm.

The issue of recognizing whether a general sparse triangular matrix is permutable into either of the special forms is not addressed in this paper. We are concerned with sparse triangular matrices that are either Cholesky or QR factors. In these cases the special structures are plainly exhibited by any permutation that is a postorder traversal of the corresponding elimination tree. Further, the elimination tree provides a simple characteristic to distinguish whether to apply the standard or the generalized incremental step—the standard step is used when the corresponding node of the elimination tree has one (or zero) children, while the existence of two or more children implies block-bordered diagonal form and use of the general step.

The block-bordered diagonal form problem has the following structure. Assume that we have lower triangular matrices $L_1, \ldots, L_p$ and their corresponding approximate singular vectors $x_1, \ldots, x_p$. Thus, x_i is a large norm solution to $L_i x_i = d_i$, where $\|d_i\|_2 = 1$ for $i = 1, 2, \ldots, p$. We now want to find a vector $y = (\hat{x}_1, \ldots, \hat{x}_p, \beta)^T$ such that $\|y\|_2$ is maximized subject to

$$\mathcal{L}y = \begin{pmatrix} L_1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & L_p & 0 \\ w_1^T & \cdots & \cdots & w_p^T & \gamma \end{pmatrix} y = \begin{pmatrix} \alpha_1 d_1 \\ \vdots \\ \vdots \\ \alpha_p d_p \\ \alpha_{p+1} \end{pmatrix} \quad \text{where} \quad \sum_{i=1}^{p+1} \alpha_i^2 = 1. \tag{7}$$

It easily follows from (7) that

$$\hat{x}_i = \alpha_i x_i, \quad i = 1, \ldots, p$$
$$\beta = \tfrac{1}{\gamma} \left(\alpha_{p+1} - \textstyle\sum_{i=1}^{p} \alpha_i (w_i^T x_i) \right).$$

The problem of maximizing $\|y\|_2{}^2$ can be restated as

$$\max \left[\textstyle\sum_{i=1}^{p} \alpha_i^2 x_i^T x_i + \frac{1}{\gamma^2} \left(\alpha_{p+1} - \sum_{i=1}^{p} \alpha_i (w_i^T x_i) \right)^2 \right]$$
$$\text{subject to} \textstyle\sum_{i=1}^{p+1} \alpha_i^2 = 1.$$

This problem is equivalent to

$$\begin{aligned} &\max a^T B a \\ &\text{subject to } \|a\|_2 = 1, \end{aligned} \tag{8}$$

where

$$a = (\alpha_1, \ldots, \alpha_{p+1})^T$$

and $B = (b_{ij})$ is defined as

$$\begin{array}{ll} b_{i,i} = x_i^T x_i + \frac{1}{\gamma^2}(w_i^T x_i)^2, & i = 1, \ldots, p \\ b_{p+1,p+1} = \frac{1}{\gamma^2} & \\ b_{i,p+1} = b_{p+1,i} = -\frac{1}{\gamma^2} w_i^T x_i, & i = 1, \ldots, p \\ b_{i,j} = b_{j,i} = \frac{1}{\gamma^2}(w_i^T x_i)(w_j^T x_j) & \text{otherwise.} \end{array} \tag{9}$$

The solution to (8) is a normalized eigenvector corresponding to the largest eigenvalue, $\mu_{\max}$, of B and in particular $\mu_{\max} = \|y\|_2^2$. To efficiently compute the largest eigenvalue and corresponding eigenvector of B, we take advantage of structure in (9). Define

$$\left.\begin{array}{l} \left.\begin{array}{l} \delta_i = \sqrt{x_i^T x_i} = \|x_i\|_2 \\ \zeta_i = w_i^T x_i \end{array}\right\}, \quad i = 1, \ldots, p, \\ z = (\zeta_1, \ldots, \zeta_p, -1)^T \\ D = \operatorname{diag}(\delta_1, \ldots, \delta_p, 0)\,. \end{array}\right\} \tag{10}$$

Then we may write B as

$$B = D^2 + \frac{1}{\gamma^2} z z^T = R R^T,$$

where

$$R = (D + \frac{1}{\gamma} z e_{p+1}^T)$$

and e_{p+1} is the $(p+1)$st canonical unit vector. Notice that B is a scaled symmetric rank-one update to the diagonal matrix D^2. Following [4], [5], [12], the roots

$$\mu_{\max} = \mu_1 \geq \mu_2 \geq \mu_{p+1} \geq 0$$

of the secular equation

$$g(\mu) = 1 + \frac{1}{\gamma^2}\left(\sum_{i=1}^{p} \frac{\zeta_i^2}{\delta_i^2 - \mu} - \frac{1}{\mu}\right) \tag{11}$$

determine the eigenvalues of B. In particular,

$$\mu_{\max} = \sigma_{max}^2(R),$$

and the vector a is the left singular vector of R corresponding to $\sigma_{\max}(R)$. It is shown in [4], [5] how the roots of $g(\mu)$ can be computed cheaply using rational approximations of $g(\mu)$. Moreover, the technique in [4] allows one to compute only the largest eigenvalue. Once $\mu_{\max}$ has been computed, a is obtained by

$$a = \frac{1}{\|h\|_2} h,$$

where

$$h = (D^2 - \mu_{\max} I)^{-1} z\,.$$

The corresponding estimate for the smallest singular value of $\mathcal{L}$ is

$$\hat{\sigma}_{\min}(\mathcal{L}) = \frac{1}{\sqrt{\mu_{\max}}}.$$

However, it should be noted that in [4] the vector a is computed simultaneously with $\mu_{\max}$.

The cost of applying this incremental condition estimator to a sparse factor consists of three parts. One is access to the entries in the matrix, each of which is used once in an inner product. This is essentially the cost of a forward solve, $\mathcal{O}(nz)$ operations, where nz is the number of nonzeros in L. Scaling the singular vector x would appear to require $n^2/2$ operations overall. However, explicit scaling is not required, as will be seen in §6; the real requirement is a small constant number of operations per row. Finally we have the cost of n small eigenproblems, most of which require finding the largest solution of a quadratic equation.

4. The estimator as approximation of secular equations. In this section we examine the relationship between our estimator and the secular equation of the matrix $\mathcal{L}^T\mathcal{L}$. In particular, we show that our estimator finds the smallest root of the secular equation truncated to $(p+1)$ terms. This truncated secular equation also corresponds to the exact secular equation of a submatrix of $\mathcal{L}$. We begin by considering the singular value decompositions

$$L_i = U_i S_i V_i^T$$

of L_i. Further, assume that incremental condition estimation was exact and computed

$$x_i = \frac{1}{\sigma_{\min}(L_i)} v_{n_i}^{(i)}, \tag{12}$$

where $v_{n_i}^{(i)}$ is the right singular value of L_i corresponding to the smallest singular value $\sigma_{\min}(L_i)$. In particular, then

$$\mathcal{L} = \begin{pmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_p & \\ w_1^T & w_2^T & \cdots & w_p^T & \gamma \end{pmatrix} = \begin{pmatrix} U_1 S_1 V_1^T & & & & \\ & U_2 S_2 V_2^T & & & \\ & & \ddots & & \\ & & & U_p S_p V_p^T & \\ w_1^T & w_2^T & \cdots & w_p^T & \gamma \end{pmatrix} =$$

$$\begin{pmatrix} U_1 & & & & \\ & U_2 & & & \\ & & \ddots & & \\ & & & U_p & \\ & & & & 1 \end{pmatrix} \begin{pmatrix} S_1 & & & & \\ & S_2 & & & \\ & & \ddots & & \\ & & & S_p & \\ y_1^T & y_2^T & \cdots & y_p^T & \gamma \end{pmatrix} \begin{pmatrix} V_1 & & & & \\ & V_2 & & & \\ & & \ddots & & \\ & & & V_p & \\ & & & & 1 \end{pmatrix}$$

where

$$y_i^T = w_i^T V_i.$$

Hence the singular values of $\mathcal{L}$ are those of

$$\mathcal{S} = \begin{pmatrix} S_1 & & & & \\ & S_2 & & & \\ & & \ddots & & \\ & & & S_p & \\ y_1^T & y_2^T & \cdots & y_p^T & \gamma \end{pmatrix}.$$

The secular equation, f, of $\mathcal{L}^T\mathcal{L}$ is given by

$$f(\lambda) = \sum_{i=1}^{p} \left(\sum_{j=1}^{n_i} \frac{(y_i^{(j)})^2}{\sigma_j^2(L_i) - \lambda} \right) - \frac{\gamma^2}{\lambda} + 1, \tag{13}$$

where $y_i^{(j)}$ is the jth entry of y_i. The roots

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n, \qquad n = 1 + \sum_{i=1}^{p} n_i,$$

of f determine the singular values of $\mathcal{L}$ in that

$$\sigma_i(\mathcal{L}) = \sqrt{\lambda_i}, \qquad i = 1, \ldots, n.$$

It is the goal of our estimator to approximate λ_n, the smallest root of f. A natural approach to such an approximation would be to take, say, k terms of f, forming $\tilde{f}$ and use the roots of $\tilde{f}$ to approximate some of those of f. Since we seek to approximate λ_n, we should take those terms in f which "contribute" to the root λ_n. For example, let

$$\mathcal{L} = \begin{pmatrix} 1 & & & \\ & \sqrt{2} & & \\ & & \sqrt{3} & \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Then the secular equation for $\mathcal{L}^T\mathcal{L}$ is given by

$$f(x) = \frac{1}{x} + \frac{1}{(x-1)} + \frac{1}{(x-2)} + \frac{1}{(x-3)} - 1.$$

Note that the smallest root of

$$\frac{1}{x} - 1$$

and even more so the smallest root of

$$\frac{1}{x} + \frac{1}{(x-1)} - 1$$

are close approximations to the smallest root of f. Along this same line then, a natural choice for $\tilde{f}$ would be to take from each inner sum in (13) that term with smallest $\sigma_j^2(L_i)$ in the denominator. Then $\tilde{f}$ is given by

$$\tilde{f}(\lambda) = \sum_{i=1}^{p} \frac{(y_i^{\min})^2}{\sigma_{\min}^2(L_i) - \lambda} - \frac{\gamma^2}{\lambda} + 1, \tag{14}$$

where $y_i^{\min}$ is used as a shorthand for $y_i^{(n_i)}$, the last entry of y_i. Note that $\tilde{f}(\lambda)$ is the secular equation of the matrix $T^T T$, where

$$T = \begin{pmatrix} \sigma_{\min}(L_1) & & & & \\ & \sigma_{\min}(L_2) & & & \\ & & \ddots & & \\ & & & \sigma_{\min}(L_p) & \\ y_1^{\min} & y_2^{\min} & \cdots & y_p^{\min} & \gamma \end{pmatrix},$$

which is a submatrix of $\mathcal{S}$. Our condition number estimator finds the reciprocal of the smallest singular value of T . This is seen algebraically by using (10) and (12) to obtain

$$\lambda\gamma^2 g(\lambda) = \gamma^2\lambda + \sum_{i=1}^{p} \frac{(w_i^T v_{\min}^{(i)})^2}{\frac{1}{\lambda} - \sigma_{\min}^2(L_i)} - 1.$$

By substituting $\mu = \frac{1}{\lambda}$, we have

$$\lambda\gamma^2 g(\lambda) = -\tilde{f}(\mu).$$

Hence the largest root of g is the reciprocal of the smallest root of $\tilde{f}$, which in turn is the square of the smallest singular value of T. This shows that our incremental condition estimation scheme can be viewed as *approximating the smallest root of the secular equation by a truncated form of the secular equation or equivalently as approximating the smallest singular value of $\mathcal{L}$ by that of the smallest singular value of T.*

As an example, consider the 7×7 matrix

$$\mathcal{L} = \begin{pmatrix} -3.7303 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.2246 & -3.5687 & 0 & 0 & 0 & 0 & 0 \\ 0.5444 & -0.3719 & 3.0423 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3.2895 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.0722 & -2.5597 & 0 & 0 \\ 0 & 0 & 0 & 1.3312 & 0.2539 & 5.7006 & 0 \\ -1.6300 & 2.3245 & 0.4369 & -1.5376 & -2.5385 & 1.0620 & -1.3614 \end{pmatrix}.$$

The singular values of L_1 are $5, 3$, and 2.7; the singular values of L_2 are $6, 4$, and 2. The square of these values and $\lambda = 0$ are the poles for $f(\lambda)$. Since we are concerned only with the smallest singular value of $\mathcal{L}$, Fig. 1 is only for the interval $[0, 10]$. In this figure the poles of $\tilde{f}$ are shown as dashed lines, and f (which is the upper curve) has four roots. $\tilde{f}$ (which is the lower curve) has all three roots in the interval. We see that the smallest roots of f and $\tilde{f}$ lie close together.

Our estimator is computed by an a priori choice of $p + 1$ terms to approximate the complete secular equation. Equivalently, we have chosen a principal submatrix of order $p + 1$ whose singular values are approximations to the singular values of the larger matrix. The use of fewer than $p + 1$ terms, or a submatrix of order less than $p + 1$, is unreasonable because it would necessarily ignore some diagonal block(s). Hence our estimator uses the "fewest" possible terms. We can use the submatrix formulation of this approximation to show an optimality result for this particular a priori choice—this is the only choice of $p + 1$ terms that is guaranteed to provide an estimate that is always at least as small as the smallest singular value from each block, that is, is consistent with the information from the diagonal blocks of the matrix.

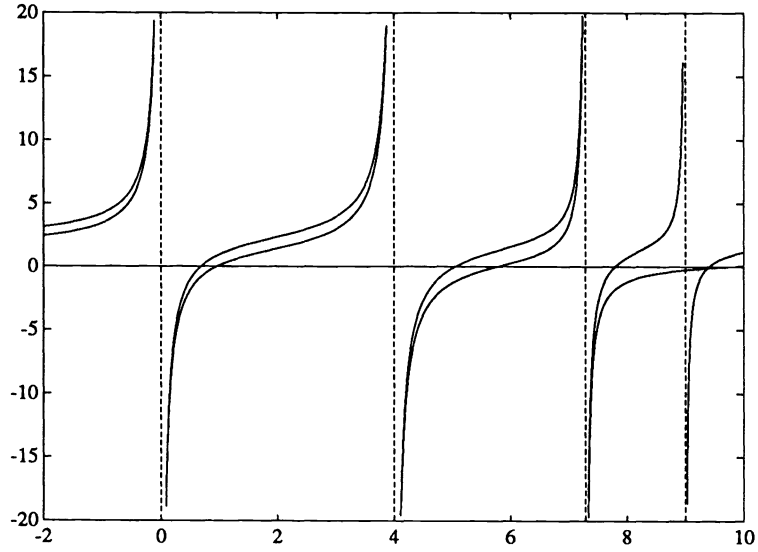


FIG. 1. *Approximation of f by $\tilde{f}$ for the matrix $\mathcal{L}$ in* (4).

The triangular structure implies that $T^T T$ is a diagonal block of $P\mathcal{S}^T\mathcal{S}P^T$ for a suitably chosen permutation matrix P. As a result, the singular values of T interlace with those of $\mathcal{S}$. Further, the same argument applies to each diagonal entry of T itself to show

$$\sigma_{\min}(\mathcal{L}) \leq \sigma_{\min}(T) \leq \min\left\{\{\sigma_{\min}^{(i)}\}_{i=1}^{p}, \gamma\right\}.$$

The terms over which the minimization is taken are precisely the least singular values (or more generally, our estimate thereof) of each of the diagonal blocks, independent of the values of the y_i's. Thus, our estimate is consistent and decreases, as does the least singular value of the augmented triangular matrix. Clearly no other a priori choice of a principal submatrix of order $p+1$ can have all of the entries

$$\left\{\{\sigma_{\min}^{(i)}\}_{i=1}^{p}, \gamma\right\}$$

on its diagonal, and so cannot guarantee to show consistency for all values of y_i, particularly for the case $y_i = 0$.

5. Numerical experiments. To assess the numerical reliability of the suggested scheme, we performed a suite of tests using PRO-MATLAB [18]. We generated block-diagonal matrices (where each diagonal block is lower triangular) of order 100 and added one dense column with random values from the uniform distribution on $[-1, 1]$. For each matrix, we varied the number and size of the diagonal blocks: we either generated fifty 2×2 matrices, ten 10×10 matrices, two 50×50 matrices, or varied the size of the blocks as shown in Fig. 2. We experimented with several singular value distributions for the diagonal blocks: A random distribution, where the singular values were random numbers from the uniform distribution on $[-1, 1]$, an exponential distribution, where

$$\sigma_j = \alpha^j, j = 1, \ldots, n_i \text{ and } \alpha = 10^{-6/n_i}$$

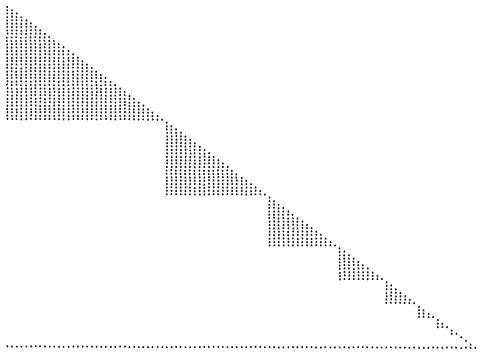


FIG. 2. *Lower triangular matrix with diagonal blocks of varying size.*

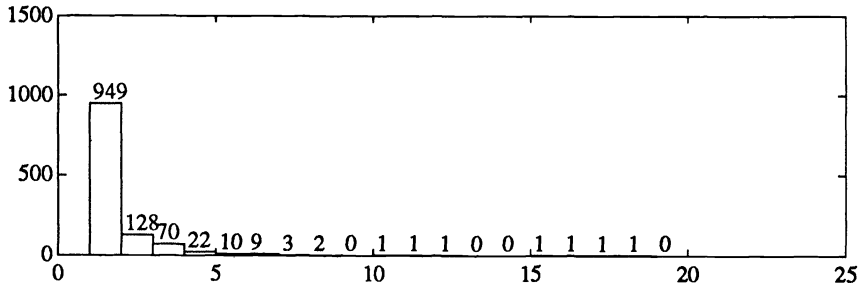


FIG. 3. *Overall distribution of* $\frac{\sigma_{\min}(L)}{\tilde{\sigma}_{\min}(L)}$.

and a sharp-break distribution, where all singular values are 1 except for the smallest, which is 10^{-6}. For each singular value distribution, we generated a diagonal matrix containing the desired singular values and then multiplied this matrix from the left and right with random orthogonal matrices generated using the method of Stewart [20]. A lower triangular matrix with the same singular value decomposition was then obtained by performing a QR factorization with and without pivoting of the resulting matrix and transposing the triangular factor so obtained. For each class of matrices, we performed fifty tests, resulting in a total of 1200 examined matrices.

Figure 3 shows a histogram of the overestimate of the smallest singular value that our scheme produces. We see that in 949 cases, we overestimated the smallest singular value at most by a factor of two and in no case did we overestimate the smallest singular value by more than a factor of 18.

The distribution of the singular values of the diagonal blocks and the size of the diagonal blocks had some influence on the accuracy of the estimate, whereas the other factors played no significant role. For the sharp-break distribution, we never overestimated the smallest singular value by more than a factor of two. The same was true when the sizes of the diagonal blocks were 2 and 10, respectively. The results for the random and exponential distributions are shown in Fig. 4. The structure of $g(\mu)$ in (11) is quite different for these two cases. If the singular value estimates for the individual blocks were exact, $g(\mu)$ would have only two roots for the exponential distribution, since the smallest singular values of all diagonal blocks are identical. We can infer that the $g(\mu)$ actually computed will have p roots that are very close. In contrast, the roots of $g(\mu)$ should be well distributed in the case of the random

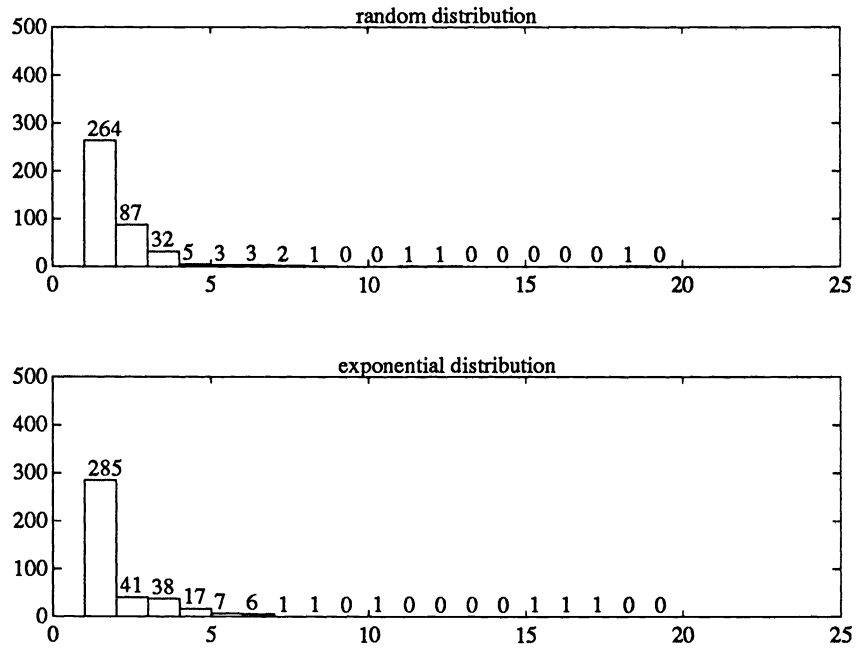


FIG. 4. *Distribution of* $\frac{\sigma_{\min}(L)}{\tilde{\sigma}_{\min}(L)}$ *by singular values of diagonal blocks.*

distribution. We see that our estimator performs well in both cases.

As for the influence of the size of the diagonal blocks on the quality of our estimate, the situation for diagonal blocks of size 50 and diagonal blocks of varying size is shown in Fig. 5. It is not surprising that the estimate deteriorates somewhat as the sizes of the diagonal blocks increase. This results when the incremental condition estimation scheme described in § 2 becomes less accurate as the size of the matrix increases.

6. Implementation for sparse problems. In this section we show how our estimator can fit naturally into algorithms for computing triangular factorizations of sparse matrices. Here we examine the particular case of a Cholesky factorization, but the same ideas apply to a QR factorization. A particular issue that we face in the context of sparse factorizations is that the sparse factor L is usually not stored by rows. Column storage of one form or other is typical, as in [8], [10], [11], yet our estimator appears to depend on access to the rows of L in order to compute the inner products $w^T x$. Our solution to this dilemma uses the recursive block structure that is found in sparse factor matrices, together with a key property of the updating of our estimator.

It will suffice to consider the Cholesky factorization for a symmetric positive definite matrix $\mathcal{A}$ of the form

$$\mathcal{A} = \begin{pmatrix} A_{11} & & A_{31}^T \\ & A_{22} & A_{32}^T \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \mathcal{L}\mathcal{L}^T,$$

where A_{ii} is n_i by n_i. This form suffices because it, together with the obvious generalizations to bordered forms with p diagonal blocks, is the form that appears as the active submatrix when the node corresponding to the first row of A_{33} has more than one child in the elimination tree for some larger sparse matrix. Note that the

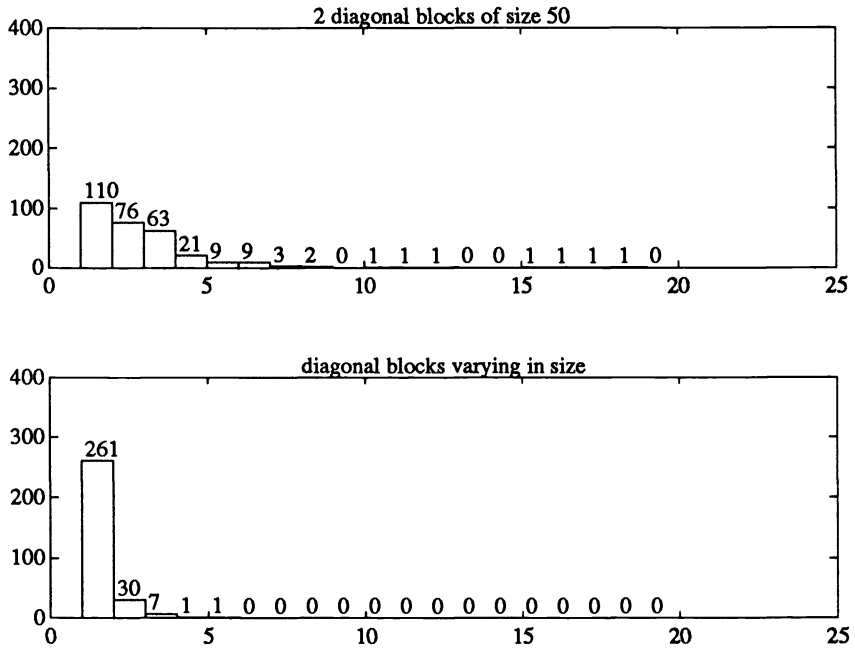


FIG. 5. *Distribution of* $\frac{\sigma_{\min}(L)}{\hat{\sigma}_{\min}(L)}$ *by size of diagonal blocks.*

Cholesky factor of $\mathcal{A}$ is of the form

$$\mathcal{L} = \begin{pmatrix} L_{11} & & \\ & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{pmatrix}.$$

We can compute the Cholesky factorization L_{11} for A_{11}

$$A_{11} = L_{11}L_{11}^T$$

and the entries in L_{31},

$$L_{31} = A_{31}L_{11}^{-T}$$

by some form of a block Cholesky algorithm or by inner or outer product factorizations of the first n_1 columns of $\mathcal{A}$. Concurrently we compute an approximate minimum singular vector, x_1, for A_{11}. We use these to compute the vector

$$t_1 = L_{31}x_1\,.$$

The entries in this vector represent all of the nonzero inner products between later rows of $\mathcal{L}$ and the current value of x_1. Note that t_1 can be computed as the sum of a set of sparse vectors in cases where L_{31} is stored by columns. We save t_1 and $\|x_1\|_2^2$. As will be seen shortly, we have no further need to access x_1, L_{11}, or L_{31}.

Similarly, we compute the Cholesky decomposition of the second block column, computing L_{22} and L_{32} by

$$\begin{aligned} A_{22} &= L_{22}L_{22}^T \\ L_{32} &= A_{32}L_{22}^{-T}. \end{aligned}$$

In the standard way, we compute an approximate minimum singular vector x_2. Again, we compute and save $\|x_2\|_2^2$ and the vector $t_2 = L_{32}x_2$.

It is necessary to use the general form of the condition estimator when the first diagonal element d_{11} of A_{33} is encountered. At this point, w_1^T of (9) is the first row of L_{31} and w_2^T of (9) is the first row of L_{32}. The estimator requires only the inner products of these rows with x_1 and x_2, respectively, and the norms of x_1 and x_2. The norms were saved and the inner products are given by the first components of t_1 and t_2, respectively:

$$t_1^{(1)} = w_1^T x_1 \quad \text{and} \quad t_2^{(1)} = w_2^T x_2.$$

We have sufficient information to compute the triple $(\alpha_1, \alpha_2, \alpha_3)$, as required by our general estimator. The last entry of the new minimum singular vector is given by

$$\beta = \frac{1}{\gamma}\left(\alpha_3 - \alpha_1 t_1^{(1)} - \alpha_2 t_2^{(1)}\right).$$

Had we saved the singular vector approximations from the leading subblocks, we would have a new approximate minimum singular vector

$$\tilde{x}^{(1)} = \begin{pmatrix} \alpha_1 x_1 \\ \alpha_2 x_2 \\ \beta \end{pmatrix}.$$

We will use the structure of $\tilde{x}^{(1)}$ to compute minimum singular value and, implicitly, vector approximations for the remaining rows in L_{33}.

In computing the entries corresponding to the second and later rows of L_{33}, we use the dense condition estimator, generating 2×2 eigenproblems, implicitly given approximations to singular vectors $\tilde{x}^{(i)}$ and scaling factors $s^{(i)}$ and $c^{(i)}, i = 2, \ldots, n_3$. The dense estimator applies the same multiplier $s^{(i)}$ to all previous entries in the approximate minimum singular vector. This will allow us to recover the necessary inner products from t_1 and t_2.

For example, the computation of the entry corresponding to the third row of L_{33} requires the values of $\|\tilde{x}^{(2)}\|_2^2$ and

$$\begin{pmatrix} L_{31}^{(3)} & L_{32}^{(3)} & L_{33}^{(3)} \end{pmatrix} \tilde{x}^{(2)},$$

where $L_{31}^{(3)}$ and $L_{32}^{(3)}$ are the third rows of L_{31} and L_{32}, respectively, and $L_{33}^{(3)}$ is the third row of L_{33}, excluding the diagonal. But

$$\begin{aligned} \left(L_{31}^{(3)} \mid L_{32}^{(3)} \mid L_{33}^{(3)} \right) \tilde{x}^{(2)} &= \left(L_{31}^{(3)} \mid L_{32}^{(3)} \mid L_{33}^{(3)} \right) \left(\begin{array}{c} s^{(2)}\alpha_1 x_1 \\ \hline s^{(2)}\alpha_2 x_2 \\ \hline s^{(2)}\beta \\ \\ c^{(2)} \end{array} \right) \\ &= s^{(2)}(\alpha_1 t_1^{(3)} + \alpha_2 t_2^{(3)}) + L_{33}^{(3)} \begin{pmatrix} s^{(2)}\beta \\ c^{(2)} \end{pmatrix}. \end{aligned}$$

The determination of α_1 and α_2 essentially changes the separate singular vector estimates from the first and second blocks into a single vector, namely,

$$x = \begin{pmatrix} \alpha_1 x_1 \\ \alpha_2 x_2 \end{pmatrix}.$$

Similarly, the vectors of inner products t_1 and t_2 can be coalesced into a single vector

$$t = \alpha_1 t_1 + \alpha_2 t_2 .$$

Then the inner products required by the estimator can be found by repeatedly applying the scaling factors $s^{(i)}$ to t, and from β and $\{s^{(i)}, c^{(i)}\}$.

Thus we are able to avoid any references to L_{31}, L_{32}, x_1, or x_2 by computing the vectors t_1 and t_2. In practical cases the block-bordered form of $\mathcal{L}$ appears as a subblock of a larger sparse matrix; there are other nonzero blocks corresponding to L_{ii}. In these cases the inner product vectors t_i are longer, but all entries are subject to the scaling operations. The key issue is that the inner product vector that results from L_{33} must be modified to account for the yet unused rows of the inner product vectors from L_{11} and L_{22}. We leave it to the reader to fill in the details for one level of recursion, which will suffice to show the general algorithm.

We note that this sparse implementation does not require ever saving the approximate singular vector x. At each block step, it uses only information which is either from the current block column or is given recursively. Moreover, the recursive information, the inner product data corresponding to each block column, has a particularly clean interpretation in the data structures of a multifrontal Cholesky factorization [8], where storage for this data can be found simply by augmenting each frontal update matrix with a single column. An equally clean interpretation can be found for QR factorizations, using the row merge scheme of Liu [17] or the multifrontal Householder scheme of Lewis, Pierce, and Wah [16].

7. Conclusions. Incremental condition estimation is a technique that allows one to maintain cheap estimates of the smallest singular value of a triangular matrix as it is generated one column or row at a time. In this paper, we generalized incremental condition estimation to handle arbitrary, and particularly sparse factor, matrices. The computation requirements are similar in cost to other condition estimation schemes for sparse matrices [13], but our scheme provides a Euclidean norm estimate. In addition, it is not necessary to reaccess the previously generated triangular matrix when a new row or column is added, making the scheme attractive in parallel or out-of-core environments. We showed how our scheme can be interpreted as approximating the secular equations determining the true singular values. Numerical experiments indicate that the scheme is reliable despite its small computational cost. Moreover, we demonstrated its applicability in a sparse setting, thus making our estimator both practical and effective.

Acknowledgments. The first author would like to thank Andreas Griewank for some enlightening discussions.

REFERENCES

[1] C. H. BISCHOF, *A block QR factorization algorithm using restricted pivoting*, Tech. Report ANL/MCS-P94-0789, Mathematics and Computer Sciences Division, Argonne National Laboratory, Argonne, IL, 1988.

[2] ———, *Incremental condition estimation*, Tech. Report ANL/MCS–P15–1088, Mathematics and Computer Sciences Division, Argonne National Laboratory, Argonne, IL, 1988; SIAM J. Matrix Anal. Appl., to appear.

[3] ———, *A parallel QR factorization algorithm with controlled local pivoting*, Tech. Report ANL/MCS–P21–1088, Mathematics and Computer Sciences Division, Argonne National Laboratory, Argonne, IL, 1988.

[4] J. R. BUNCH AND C. R. NIELSEN, *Updating the singular value decomposition*, Numer. Math., 31(1978), pp. 111–129.

[5] J. R. BUNCH, C. R. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31(1978), pp. 31–48.

[6] A. K. CLINE, A. R. CONN, AND C. F. VAN LOAN, *Generalizing the* LINPACK *Condition Estimator*, Lecture Notes in Mathematics Vol. 909, Springer–Verlag, Berlin, New York, 1982, pp. 73–83.

[7] A. K. CLINE, C. B. MOLER, G. W. STEWART, AND J. H. WILKINSON, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal., 16(1979), pp. 368–375.

[8] I. S. DUFF AND J. K. REID, *The multifrontal solution of indefinite sparse systems*, ACM Trans. Math. Software, 9(1983), pp. 302–325.

[9] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct methods for Sparse Matrices*, Oxford Press, London, 1987.

[10] S. C. EISENSTAT, M. C. GURKY, M. H. SCHULTZ, AND A. H. SHERMAN, *Yale sparse matrix package, 1: The symmetric codes*, Internat. J. Numer. Meth. Engrg., 18(1982), pp. 235–250.

[11] A. GEORGE AND J. W. H. LIU, *Computer Solution of Large Sparse Positive-Definite Systems*, Prentice-Hall, NJ, 1981.

[12] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15(1973), pp. 318–334.

[13] J. G. LEWIS AND R. G. GRIMES, *Condition number estimation for sparse matrices*, SIAM J. Sci. Statist. Comput., 2(1981), pp. 384–388.

[14] N. J. HIGHAM, *Efficient algorithms for computing the condition number of a tridiagonal matrix*, SIAM J. Sci. Statist. Comput., 7(1986), pp. 150–165.

[15] ———, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29(1987), pp. 575–596.

[16] J. G. LEWIS, D. J. PIERCE, AND D. C. WAH, *A multifrontal Householder QR factorization*, Tech. Report ECA–TR–127, Boeing Computer Services, Engineering and Scientific Services Division, Seattle, WA, 1989.

[17] J. W. H. LIU, *On general row merging schemes for sparse Givens transformations*, SIAM J. Sci. Statist. Comput., 7(1986), pp. 1190–1211.

[18] C. MOLER, J. LITTLE, AND S. BANGERT, PRO-MATLAB *User's Guide*, The Mathworks, Sherborn, MA, 1987.

[19] R. SCHREIBER, *A new implementation of sparse Gaussian elimination*, ACM Trans. Math. Software, 8(1982), pp. 256–276.

[20] 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.